



Pengendalian lampu dan door lock berbasis IOT untuk efisiensi kontrol di Yayasan Gunung Wayang Manunggal

IOT-based light and door lock control for control efficiency at the Gunung Wayang Manunggal Foundation

Ahmad Qashid Husaini*, Herliyani Hasanah, Nurchim

*Fakultas Ilmu Komputer Universitas Duta Bangsa Surakarta, Jl. Bhayangkara No.55, Tipes, Kec. Serengan, Kota Surakarta, Jawa Tengah, Indonesia 57154

INFORMASI ARTIKEL

Article History:

Submission: 29-06-2024

Revised: 23-07-2024

Accepted: 02-08-2024

Kata Kunci:

Internet of Things;

Telegram; ESP32; Efisiensi control; Kendali jarak jauh.

Keywords:

Internet of Things;

Telegram; ESP32; Control Efficiency; Remote Control.

* Korespondensi:

Ahmad Qashid Husaini

202020540@mhs.udb.ac.id

ABSTRAK

Perkembangan *teknologi Internet of Things (IoT)* telah mengubah cara kita berinteraksi dengan lingkungan, termasuk pengendalian jarak jauh. Saat ini, Yayasan Gunung Wayang Manunggal sedang menghadapi permasalahan dalam pengendalian jarak jauh mereka, yang meliputi pengendalian lampu dan kunci pintu ruangan bimbel (*door lock*). Untuk mengatasi permasalahan tersebut, penelitian ini bertujuan merancang sistem pengendalian lampu dan *door lock* jarak jauh menggunakan HP guna meningkatkan *efisiensi kontrol* di Yayasan. Penelitian ini menggunakan metode *Waterfall* karena sistem yang dihasilkan akan sesuai dengan kebutuhan admin yayasan. Setelah dilakukan pengujian, hasil dari 10 kali pengujian menunjukkan bahwa sistem berhasil mengontrol semua alat tanpa ada error, dengan tingkat akurasi 100%. *Analisis* mencakup dua kebutuhan, pertama admin harus dapat mengontrol lampu dan pintu (*door lock*) yang ada di Yayasan dari jarak jauh menggunakan aplikasi Telegram, dan kedua admin harus dapat memantau kondisi lampu dan pintu. Fitur ini memungkinkan pengendalian dan pemantauan tanpa harus hadir di yayasan. Sistem ini mempermudah admin mengontrol lampu dan pintu dari jarak jauh menggunakan HP, sehingga lebih efisien dan mudah dioperasikan. Respon positif dari admin dan peminjam ruangan menunjukkan bahwa sistem ini sangat membantu dan mempercepat akses ke fasilitas yayasan.

ABSTRACT

The development of Internet of Things (IoT) technology has changed the way we interact with the environment, including remote control. Currently, the Gunung Wayang Manunggal Foundation is facing problems with their remote control, which includes controlling the lights and the door lock for the tutoring room. To overcome this problem, this research aims to design a remote lighting and door lock control system using a cellphone to increase control efficiency at the Foundation. This research uses the Waterfall method because the resulting system will suit the needs of the foundation admin. After testing, the results of 10 tests showed that the system succeeded in controlling all tools without any errors, with an accuracy rate of 100%. The analysis includes two needs, firstly the admin must be able to control the lights and doors (door locks) at the Foundation remotely using the Telegram application, and secondly the admin must be able to monitor the condition of the lights and doors. This feature allows control and monitoring without having to be present at the foundation. This system makes it easier for admins to control lights and doors remotely using a cellphone, making it



more efficient and easier to operate. Positive responses from admins and room borrowers show that this system is very helpful and speeds up access to foundation facilities.

1. PENDAHULUAN

Perkembangan teknologi *Internet of Things* (IoT) telah membawa dampak signifikan dalam mengubah cara kita berinteraksi dengan lingkungan sekitar dan mempengaruhi kehidupan manusia secara lebih baik [1]. IoT merupakan paradigma baru yang memungkinkan perangkat digital seperti sensor dan aktuator berkomunikasi dan berinteraksi tanpa campur tangan manusia [2]. Benda-benda disekitar kita dapat terhubung ke internet melalui IOT dan dapat dikontrol melalui smartphone [3]. Salah satu implementasi inovatif dari konsep IOT adalah sistem pengendalian jarak jauh yang memungkinkan pengendalian perangkat elektronik dari jarak jauh menggunakan koneksi internet [4]. Yayasan Gunung Wayang Manunggal yang merupakan sebuah lembaga bakti sosial yang mengembangkan potensi anak melalui kegiatan bimbingan belajar, saat ini menghadapi masalah dalam *efisiensi* kontrol fasilitas mereka. Yayasan hanya memiliki satu orang admin yang bertanggung jawab untuk mengelola dan memberikan akses fasilitas seperti lampu dan kunci ruangan bimbingan belajar. Seringkali admin yayasan harus meninggalkan lokasi untuk keperluan mendesak, seperti perjalanan dinas. Hal ini mengharuskan admin kembali ke yayasan untuk membuka kunci dan menyalakan lampu secara manual setiap kali ada penyewa atau peminjam ruangan. Situasi ini menyebabkan keterbatasan *aksesibilitas* dan ketergantungan pada tugas manual [5], yang dapat mengganggu *efisiensi* pengelolaan fasilitas dan dapat menguras waktu serta tenaga [6]. Dengan adanya koneksi internet dan perangkat IoT yang terhubung proses kontrol dapat dilakukan secara otomatis dan terpusat sehingga menghemat waktu dan tenaga [7]. Oleh karena itu, Penelitian ini bertujuan untuk merancang dan membangun sistem pengendalian lampu dan pintu ruangan jarak jauh dengan aplikasi telegram berbasis *Internet of Things* (IOT) di Yayasan Gunung Wayang Manunggal yang dapat mengurangi ketergantungan pada tugas manual, dan memberikan akses yang lebih cepat dan mudah. Teknologi ini memungkinkan pengendalian jarak jauh, sehingga admin yayasan dapat mengontrol lampu dan pintu tanpa harus berada di lokasi fisik yayasan. Implementasi teknologi ini menggunakan mikrokontroler ESP32 sebagai pusat kendali yang terhubung ke jaringan Wi-Fi memungkinkan integrasi dengan berbagai sensor dan aktuator [8]. Aplikasi telegram dipilih untuk sistem pengendalian ini karena beberapa alasan antara lain karena *library* nya mendukung berbagai mikrokontroler, terutama ESP32, serta telegram menyediakan API (*Application Programming Interface*) gratis yang mudah digunakan atau diintegrasikan untuk pengembangan telegram bot [9].

Beberapa penelitian sebelumnya yang diambil sebagai bahan pertimbangan dan referensi untuk penelitian ini. Misalnya, penelitian berjudul "Prototipe pengendalian lampu dan AC jarak Jauh Dengan Jaringan Internet menggunakan aplikasi telegram" menunjukkan bahwa aplikasi telegram berhasil terhubung dengan ESP8266 dan mampu mengendalikan lampu dan AC dengan jarak jauh [10]. Perbedaan dengan penelitian kali ini terletak pada *mikrokontroler* yang digunakan, meskipun sama-sama menggunakan telegram bot untuk pusat kontrolnya. Penelitian kedua yang masih berkaitan dengan pengendalian lampu jarak jauh tentang rancangan *prototype* alat kendali otomatis penjemur pakaian berbasis IOT menggunakan NodeMCU ESP32 dengan telegram bot sebagai pusat kontrol, juga menunjukkan implementasi yang relevan [11], meskipun perbedaannya terletak pada konsep dan perangkat *mikrokontroler* yang digunakan. Selanjutnya penelitian mengenai pengendalian lampu berbasis *mikrokontroler* menggunakan aplikasi android dan blynk, penelitian ini juga relevan, tetapi perbedaan utama terletak pada aplikasi atau perangkat lunak yang digunakan untuk mengontrol lampu dan *door lock* [12].

Selanjutnya penelitian yang berkaitan dengan penggunaan *solenoid door lock* untuk kontrol jarak jauh adalah "Sistem Keamanan Rumah Berbasis *Mikrokontroler* Arduino dan SMS Gateway" penelitian ini mengembangkan sistem pengamanan pintu menggunakan kunci *elektronik* yang dikendalikan melalui SMS *gateway*, dimana pengguna dapat membuka pintu melalui SMS [13].

Perbedaan utama dengan penelitian ini adalah penggunaan SMS sebagai media komunikasi, sedangkan penelitian di Yayasan Gunung Wayang Manunggal menggunakan aplikasi telegram berbasis IOT untuk pengendalian jarak jauh, yang memungkinkan interaksi yang lebih interaktif dan *real-time* dibandingkan SMS. Penelitian lainnya yang masih berkaitan adalah "Prototype *smart home* dengan Nodemcu berbasis IOT" penelitian ini mengimplementasikan sistem *smart door lock* yang dikendalikan melalui *Internet of Things* (IoT) menggunakan ESP8266 [14]. Perbedaan dengan penelitian ini adalah penggunaan ESP8266 sebagai *mikrokontroler* dan aplikasi blynk sebagai kontrolnya, sedangkan penelitian ini menggunakan ESP32 dan Bot Telegram. ESP32 dipilih karena kemampuannya yang lebih unggul dalam integrasi dengan berbagai perangkat dan kecepatan dalam pengiriman perintah melalui Telegram [15]. Selain itu, penelitian "Keamanan pintu rumah berbasis *Internet Of Things* dengan ESP32" ini juga relevan dengan penelitian ini mengembangkan sistem keamanan rumah yang dapat dikendalikan melalui aplikasi blynk dengan menggunakan mikrokontroler ESP32. Sistem ini memungkinkan pengguna untuk mengontrol dan memantau kondisi rumah dari jarak jauh [16]. Sedangkan perbedaan utama terletak pada penggunaan aplikasi untuk mengontrolnya, di mana penelitian di Yayasan Gunung Wayang Manunggal menggunakan aplikasi telegram, dikarenakan terdapat API bot serta lebih populer dan *fleksibel* untuk diintegrasikan dengan berbagai perangkat [17]. Dengan meninjau penelitian-penelitian sebelumnya, penelitian ini bertujuan untuk memberikan solusi yang lebih efisien dan responsif dalam pengelolaan fasilitas di Yayasan, mengatasi keterbatasan aksesibilitas dan ketergantungan pada tugas manual dengan menggunakan teknologi berbasis IoT dengan aplikasi Telegram.

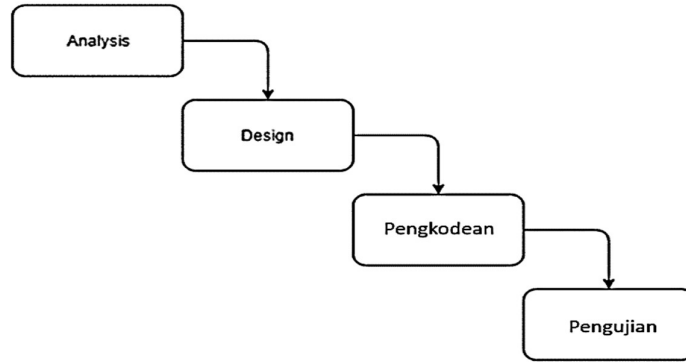
Penelitian ini menggunakan beberapa alat IoT dengan spesifikasi tertentu untuk mencapai tujuan pengelolaan fasilitas yang efisien dan responsif di Yayasan. Alat utama yang digunakan adalah mikrokontroler ESP32, yang memiliki prosesor dual-core dengan kecepatan hingga 240 MHz, 520 KB SRAM, mendukung flash eksternal hingga 16 MB, dan sudah memiliki Wi-Fi serta Bluetooth di dalamnya. ESP32 juga memiliki 34 pin GPIO yang dapat digunakan untuk berbagai fungsi seperti PWM, ADC, DAC, I2C, SPI, dan UART, serta fitur low-power mode untuk menghemat daya. Selain itu, sistem ini menggunakan relay module 4 channel dengan tegangan operasional 5V DC dan kapasitas kontak 10A 250V AC atau 10A 30V DC, yang berfungsi mengendalikan saklar untuk lampu dan door lock secara terpisah. *Solenoid door lock* yang digunakan beroperasi pada tegangan 12V DC dengan kapasitas tarikan 2A, yang berfungsi untuk mengunci dan membuka pintu secara elektronik, untuk sumber tegangan pada *solenoid* diperlukan baterai atau aki dengan tegangan 12V DC. Sistem ini diprogram menggunakan *software* Arduino IDE versi 2.2.1 dengan bahasa pemrograman C/C++, serta *library* WiFi.h, *WiFiClientSecure.h*, dan *UniversalTelegramBot.h* untuk mengintegrasikan ESP32 dengan aplikasi Telegram. Dengan spesifikasi alat-alat IoT ini, sistem diharapkan dapat memberikan solusi yang lebih *efisien* dan *responsif* dalam pengelolaan fasilitas yayasan, memungkinkan admin untuk mengontrol lampu dan door lock secara jarak jauh melalui aplikasi Telegram, mengatasi keterbatasan *aksesibilitas* dan ketergantungan pada tugas manual.

2. METODE

Penelitian ini menggunakan metode *Waterfall* dalam merancang sistem ini dikarenakan, metode *Waterfall* ini merupakan metode yang dilakukan secara bertahap dan terurut [18]. **Gambar 1** menggambarkan tahapan metode *Waterfall* yang digunakan dalam penelitian ini.

Analisis Kebutuhan (*Analysis*): Pada tahap analisis ini akan membahas tentang kebutuhan sistem serta akan menjelaskan gambaran awal, komponen apa saja yang dibutuhkan dan spesifikasinya. Penjelasan mengenai kebutuhan sistem dan alat atau komponen yang dibutuhkan untuk sistem ini, akan dibahas di bagian hasil dan pembahasan.

Desain (*Design*): Pada tahap desain ini dibuat perancangan alur sistem yang dengan tujuan untuk memberikan gambaran tentang konsep sistem yang diinginkan serta bagaimana sistem rangkaianannya. Penjelasan mengenai desain dan skema rangkaian alat akan dibahas di bagian hasil dan pembahasan.



Gambar 1. Metode *Waterfall*

Pembuatan kode (Pengkodean): Setelah perancangan selesai selanjutnya melakukan *coding* dengan menggunakan Arduino IDE di tahap ini peneliti mulai mengimplementasikan desain sistem menjadi bentuk kode program. Dalam hal ini, peneliti akan menggunakan *platform* pengembangan Arduino IDE untuk memprogram *mikrokontroler* ESP32 sesuai dengan *spesifikasi* yang telah dirancang. Penjelasan mengenai Pengkodean ini akan dibahas di bagian hasil dan pembahasan.

Pengujian: Setelah selesai melakukan pengkodean, tahap selanjutnya adalah pengujian, tahap ini memastikan bahwa sistem yang dirancang sudah sesuai dengan yang diinginkan tanpa adanya *error* di setiap perintahnya, untuk penjelasan tentang hasil pengujian akan dijelaskan di bagian hasil dan pembahasan.

3. HASIL DAN PEMBAHASAN

Kebutuhan sistem dan komponen yang digunakan

Analisis kebutuhan sistem mencakup *identifikasi* kebutuhan *fungsi*ional dan *nonfungsi*ional untuk sistem pengendalian lampu dan *door lock* berbasis IoT dengan integrasi aplikasi Telegram di Yayasan Gunung Wayang Manunggal. Kebutuhan *fungsi*ional menyelidiki berbagai informasi yang diberikan untuk sistem berdasarkan *identifikasi* masalah yang ada di Yayasan setelah melakukan wawancara dengan admin yayasan dan melakukan *observasi*. Hasil *analisis* ini mengungkapkan dua kebutuhan utama yang harus dipenuhi oleh sistem. Pertama, admin yayasan harus dapat mengontrol masing-masing lampu dan pintu (*door lock*) dari jarak jauh menggunakan aplikasi Telegram. Ini memungkinkan admin untuk menyalakan atau mematikan lampu serta membuka atau mengunci pintu ruangan tanpa perlu hadir secara fisik di yayasan. Kedua, sistem harus memungkinkan admin untuk memantau kondisi masing-masing lampu dan pintu ruangan (*door lock*) secara jarak jauh. Dengan fitur ini, admin dapat mengetahui status lampu apakah menyala atau tidak, serta mengetahui status pintu apakah dalam keadaan tertutup atau terbuka. *Identifikasi* kebutuhan ini didasarkan pada masalah yang sering dihadapi oleh admin seperti keterbatasan aksesibilitas dan ketergantungan pada tugas manual terutama saat admin tidak berada di lokasi yayasan. Kebutuhan *non-fungsi*ional yang diidentifikasi meliputi keandalan sistem untuk memastikan pengendalian perangkat yang tepat dan konsisten, serta kemampuan sistem untuk memberikan respon cepat dengan waktu tunda yang minimal antara saat perintah dikirim dan saat perintah tersebut dijalankan. Komponen yang digunakan dalam sistem ini, yang dirangkum dalam [Tabel 1](#).

Tabel 1. Komponen yang digunakan

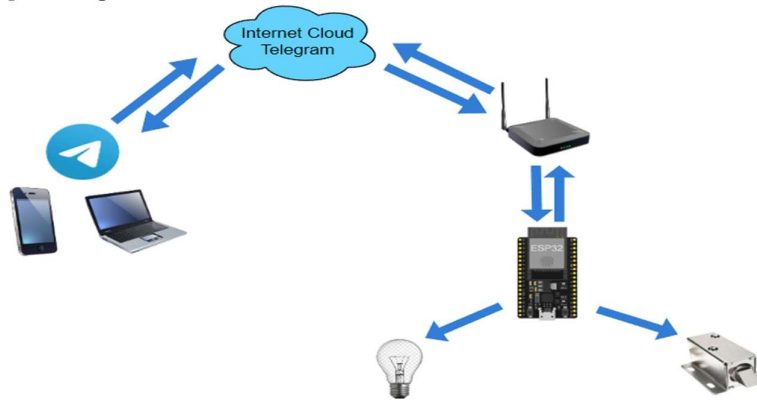
KOMPONEN	SPEKIFIKASI	FUNGSI
ESP32	Prosesor dual-core Tensilica LX6, Wi-Fi 802.11, dan Bluetooth 4.2	Pusat pengendali dan penghubung ke jaringan internet
Relay Module 4 Channel	Tegangan operasional 5V DC, Kapasitas kontak 10A 250V AC atau 10A 30V DC	Mengendalikan saklar untuk lampu dan door lock

Solenoid Door Lock	Tegangan operasional 12V DC, Kapasitas tarikan 2A	Mengunci dan membuka pintu secara elektronik
Baterai/Aki 12V	Tegangan 12V DC	Sumber daya untuk solenoid door lock
Lampu	Sesuai kebutuhan daya lampu	Penerangan ruangan
Kabel Jumper	Panjang dan jenis sesuai kebutuhan	Menghubungkan ESP32 dengan relay module dan komponen lainnya
Software Arduino IDE	Versi 2.2.1	Platform untuk memprogram ESP32
Library ESP32	WiFi.h, WiFiClientSecure.h, UniversalTelegramBot.h	Mengintegrasikan ESP32 dengan aplikasi Telegram

Desain sistem

a. Arsitektur Perancangan Sistem

Tujuan utama perancangan sistem ini adalah untuk mengimplementasikan teknologi *Internet of Things* (IoT) dalam mengendalikan lampu dan kunci pintu (*door lock*) melalui aplikasi Telegram. Sistem ini dirancang agar admin yayasan dapat mengontrol dan memantau kondisi lampu dan pintu dari jarak jauh. *Implementasi* ini diharapkan dapat meningkatkan *efisiensi* pengelolaan fasilitas yayasan, mengatasi keterbatasan *aksesibilitas* serta mengurangi ketergantungan pada tugas manual.

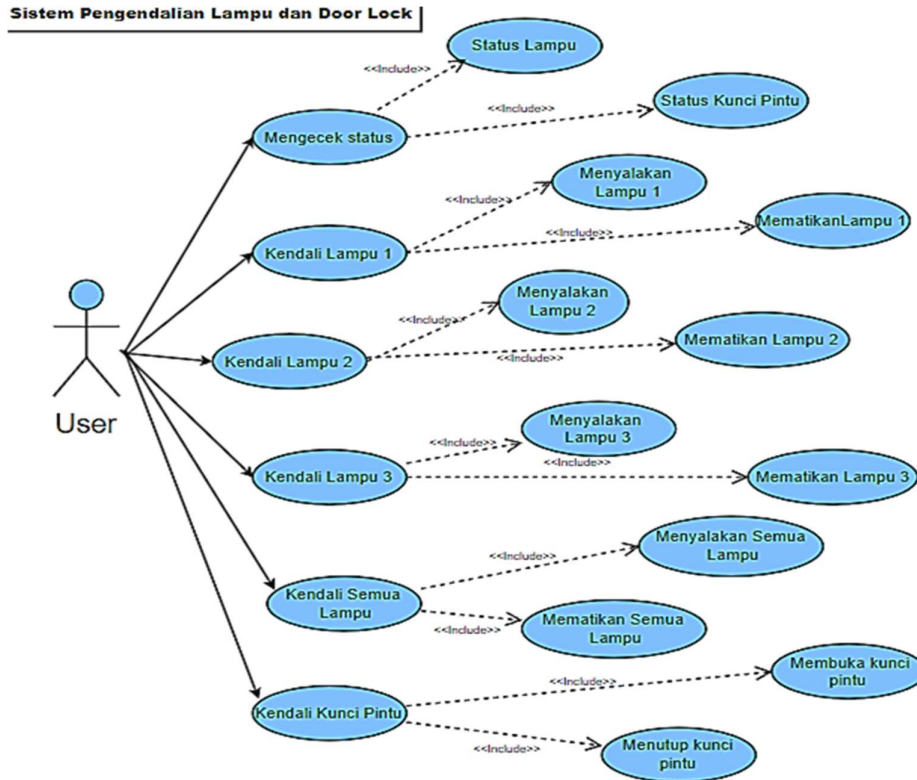


Gambar 2. Arsitektur perancangan sistem

Desain arsitektur sistem yang lebih rinci dapat dilihat pada Gambar 2. Gambar ini menjelaskan bagaimana perangkat Android mengirim perintah melalui Telegram, yang kemudian diproses oleh mikrokontroler ESP32 untuk mengendalikan lampu dan door lock.

b. Rancangan *Use-Case* Diagram

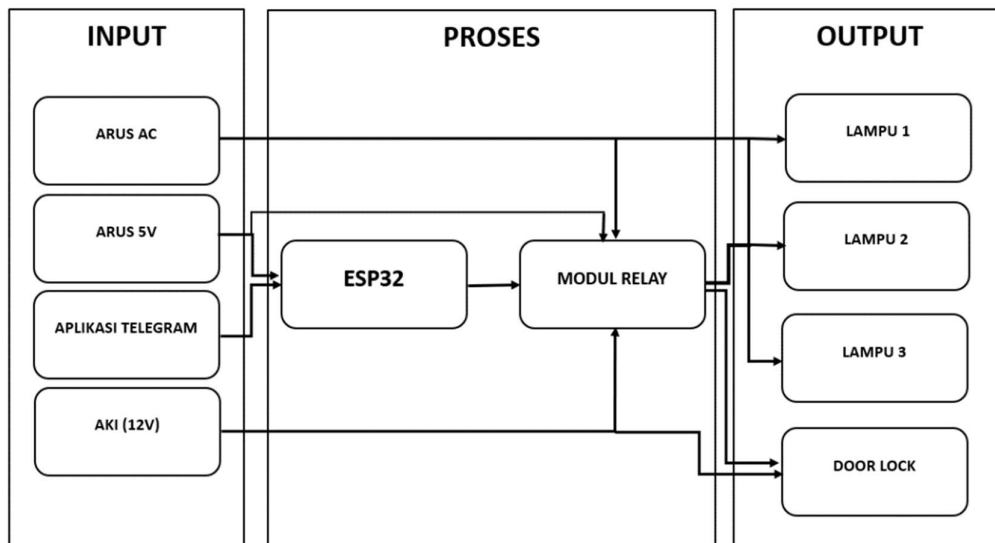
Rancangan ini meliputi admin yayasan yang diperankan sebagai user dalam menjalankan fungsi perintah di dalamnya pada sistem kontrol lampu dan *door lock* jarak jauh tersebut. Gambar 3 menunjukkan diagram yang dirancang pada sistem tersebut.



Gambar 3. Rancangan Use-Case Diagram

Berdasarkan tampilan pada Gambar 3 menggambarkan sistem pengendalian lampu dan kunci pintu yang memungkinkan admin yayasan untuk mengecek status lampu dan kunci pintu, mengendalikan setiap lampu secara *individual* (menghidupkan dan mematikan lampu 1, lampu 2, dan lampu 3), mengendalikan semua lampu secara bersamaan, serta mengendalikan kunci pintu (membuka dan menutup kunci pintu). Diagram ini menunjukkan bahwa setiap aksi kontrol mencakup aksi *spesifik* yang terhubung dengan relasi <<includes>>, yang menandakan bahwa aksi berikut merupakan bagian dari kontrol yang dapat dilakukan oleh admin yayasan.

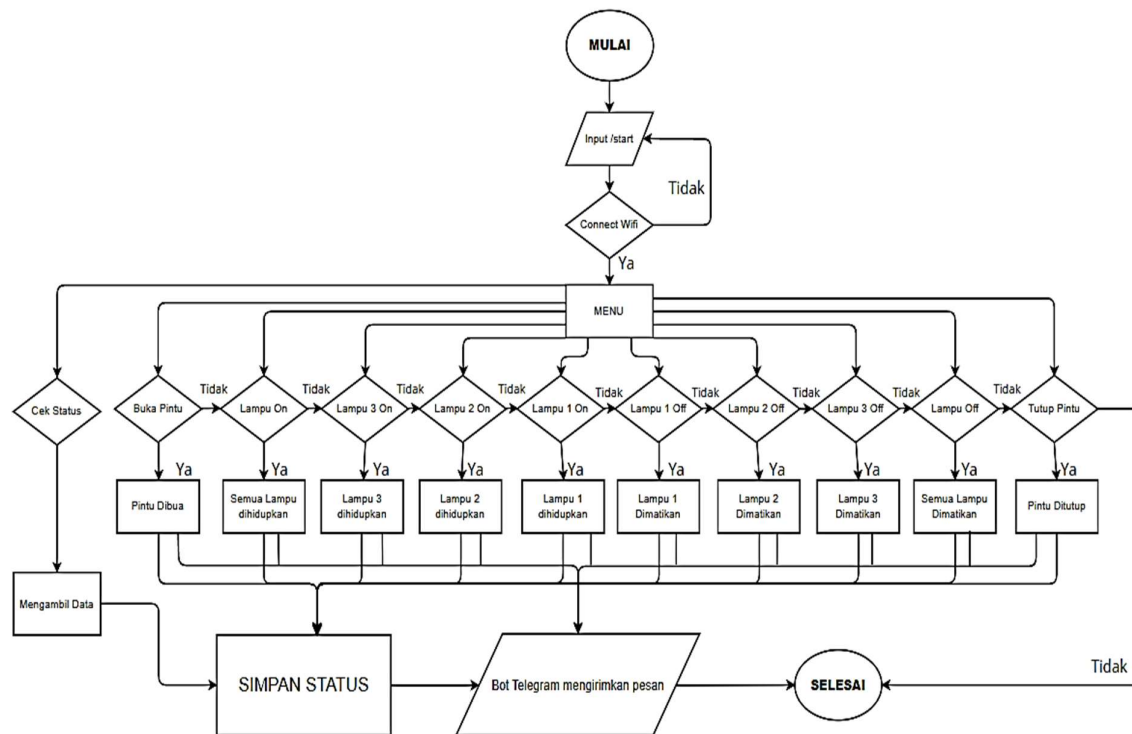
c. Diagram Blok Sistem



Gambar 4. Diagram blok sistem

Gambar 4 menjelaskan bahwa input sebagai komponen yang berfungsi sebagai bahan yang akan diolah, diantaranya arus AC merupakan arus yang diambil langsung dari listrik rumah (dari PLN), yang berfungsi untuk memberi tegangan pada Lampu. Arus 5V yaitu arus yang didapat dari *Adaptor 5V* yang terhubung ke stop kontak, yang digunakan sebagai sumber *power* untuk ESP32 dan *modul relay*. Aplikasi Telegram berfungsi untuk pusat kontrol serta pemantauan status pada alat atau sistem kali ini. AKI (12V) digunakan untuk memberikan tegangan pada *Solenoid Door Lock*. Proses merupakan komponen utama yang mengumpulkan data dari input dan menghasilkan output. Penelitian ini menggunakan ESP32 dengan tambahan Relay dalam alat ini untuk membagi setiap komponen pada alat ini. *Output* merupakan keluaran dari semua proses yang telah dijalankan.

d. Flowchart Sistem



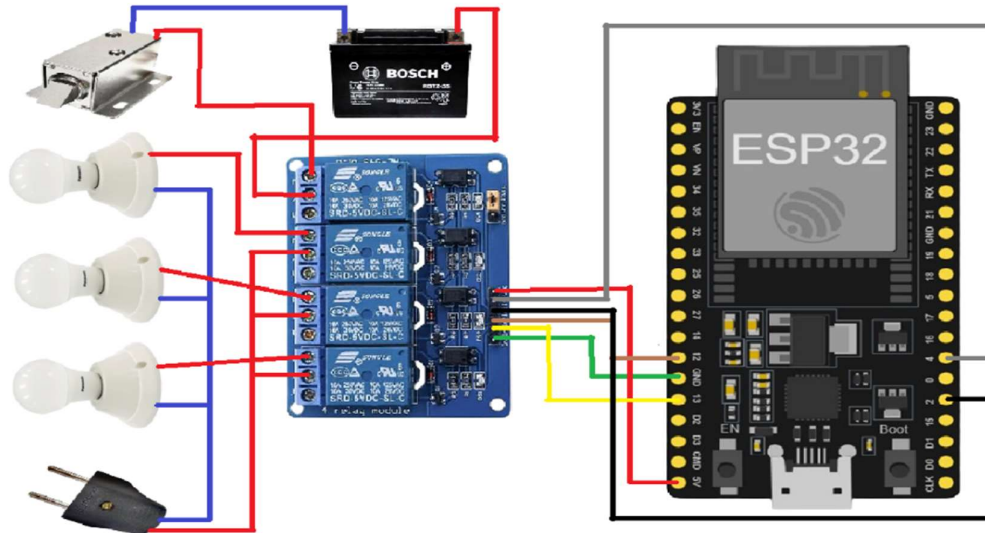
Gambar 5. Flowchart sistem

Penjelasan **Gambar 5** Flowchart Sistem merupakan alur sistem pengendalian lampu dan kunci pintu yang dilakukan oleh admin dimulai dengan input perintah */start* untuk menghubungkan sistem ke WiFi. Setelah koneksi berhasil, admin masuk ke menu utama dengan berbagai pilihan kontrol. Admin dapat memilih untuk mengecek status, yang akan mengambil data status tersimpan dan mengirimkan pesan ke admin melalui bot telegram dengan menampilkan kondisi semua lampu dan pintu. Jika admin memilih kontrol lain seperti menyalakan atau mematikan lampu (Lampu 1, Lampu 2, Lampu 3, atau semua lampu) atau membuka dan menutup pintu, sistem akan melakukan aksi tersebut dan kemudian mengirim pesan konfirmasi ke admin. Misalnya, jika admin memilih untuk menyalakan Lampu 1, sistem akan menyalakan Lampu 1 dan bot Telegram akan mengirimkan pesan "Lampu 1 dihidupkan" kepada admin. Setiap aksi kontrol langsung disimpan statusnya untuk memastikan semua perubahan tercatat dan admin selalu mendapatkan informasi tentang status terbaru, disaat admin ingin mengecek statusnya.

e. Perancangan alat

Gambar 6 perancangan alat merupakan pengendalian lampu dan *door lock* yang menggunakan *mikrokontroler* ESP32 yang terhubung dengan modul *relay* untuk mengontrol tiga lampu dan sebuah kunci pintu *elektronik*. Penjelasan Jalur port antara ESP32 dan *Relay Module*

diatur sebagai berikut GND dari ESP32 terhubung ke pin GND pada relay untuk *ground connection*, sementara 5V dari ESP32 terhubung ke pin VCC pada *relay* untuk suplai daya. Sinyal kontrol dari ESP32, yaitu D13, D12, D2, dan D4, masing-masing terhubung ke pin IN1, IN2, IN3, dan IN4 pada *relay* untuk mengontrol *switch relay*.

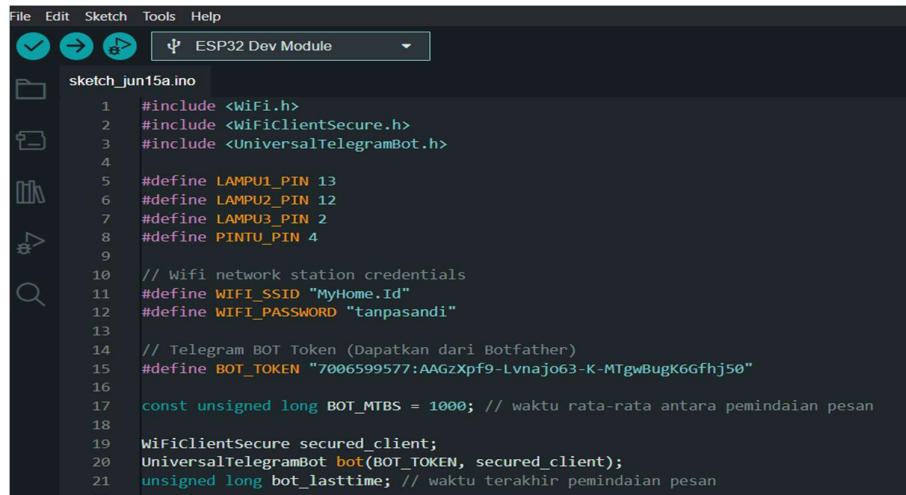


Gambar 6. Perancangan alat

Relay Module selanjutnya terhubung ke perangkat seperti lampu dan *solenoid door lock* untuk lampu, kabel *positif* (merah) dari stop kontak terhubung ke pin K1, K2, dan K3 pada *relay* dan diteruskan ke masing-masing lampu, sementara kabel *negatif* (biru) dari stop kontak langsung terhubung ke masing-masing lampu. Untuk *solenoid door lock*, kabel *positif* (merah) dari baterai terhubung ke pin K4 pada relay dan diteruskan ke *solenoid* (kabel merah), sementara kabel *negatif* (biru) dari baterai langsung terhubung ke *solenoid* (kabel biru). Adaptor daya terhubung ke port USB pada ESP32 dengan tegangan minimal. Dengan konfigurasi ini, ESP32 dapat mengontrol pengaktifan dan pemutusan arus listrik pada lampu dan solenoid door lock menggunakan relay sebagai perantara kontrol.

Pengkodean

Saat ini Arduino IDE yang digunakan untuk pemrograman ESP32 adalah versi 2.2.1. Versi ini dipilih karena mendukung berbagai *library* dan fitur terbaru yang dibutuhkan untuk mengintegrasikan ESP32 dengan aplikasi Telegram. Untuk memulai, unduh Arduino IDE versi 2.2.1 dari situs resmi Arduino, dan pilih versi yang sesuai dengan sistem operasi yang digunakan (Windows, macOS, atau Linux). Setelah mengunduh, ikuti petunjuk instalasi yang diberikan oleh *installer* untuk menyelesaikan proses instalasi. Setelah *instalasi* selesai, buka Arduino IDE dan buka menu *File > Preferences*. Tambahkan URL berikut pada *Additional Board Manager URLs*: https://dl.espressif.com/dl/package_esp32_index.json, lalu klik *OK* untuk menyimpan pengaturan. Selanjutnya, buka menu *Tools > Board > Boards Manager*, ketik "ESP32" pada kolom pencarian, temukan "esp32 by Espressif Systems", dan klik *Install*. Gunakan kabel data USB untuk menghubungkan ESP32 ke PC atau laptop. Buka menu *Tools > Port* dan pilih port yang sesuai dengan ESP32 yang terhubung (contoh: COM3 atau /dev/ttyUSB0). Pilih board ESP32 dengan membuka menu *Tools > Board* dan pilih "ESP32 Dev Module". Tulis atau buka kode yang ingin diunggah ke ESP32 pada Arduino IDE, klik tombol *Verify* (centang) untuk memeriksa kode, dan jika tidak ada kesalahan, klik tombol *Upload* (panah kanan) untuk mengunggah kode ke ESP32. Pastikan *driver USB* yang sesuai dengan ESP32 telah terinstal pada PC atau laptop untuk memastikan koneksi yang stabil. Dengan langkah-langkah ini, Arduino IDE akan siap digunakan untuk memprogram ESP32, setelah sudah melakukan langkah Langkah awal berikut selanjutnya bisa untuk melakukan pengkodean pada ESP3.



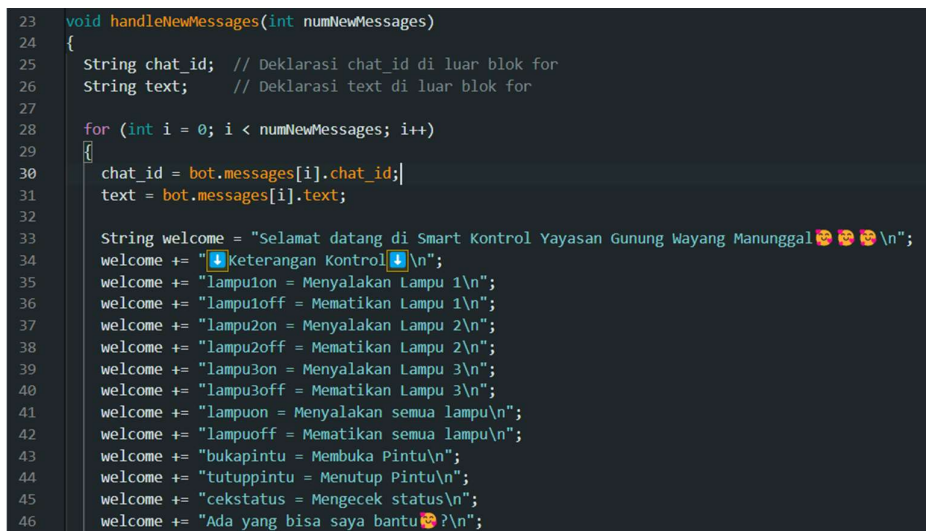
```

File Edit Sketch Tools Help
ESP32 Dev Module
sketch_jun15a.ino
1 #include <WiFi.h>
2 #include <WiFiClientSecure.h>
3 #include <UniversalTelegramBot.h>
4
5 #define LAMPU1_PIN 13
6 #define LAMPU2_PIN 12
7 #define LAMPU3_PIN 2
8 #define PINTU_PIN 4
9
10 // Wifi network station credentials
11 #define WIFI_SSID "MyHome.Id"
12 #define WIFI_PASSWORD "tanpasandi"
13
14 // Telegram BOT Token (Dapatkan dari Botfather)
15 #define BOT_TOKEN "7006599577:AAGzXpf9-LvnaJe63-K-MTgwBugK6Gfhj50"
16
17 const unsigned long BOT_MTBS = 1000; // waktu rata-rata antara pemindaian pesan
18
19 WiFiClientSecure secured_client;
20 UniversalTelegramBot bot(BOT_TOKEN, secured_client);
21 unsigned long bot_lasttime; // waktu terakhir pemindaian pesan

```

Gambar 7. Program 1

Berdasarkan tampilan pada Gambar 7 fungsi dari `WiFi.h` dan `WiFiClientSecure.h` digunakan sebagai *library* untuk menghubungkan ESP32 ke WiFi untuk menggunakan koneksi yang aman. `UniversalTelegramBot.h` merupakan *library* untuk berkomunikasi dengan bot Telegram. Sedangkan `#define LAMPU1_PIN`, `LAMPU2_PIN`, `LAMPU3_PIN`, `PINTU_PIN` untuk menetapkan nomor pin pada ESP32 yang terhubung dengan masing masing lampu dan pintu. Untuk bagian `WIFI_SSID` dan `WIFI_PASSWORD` disesuaikan dengan wifi yang akan digunakan, sedangkan `BOT_TOKEN` merupakan Token Telegram Bot yang didapat dari Botfather, selanjutnya token yang didapat dari Botfather ditempel pada bagian tersebut untuk di integrasikan ke ESP32, agar Aplikasi Telegram dapat digunakan untuk pemberian perintah atau pusat kontrol pada sistem ini.



```

23 void handleNewMessages(int numNewMessages)
24 {
25     String chat_id; // Deklarasi chat_id di luar blok for
26     String text; // Deklarasi text di luar blok for
27
28     for (int i = 0; i < numNewMessages; i++)
29     {
30         chat_id = bot.messages[i].chat_id;
31         text = bot.messages[i].text;
32
33         String welcome = "Selamat datang di Smart Kontrol Yayasan Gunung Wayang Manunggal 🤖🤖🤖\n";
34         welcome += "📄Keterangan Kontrol📄\n";
35         welcome += "lampu1on = Menyalakan Lampu 1\n";
36         welcome += "lampu1off = Mematikan Lampu 1\n";
37         welcome += "lampu2on = Menyalakan Lampu 2\n";
38         welcome += "lampu2off = Mematikan Lampu 2\n";
39         welcome += "lampu3on = Menyalakan Lampu 3\n";
40         welcome += "lampu3off = Mematikan Lampu 3\n";
41         welcome += "lampuon = Menyalakan semua lampu\n";
42         welcome += "lampuoff = Mematikan semua lampu\n";
43         welcome += "bukapintu = Membuka Pintu\n";
44         welcome += "tutuppintu = Menutup Pintu\n";
45         welcome += "cekstatus = Mengecek status\n";
46         welcome += "Ada yang bisa saya bantu 🤖?\n";

```

Gambar 8. Program 2

Gambar 8 merupakan pengkodean dan pembuatan keterangan kontrol pada sistem pengendalian lampu dan door lock yang ada di yayasan. dengan menggunakan fungsi `handleNewMessages` untuk mengelola pesan-pesan baru yang diterima. Dengan menerima jumlah `numNewMessages` sebagai parameter, fungsi ini iterasi melalui setiap pesan baru untuk mengambil `chat_id` dan `text`. Selain itu, fungsi ini menyusun pesan selamat datang yang menjelaskan berbagai perintah keterangan kontrol yang dapat digunakan dalam sistem, seperti

mengontrol lampu, pintu, dan memeriksa status. Ini membantu admin dalam memahami pengontrolan sistem dengan lebih baik.

```
if (text == "/start")
{
    bot.sendMessage(chat_id, welcome);
}
else if (text == "lampu1on")
{
    digitalWrite(LAMPU1_PIN, LOW);
    String welcome = "Lampu 1 dihidupkan.\n";
    bot.sendMessage(chat_id, welcome);
}
else if (text == "lampu1off")
{
    digitalWrite(LAMPU1_PIN, HIGH);
    String welcome = "Lampu 1 dimatikan.\n";
    bot.sendMessage(chat_id, welcome);
}
else if (text == "lampu2on")
{
    digitalWrite(LAMPU2_PIN, LOW);
    String welcome = "Lampu 2 dihidupkan.\n";
    bot.sendMessage(chat_id, welcome);
}
else if (text == "lampu2off")
{
    digitalWrite(LAMPU2_PIN, HIGH);
    String welcome = "Lampu 2 dimatikan.\n";
    bot.sendMessage(chat_id, welcome);
}
else if (text == "lampu3on")
{
    digitalWrite(LAMPU3_PIN, LOW);
    String welcome = "Lampu 3 dihidupkan.\n";
    bot.sendMessage(chat_id, welcome);
}
}
```

Gambar 9. Program 3

Gambar 9 merupakan pembuatan kontrol pada masing masing alat sekaligus monitoring melalui bot Telegram menggunakan ESP32. Saat pengguna mengirimkan pesan, program memprosesnya dengan memeriksa nilai dari variabel *text*. Jika *text* adalah *"/start"*, bot akan mengirimkan pesan yang berisi keterangan kontrol. Setiap perintah, seperti *"lampu1on"* untuk menhidupkan Lampu 1 atau *"lampu1off"* untuk mematikannya, mengatur status pin GPIO yang sesuai (*LAMPU1_PIN*) dengan menggunakan *digitalWrite*. Setelah mengubah status pin, program memberikan umpan balik melalui bot telegram dengan mengirimkan pesan yang sesuai, seperti *"Lampu 1 dihidupkan"* atau *"Lampu 1 dimatikan"*. Proses ini berlaku juga untuk mengendalikan Lampu 2 (*LAMPU2_PIN*) dan Lampu 3 (*LAMPU3_PIN*). Ini memungkinkan pengguna untuk mengontrol dari jarak jauh melalui aplikasi telegram.

```
else if (text == "lampu3off")
{
    digitalWrite(LAMPU3_PIN, HIGH);
    String welcome = "Lampu 3 dimatikan.\n";
    bot.sendMessage(chat_id, welcome);
}
else if (text == "bukapintu")
{
    digitalWrite(PINTU_PIN, LOW);
    String welcome = "Pintu dibuka.\n";
    bot.sendMessage(chat_id, welcome);
}
else if (text == "tutuppintu")
{
    digitalWrite(PINTU_PIN, HIGH);
    String welcome = "Pintu ditutup.\n";
    bot.sendMessage(chat_id, welcome);
}
else if (text == "lampuon")
{
    digitalWrite(LAMPU1_PIN, LOW);
    digitalWrite(LAMPU2_PIN, LOW);
    digitalWrite(LAMPU3_PIN, LOW);

    String welcome = "Semua lampu dihidupkan.\n";
    bot.sendMessage(chat_id, welcome);
}
else if (text == "lampuoff")
{
    digitalWrite(LAMPU1_PIN, HIGH);
    digitalWrite(LAMPU2_PIN, HIGH);
    digitalWrite(LAMPU3_PIN, HIGH);

    String welcome = "Semua lampu dimatikan.\n";
    bot.sendMessage(chat_id, welcome);
}
}
```

Gambar 10. Program 4

Untuk penjelasan dari Gambar 10 masih sama dengan yang ada di gambar 9 seperti *"/start"* untuk memulai atau *"lampu1on"* untuk menyalakan Lampu 1, diidentifikasi berdasarkan isi pesan menggunakan struktur *if-else if*. Ketika pengguna mengirimkan perintah untuk mengendalikan lampu atau pintu, fungsi ini menggunakan *digitalWrite* untuk mengubah status pin GPIO yang terhubung ke perangkat fisik, contohnya seperti mengatur pin menjadi *LOW*

untuk menyalakan lampu atau membuka pintu. Selain itu, fungsi ini juga mengirimkan balasan kepada pengguna melalui bot telegram dengan pesan yang sesuai, seperti konfirmasi bahwa lampu telah dinyalakan atau pintu telah dibuka.

```

else if (text == "cekstatus")
{
    String statusMessage = "Status Sekarang:\n";
    statusMessage += "Lampu 1: " + String(digitalRead(LAMPUI_PIN) == LOW ? "ON" : "OFF") + "\n";
    statusMessage += "Lampu 2: " + String(digitalRead(LAMPU2_PIN) == LOW ? "ON" : "OFF") + "\n";
    statusMessage += "Lampu 3: " + String(digitalRead(LAMPU3_PIN) == LOW ? "ON" : "OFF") + "\n";
    statusMessage += "Pintu: " + String(digitalRead(PINTU_PIN) == LOW ? "BUKA" : "TUTUP") + "\n";

    bot.sendMessage(chat_id, statusMessage);
}
}

void setup()
{
    Serial.begin(115200);
    Serial.println();
    pinMode(LAMPUI_PIN, OUTPUT);
    pinMode(LAMPU2_PIN, OUTPUT);
    pinMode(LAMPU3_PIN, OUTPUT);
    pinMode(PINTU_PIN, OUTPUT);

    digitalWrite(LAMPUI_PIN, HIGH);
    digitalWrite(LAMPU2_PIN, HIGH);
    digitalWrite(LAMPU3_PIN, HIGH);
    digitalWrite(PINTU_PIN, HIGH);
}

```

Gambar 11. Program 5

Untuk Penjelasan dari Gambar 11 Bagian else if pada kode tersebut berfungsi untuk menangani pesan yang memiliki teks "cekstatus". Ketika pesan ini diterima, sistem akan merespons dengan pesan yang berisi status terkini dari komponen-komponen yang terhubung, yaitu Lampu 1, Lampu 2, Lampu 3, dan Pintu (*door lock*). Setiap status diambil menggunakan `digitalRead` untuk membaca kondisi pin yang terhubung. Misalnya, untuk setiap lampu, kondisi *LOW* menunjukkan lampu menyala ("ON"), sedangkan *HIGH* menunjukkan lampu mati ("OFF"). Untuk pintu, kondisi *LOW* menunjukkan pintu terbuka ("BUKA"), dan *HIGH* menunjukkan pintu tertutup ("TUTUP"). Setelah mengambil status dari setiap komponen, pesan yang berisi informasi ini dikirim kembali ke pengguna menggunakan `bot.sendMessage` dengan `chat_id` yang sesuai. Sedangkan untuk fungsi setup di bagian bawah kode digunakan untuk inisialisasi awal sistem saat perangkat dinyalakan. Ini mencakup pengaturan serial untuk komunikasi debug, pengaturan mode pin untuk masing-masing lampu dan pintu sebagai *OUTPUT*, serta kondisi awal di mana semua lampu dan pintu dinyalakan dengan menetapkan *HIGH* pada masing-masing pin.

```

// Mencoba terhubung ke jaringan Wifi:
Serial.print("Menghubungkan ke Wifi SSID ");
Serial.print(WIFI_SSID);
WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
securedClient.setCACert(TELEGRAM_CERTIFICATE_ROOT); // Menambahkan sertifikat root untuk api.telegram.org
while (WiFi.status() != WL_CONNECTED)
{
    Serial.print(".");
    delay(500);
}
Serial.print("\nWifi terhubung. Alamat IP: ");
Serial.println(WiFi.localIP());

Serial.print("Mendapatkan waktu: ");
configTime(0, 0, "pool.ntp.org"); // mendapatkan waktu UTC melalui NTP
time_t now = time(nullptr);
while (now < 24 * 3600)
{
    Serial.print(".");
    delay(100);
    now = time(nullptr);
}
Serial.println(now);

void loop()
{
    if (millis() - bot_lasttime > BOT_MTBS)
    {
        int numNewMessages = bot.getUpdates(bot.last_message_received + 1);
        while (numNewMessages)
        {
            Serial.println("mendapatkan respons");
            handleNewMessages(numNewMessages);
            numNewMessages = bot.getUpdates(bot.last_message_received + 1);
        }
        bot_lasttime = millis();
    }
}

```

Gambar 12. Program 6

Gambar 12 berfungsi menghubungkan perangkat ke jaringan WiFi, untuk mendapatkan waktu dari server NTP, dan menangani pesan baru dari bot Telegram secara berkala. Bagian pertama mencoba menghubungkan ke jaringan WiFi dengan menampilkan SSID yang digunakan dan memulai koneksi menggunakan `WiFi.begin(WIFI_SSID, WIFI_PASSWORD)`. Selama proses ini, titik-titik dicetak di *serial monitor* untuk menunjukkan upaya ESP32 mencoba menghubungkan ke WiFi, setelah *serial monitor* berubah menjadi terhubung (`WL_CONNECTED`) dan alamat IP lokal perangkat ditampilkan maka dapat disimpulkan bahwa ESP32 sudah terhubung ke internet dan sistem siap dijalankan sesuai dengan perintah yang dikirimkan oleh *user* atau admin melalui aplikasi telegram.



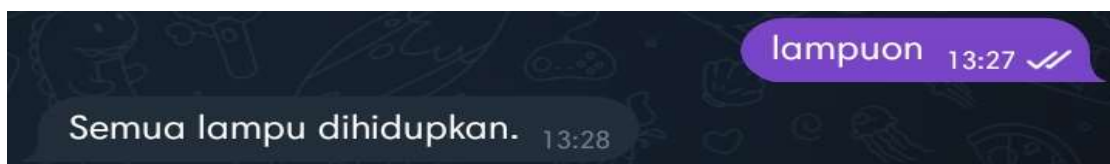
Gambar 13. Tampilan telegram bot

Gambar 13 merupakan tampilan awal pada telegram bot setelah ESP32 sudah terhubung ke internet dan *user* mengirimkan *command* `/start` maka bot telegram akan mengirimkan pesan balasan secara otomatis dan akan keluar "Keterangan Kontrol" pada perangkat yang digunakan *user*. Aksi diatas akan dapat terjadi dengan syarat jika sebelumnya ESP32 sudah terkoneksi ke Internet. Dan jika bot telegram langsung merespon ketika *user* mengirimkan *command* sesuai dengan perintah yang sudah ditentukan maka dipastikan ESP32 sudah terkoneksi ke internet, begitu pula sebaliknya jika bot telegram tidak merespon atau tidak mengirimkan pesan balasan ketika *user* mengirimkan *command* sebaiknya diperhatikan apakah perintah yang dikirimkan ke Bot Telegram sudah benar atau belum, jika perintah atau pesan yang dikirimkan ke bot telegram sudah benar akan tetapi masih belum memberikan pesan balasan maka dapat disimpulkan ESP32 belum terhubung ke Internet, atau adanya gangguan Internet pada perangkat yang digunakan oleh admin atau *user*.

Pengujian sistem

a. Pengujian Menyalakan Semua Lampu

Dalam pengujian ini admin atau *user* mengirimkan perintah ke bot telegram untuk menyalakan semua lampu, dengan mengirimkan *command* atau perintah ke bot telegram yang berupa `lampuon` yang sesuai dengan parameter yang sudah diprogram di ESP32.



Gambar 14. Menyalakan semua lampu

Pada **Gambar 14** maka dapat dilihat, bahwa bot telegram berhasil mengirimkan Pesan balasan artinya ESP32 sudah terkoneksi dengan Internet maka dapat disimpulkan ESP32

berhasil menyalakan semua lampu sesuai perintah yang dikirimkan oleh admin atau *user* melalui perangkat yang digunakan.

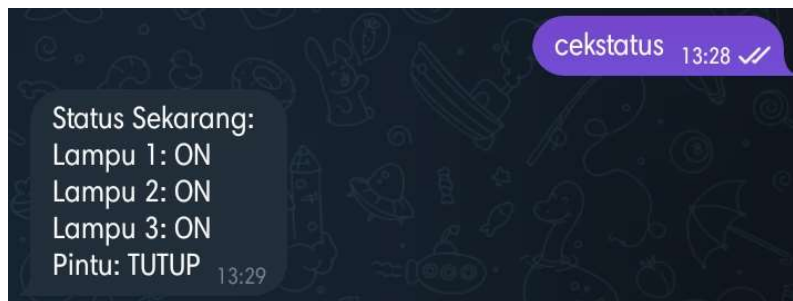


Gambar 15. Semua lampu menyala

Pada Gambar 15 dapat dilihat bahwa semua lampu berhasil dinyalakan dan komponen yang tidak mendapat perintah tidak akan berpengaruh atau dengan kata lain masih dalam kondisi sebelumnya.

b. Pengujian Mengecek Status

Dalam pengujian ini admin atau *user* mengirimkan perintah ke bot telegram untuk mengecek status pada *door lock* dan masing-masing lampu, dengan mengirimkan *command* atau perintah ke Bot Telegram yang berupa “cekstatus” yang sesuai dengan parameter yang sudah diprogram di ESP32.



Gambar 16. Tampilan hasil cek status

Pada Gambar 16 dapat dilihat bahwa Bot Telegram Berhasil mengirimkan pesan balasan dengan menampilkan status saat ini dengan hasil lampu 1, 2 dan 3 dalam keadaan menyala (*ON*) sedangkan dapat dilihat bahwa pintu (*Door lock*) masih dalam keadaan menutup sesuai dengan kondisi sebelumnya.

c. Pengujian Keseluruhan

Dalam pengujian ini admin atau *user* mengirimkan perintah ke Bot Telegram untuk memastikan masing-masing lampu dan door lock berjalan sesuai parameter yang dikirim oleh admin atau *user* ke Bot Telegram.

Tabel 2. Hasil pengujian keseluruhan

PARAMETER	PENGUJIAN KE	NOTIFIKASI DARI BOT TELEGRAM	HASIL
lampu1on	LAMPU 1 (Memberi tegangan pada Relay IN1)	Lampu 1 dihidupkan	Berhasil menyalakan lampu 1

PARAMETER	PENGUJIAN KE	NOTIFIKASI DARI BOT TELEGRAM	HASIL
lampu1off	LAMPU 1 (Memutus tegangan pada Relay IN1)	Lampu 1 dimatikan	Berhasil mematikan lampu 1
lampu2on	LAMPU 2 (Memberi tegangan pada Relay IN2)	Lampu 2 dihidupkan	Berhasil menyalakan lampu 2
lampu2off	LAMPU 2 (Memutus tegangan pada Relay IN2)	Lampu 2 dimatikan	Berhasil mematikan lampu 2
lampu3on	LAMPU 3 (Memberi tegangan pada Relay IN3)	Lampu 3 dihidupkan	Berhasil menyalakan lampu 3
lampu3off	LAMPU 3 (Memutus tegangan pada Relay IN3)	Lampu 3 dimatikan	Berhasil mematikan lampu 3
lampuon	LAMPU 1 2 3 (Memberi tegangan pada Relay IN1 2 dan 3)	Semua lampu dihidupkan	Berhasil menyalakan semua lampu (lampu 1 2 dan 3)
lampuoff	LAMPU 1 2 3 (Memutus tegangan pada Relay IN1 2 dan 3)	Semua lampu dimatikan	Berhasil mematikan semua lampu (lampu 1 2 dan 3)
bukapintu	SOLENOID (Memberi tegangan pada Relay IN4)	Pintu dibuka	Pintu Berhasil dibuka
tutuppintu	SOLENOID (Memutus tegangan pada Relay IN4)	Pintu ditutup	Pintu Berhasil ditutup

Pengujian keseluruhan dilakukan dengan mengirimkan perintah *spesifik* ke Bot Telegram untuk memastikan setiap fungsi dari sistem berjalan dengan baik sesuai dengan *parameter* yang telah diprogram pada ESP32. Hasil pengujian dicatat dalam [Tabel 2](#) yang menunjukkan bahwa semua perintah berhasil dijalankan tanpa adanya *error*. Dari hasil pengujian yang dilakukan sebanyak 10 kali, sistem menunjukkan tingkat keberhasilan 100% dalam menyalakan dan mematikan lampu serta membuka dan menutup *door lock*. Hal ini menunjukkan bahwa sistem yang dirancang mampu berfungsi sesuai dengan *spesifikasi* yang diinginkan. Pengujian menunjukkan bahwa dengan menggunakan Bot Telegram, admin dapat mengendalikan lampu dan *door lock* secara jarak jauh dengan *respons* yang cepat. Hal ini mampu meningkatkan *efisiensi kontrol* fasilitas yayasan, karena admin tidak perlu lagi hadir secara fisik untuk melakukan pengendalian tersebut. Hasil pengujian yang menunjukkan keberhasilan 100% dalam menjalankan setiap perintah tanpa adanya *error* mencerminkan keandalan sistem yang tinggi. Sistem ini dapat diandalkan untuk mengontrol fasilitas yayasan secara *konsisten* dan tepat. Admin yayasan memberikan respon positif terhadap sistem ini, menyatakan bahwa pengendalian lampu dan *door lock* menjadi lebih *efisien* dan mudah. Peminjam ruangan juga merasakan manfaat dari sistem ini karena mereka bisa mendapatkan akses fasilitas yayasan dengan lebih cepat. Kesimpulan dari pengujian keseluruhan membuktikan bahwa sistem pengendalian lampu dan *door lock* berbasis IoT yang dikendalikan melalui Bot Telegram efektif dalam meningkatkan *efisiensi kontrol* dan mengurangi ketergantungan pada tugas manual. Tingkat keberhasilan 100% dalam pengujian menunjukkan bahwa sistem ini siap digunakan untuk kebutuhan *operasional* yayasan sehari-hari.

4. SIMPULAN

Sistem pengendalian lampu dan *door lock* jarak jauh ini menggunakan metode *Waterfall*. Pengujian dengan mengirimkan 10 perintah ke Bot Telegram menunjukkan bahwa sistem

berhasil mengontrol semua alat tanpa kesalahan, dengan tingkat akurasi 100%. Ini membuktikan bahwa sistem ini *efektif* dalam meningkatkan *efisiensi kontrol*, menghemat waktu, dan mengurangi ketergantungan pada tugas manual admin yayasan. Sistem ini mempermudah admin dalam mengontrol lampu dan kunci pintu ruangan dari jarak jauh menggunakan aplikasi Telegram di HP, tanpa perlu datang ke yayasan, jadi jika ada yang ingin meminjam ruangan di yayasan sedangkan admin sedang berada di luar yayasan, admin tidak perlu lagi ke yayasan untuk membuka kunci ruangan (*door lock*) dan menyalakan lampu secara manual atau kembali ke yayasan. Admin memberikan respon *positif*, menyatakan bahwa sistem ini sangat membantu dalam pengendalian lampu dan *door lock* mereka karena lebih *efisien* dan pengoprasiannya juga mudah dikarenakan bisa menggunakan HP. Selain itu dengan adanya pengendalian lampu dan *door lock* jarak ini, peminjam ruangan jadi lebih cepat untuk mendapatkan akses fasilitas di yayasan.

REFERENSI

- [1] D. Sawitri, "Internet Of Things Memasuki Era Society5.0," *KITEKTRO J. Komputer, Inf. Teknol. dan Elektro*, vol. 8, no. 1, pp. 31–35, 2023.
- [2] V. N. Juli, A. Jl, R. Jakarta, K. Serang, and P. Banten, "Mengamati Perkembangan Teknologi dan Bisnis Digital dalam Transisi Menuju Era Industri 5.0," no. 3, 2024.
- [3] F. Susanto, N. K. Prasiani, and P. Darmawan, "Implementasi Internet of Things Dalam Kehidupan Sehari-Hari," *J. Imagine*, vol. 2, no. 1, pp. 35–40, 2022, doi: 10.35886/imagine.v2i1.329.
- [4] I. N. B. Hartawan and I. W. Sudiarsa, "Analisis Kinerja Internet of Things Berbasis Firebase Real-Time Database," *J. Resist. (Rekayasa Sist. Komputer)*, vol. 2, no. 1, pp. 6–17, 2019, doi: 10.31598/jurnalresistor.v2i1.371.
- [5] M. Muhammad Nasir, "Pemanfaatan Google Form Untuk Pelaporan Form A di Panwascam Daha Selatan: Inovasi Teknologi dalam Peningkatan Efisiensi dan Akurasi Pengawasan Pemilu," *Pendidik. Sos. dan Hum.*, vol. 2, no. 3, pp. 10111–10121, 2023, [Online]. Available: <https://publisherqu.com/index.php/pediaqu>
- [6] M. Yusuf, M. Sodik, S. Darussalam, K. Nganjuk, and U. Blitar, "Penggunaan Teknologi Internet of Things (Iot) Dalam Pengelolaan Fasilitas Dan Infrastruktur Lembaga Pendidikan Islam," *Prophet. J. Kaji. Keislam.*, vol. 1, no. 2, pp. 1–18, 2023.
- [7] I. K. Wijayanti, Nurchim, and J. Maulindar, "Perancangan Smart Home Jemuran Otomatis Berbasis Internet of Things," *INFOTECH J.*, vol. 9, no. 1, pp. 183–189, 2023, doi: 10.31949/infotech.v9i1.5344.
- [8] M. Nigel, B. Rahmat, and S. Informatika, "DENGAN METODE IT2FL," vol. 7, no. 1, pp. 161–173, 2024.
- [9] A. Subiyantoro and Listyaningsih, "Pengembangan Media Pembelajaran Mobile Learning dengan BOT API Aplikasi Telegram Pada Mata Pelajaran PPKn di SMAN 12 Jakarta," *Kaji. Moral dan Kewarganegaraan*, vol. 08, no. 3, p. 15, 2020, [Online]. Available: <https://ejournal.unesa.ac.id/index.php/jurnal-pendidikan-kewarganegaraa/article/view/36178/32207>
- [10] A. M. Ibrahim and D. Setiyadi, "Prototype Pengendalian Lampu Dan Ac Jarak Jauh Dengan Jaringan Internet Menggunakan Aplikasi Telegram Berbasis Nodemcu Esp8266," *Infotech J. Technol. Inf.*, vol. 7, no. 1, pp. 27–34, 2021, doi: 10.37365/jti.v7i1.103.
- [11] A. Sanaris and I. Suharjo, "Prototype Alat Kendali Otomatis Penjemur Pakaian Menggunakan NodeMCU ESP32 Dan Telegram Bot Berbasis Internet of Things (IOT)," *J. Prodi Sist. Inf.*, no. 84, pp. 17–24, 2020.
- [12] C. S. Ningsih and A. F. Juwito, "Pengendalian Lampu Berbasis Android," *J. Appl. Sci. Electr. Eng. Comput. Technol.*, vol. 2, no. 1, pp. 15–23, 2021, doi: 10.30871/aseect.v2i1.2895.
- [13] R. Tullah, S. M. Mustafa, and D. E. A. Nugraha, "Sistem Keamanan Rumah Berbasis Mikrokontroler Arduino dan SMS Gateway," *Acad. J. Comput. Sci. Res.*, vol. 1, no. 1, 2019, doi: 10.38101/ajcsr.v1i1.232.

- [14] Mariza Wijayanti, "Prototype Smart Home Dengan Nodemcu Esp8266 Berbasis Iot," *J. Ilm. Tek.*, vol. 1, no. 2, pp. 101–107, 2022, doi: 10.56127/juit.v1i2.169.
- [15] D. Cakra, M. Wijaya, and H. Khariono, "JIP (Jurnal Informatika Polinema) PEMANTAUAN PH BERBASIS NODEMCU32 TERINTEGRASI BOT TELEGRAM MELALUI PLATFORM I-OT.NET," *J. Inform. Polinema*, vol. 8, pp. 53–62, 2022.
- [16] A. Fakhruddin *et al.*, "RANCANG BANGUN SISTEM KEAMANAN PINTU RUMAH BERBASIS INTERNET OF THINGS DENGAN ESP32 DAN APLIKASI BLYNK," vol. 19, pp. 53–59, 2024.
- [17] A. D. Mulyanto, "Pemanfaatan Bot Telegram Untuk Media Informasi Penelitian," *Matics*, vol. 12, no. 1, p. 49, 2020, doi: 10.18860/mat.v12i1.8847.
- [18] F. Ali Akbar, B. Nugroho, and A. Sri Indrawanti, "Perancangan Sistem Monitoring Dana Bantuan Untuk Petani Dengan Menggunakan Waterfall Dan Modelling View Controller (Mvc)," *Antivirus J. Ilm. Tek. Inform.*, vol. 16, no. 2, pp. 211–222, 2022, doi: 10.35457/antivirus.v16i2.2489.