

# Sistem Kendali Lampu Multi Ruangan Berbasis IoT Menggunakan Blynk

Hafizh Freda Mawasta<sup>1</sup>, Muhammad Solehudin<sup>2</sup>

<sup>1,2</sup> SMA Tunas Luhur

e-mail: [it.tunasluluh@gmail.com](mailto:it.tunasluluh@gmail.com)

## Abstrak

Sistem kendali lampu multi ruangan berbasis teknologi memungkinkan pengguna mengontrol pencahayaan secara efisien. Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem kendali lampu multi-ruangan berbasis Internet of Things (IoT) yang dapat dikendalikan melalui aplikasi Blynk yang dapat diakses melalui aplikasi. Penelitian ini menggunakan model pengembangan *System Development Life Cycle* (SDLC) dengan tahapan perencanaan, analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan pemeliharaan untuk menghasilkan sistem kendali lampu multi-ruangan yang terintegrasi dan dapat dioperasikan dari jarak jauh. Teknik analisis dilakukan dengan menguji respons sistem terhadap perintah pengguna serta efisiensi konsumsi energi. Hasil penelitian menunjukkan bahwa sistem ini dapat berfungsi dengan baik dalam mengontrol lampu secara real-time, memberikan fleksibilitas dalam pengaturan waktu operasional, serta meningkatkan efisiensi energi dan kenyamanan pengguna.

**Keywords:** *Sistem Kendali Lampu Otomatis, Kontrol Lampu Multi-Ruangan, Internet Of Things, Blynk, Penghematan Energi.*

## Abstract

The multi-room lighting control system based on technology allowed users to control lighting efficiently. This study aimed to design and implement a multi-room lighting control system based on the Internet of Things (IoT), which was controlled through the Blynk application. The study employed the *System Development Life Cycle* (SDLC) model with the stages of planning, requirement analysis, system design, implementation, testing, and maintenance to produce an integrated multi-room lighting control system that could be operated remotely. The analysis technique was carried out by testing the system's response to user commands as well as the efficiency of energy consumption. The results showed that the system functioned properly in controlling lights in real-time, provided flexibility in scheduling operational time, and improved both energy efficiency and user convenience.

**Keywords:** *Automatic Lighting Control System, Multi-Room Lighting Control, Internet of Things, Blynk, Energy Saving*

## PENDAHULUAN

Teknologi *Internet of Things* (IoT) telah memberikan dampak signifikan dalam kehidupan sehari-hari, termasuk dalam pengelolaan rumah tangga melalui konsep rumah pintar (*Smart Home*). Salah satu penerapan IoT yang paling umum adalah sistem kendali lampu, yang memungkinkan pengguna mengontrol pencahayaan di berbagai ruangan menggunakan perangkat seluler atau perintah suara (Fitriansyah & Suryanto, 2021). Sistem ini menawarkan kemudahan, fleksibilitas, serta efisiensi yang lebih tinggi dibandingkan saklar konvensional, terutama di rumah dengan banyak ruangan (Zainudin & Dede Supiyana, 2023).

Realisasi sistem kendali lampu multi-ruangan berbasis IoT melalui aplikasi Blynk yang memerlukan proses perancangan yang mencakup perangkat keras (*Hardware*) dan perangkat lunak (*Software*). Pada perangkat keras seperti mikrokontroler ESP32 digunakan untuk mengendalikan sistem seperti relay yang terhubung ke lampu di setiap ruangan. Sedangkan Perangkat lunak seperti Blynk berperan sebagai sistem kendali pada lampu. Pengguna dapat membuat antarmuka kontrol yang intuitif di perangkat seluler, sehingga pengguna dapat

mengontrol hidup dan mati nya lampu di semua ruangan secara *real-time* dan dimana saja (Mamahit, 2024). Dengan sistem ini, pengguna dapat mengatur jadwal hidup dan mati lampu, serta mengoptimalkan penggunaan energi berdasarkan kondisi aktual (Fuady dkk, 2021) di setiap ruangan, sehingga meningkatkan efisiensi dan kenyamanan dalam pengelolaan rumah pintar. (Zannatiamo & Jakaria, 2024). Keterbatasan kontrol jarak jauh merupakan salah satu masalah umum dalam sistem kontrol lampu konvensional. Banyak pengguna masih menggunakan saklar manual untuk menyalakan dan mematikan lampu, yang tidak praktis, terutama bagi penghuni rumah besar atau mereka yang memiliki keterbatasan mobilitas. Sistem kendali lampu berbasis IoT hadir sebagai solusi yang lebih responsif dan fleksibel karena memungkinkan pengguna mengontrol lampu melalui smartphone dari jarak jauh (Ibrahim & Setiyadi, 2021). Namun, sebagian besar sistem tersebut masih mengandalkan modul relay dan komponen tambahan lainnya yang dapat meningkatkan biaya dan kompleksitas instalasi (Tenriawaru, Saputra, & Yusril, 2023). Berdasarkan permasalahan yang dijelaskan, penelitian ini bertujuan untuk merancang dan mengembangkan Prototype sistem kontrol pencahayaan multi-ruangan berbasis IoT. Sistem ini bekerja secara khusus menggunakan mikrokontroler ESP32 WEMOS D1 R32 dan menggunakan aplikasi Blynk. Berbeda dengan penelitian sebelumnya yang masih didasarkan pada modul Relay dan komponen tambahan, pendekatan penelitian ini menunjukkan sistem yang lebih sederhana. Sistem ini berfokus langsung pada kontrol dengan pin GPIO ESP32 (semua input/output sudut pandang) tanpa perlu modul Relay atau komponen eksternal tambahan. Strategi ini membuat Prototype sistem yang dihasilkan menjadi sederhana dan mudah diimplementasikan, dan diharapkan bahwa kontrol cahaya multi-ruangan akan secara efektif dan efektif melalui perangkat seluler.

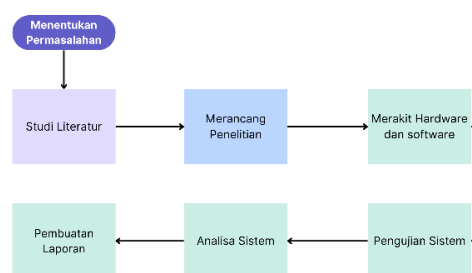
## METODE

Dalam penelitian ini, metode yang digunakan dalam penelitian ini adalah metode prototipe yang bertujuan untuk merancang dan membangun sistem kendali lampu multi-ruangan berbasis *Internet of Things* (IoT). Sistem ini menggunakan mikrokontroler Arduino ESP32 Wemos D1 R32 dan aplikasi Blynk sebagai antarmuka kontrol. Prototipe sistem diwujudkan dalam bentuk miniatur rumah dengan tiga ruangan, masing-masing dilengkapi dengan satu buah lampu LED kecil bertegangan sekitar 3,5V. Sistem ini tidak menggunakan modul relay maupun resistor, sehingga LED dikendalikan langsung melalui pin output digital dari mikrokontroler.

Selanjutnya, perancangan sistem mencakup integrasi perangkat keras dan perangkat lunak untuk menciptakan kendali lampu jarak jauh melalui aplikasi Blynk.

### Alur Penelitian

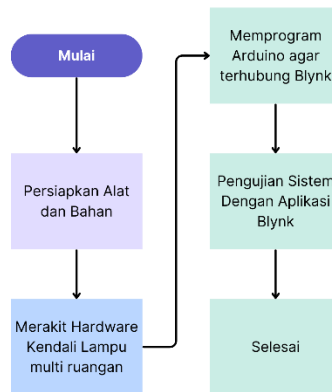
Langkah-langkah penelitian ini disusun secara sistematis untuk mencapai tujuan yang telah ditetapkan. Secara umum, alur penelitian dapat digambarkan melalui bagan pada Gambar berikut ini:



**Gambar 1. Bagan alur penelitian**

### Alur Perancangan

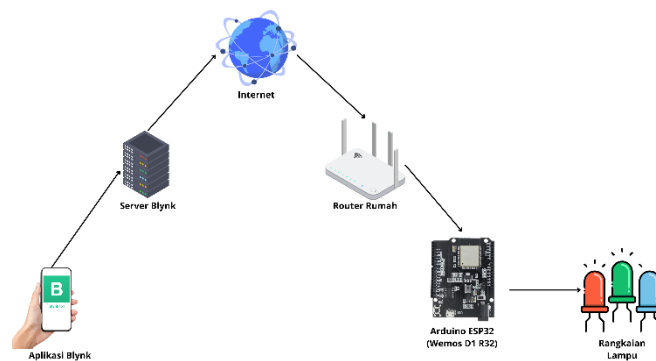
Pada bagian alur perancangan alat yang akan dilakukan dapat digambarkan melalui bagan pada gambar 2 berikut ini



**Gambar 2. Bagan alur perancangan**

### Perancangan Sistem

Pada perancangan sistem kendali lampu multi-ruangan berbasis Blynk, dilakukan tahap pemilihan komponen, perancangan rangkaian, serta pemrograman mikrokontroler untuk mengatur nyala lampu melalui aplikasi secara nirkabel. Dapat digambarkan melalui bagan pada gambar berikut ini:



**Gambar 3. Bagan Perancangan sistem**

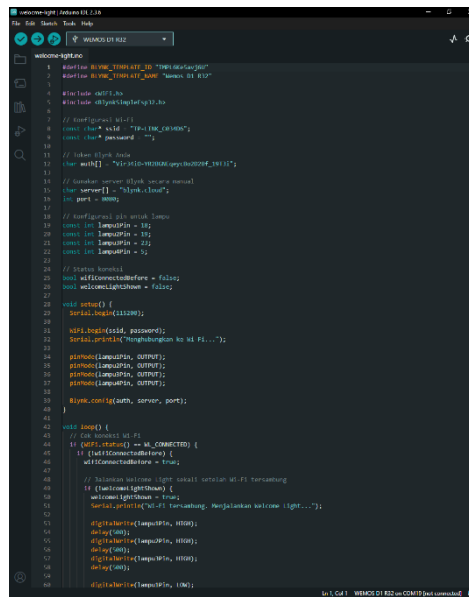
## HASIL DAN PEMBAHASAN

### Hasil rangkaian Perangkat Keras

Pada rangkaian perangkat keras dilakukan dengan cara merangkai hardware, proses pengujian, dan hasil dari pengujian. Pengujian hardware yang digunakan pada rangkaian alat perangkat keras meliputi:

1. Mikrokontroler ESP32 (Wemos D1 R32)
2. Lampu LED
3. Aplikasi Blynk
4. Integrasi system

## Pengujian Mikrokontroler ESP32



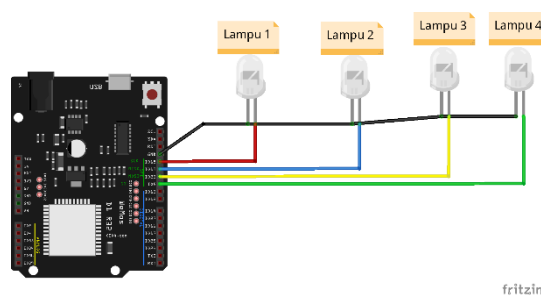
```
welcome-light.ino
1 #include <WiFi.h>
2 #include <SPI.h>
3 #include <Wire.h>
4 #include <Arduino.h>
5 #include <ESP8266WiFi.h>
6 #include <ESP8266WiFi.h>
7 // Definisi pin
8 const int LED1 = 18;
9 const int LED2 = 19;
10 const int LED3 = 23;
11 const int LED4 = 5;
12 // Definisi pin server
13 const int serverPin = 16;
14 // Definisi pin server
15 const int serverPin = 16;
16 // Definisi pin server
17 const int serverPin = 16;
18 // Definisi pin server
19 const int serverPin = 16;
20 // Definisi pin server
21 const int serverPin = 16;
22 // Definisi pin server
23 const int serverPin = 16;
24 // Definisi pin server
25 const int serverPin = 16;
26 // Definisi pin server
27 const int serverPin = 16;
28 // Definisi pin server
29 const int serverPin = 16;
30 // Definisi pin server
31 const int serverPin = 16;
32 // Definisi pin server
33 const int serverPin = 16;
34 // Definisi pin server
35 const int serverPin = 16;
36 // Definisi pin server
37 const int serverPin = 16;
38 // Definisi pin server
39 const int serverPin = 16;
40 // Definisi pin server
41 const int serverPin = 16;
42 // Definisi pin server
43 const int serverPin = 16;
44 // Definisi pin server
45 const int serverPin = 16;
46 // Definisi pin server
47 const int serverPin = 16;
48 // Definisi pin server
49 const int serverPin = 16;
50 // Definisi pin server
51 const int serverPin = 16;
52 // Definisi pin server
53 const int serverPin = 16;
54 // Definisi pin server
55 const int serverPin = 16;
56 // Definisi pin server
57 const int serverPin = 16;
58 // Definisi pin server
59 const int serverPin = 16;
60 // Definisi pin server
61 const int serverPin = 16;
62 // Definisi pin server
63 const int serverPin = 16;
64 // Definisi pin server
65 const int serverPin = 16;
66 // Definisi pin server
67 const int serverPin = 16;
68 // Definisi pin server
69 const int serverPin = 16;
70 // Definisi pin server
71 const int serverPin = 16;
72 // Definisi pin server
73 const int serverPin = 16;
74 // Definisi pin server
75 const int serverPin = 16;
76 // Definisi pin server
77 const int serverPin = 16;
78 // Definisi pin server
79 const int serverPin = 16;
80 // Definisi pin server
81 const int serverPin = 16;
82 // Definisi pin server
83 const int serverPin = 16;
84 // Definisi pin server
85 const int serverPin = 16;
86 // Definisi pin server
87 const int serverPin = 16;
88 // Definisi pin server
89 const int serverPin = 16;
90 // Definisi pin server
91 const int serverPin = 16;
92 // Definisi pin server
93 const int serverPin = 16;
94 // Definisi pin server
95 const int serverPin = 16;
96 // Definisi pin server
97 const int serverPin = 16;
98 // Definisi pin server
99 const int serverPin = 16;
100 // Definisi pin server
101 const int serverPin = 16;
102 // Definisi pin server
103 const int serverPin = 16;
104 // Definisi pin server
105 const int serverPin = 16;
106 // Definisi pin server
107 const int serverPin = 16;
108 // Definisi pin server
109 const int serverPin = 16;
110 // Definisi pin server
111 const int serverPin = 16;
112 // Definisi pin server
113 const int serverPin = 16;
114 // Definisi pin server
115 const int serverPin = 16;
116 // Definisi pin server
117 const int serverPin = 16;
118 // Definisi pin server
119 const int serverPin = 16;
120 // Definisi pin server
121 const int serverPin = 16;
122 // Definisi pin server
123 const int serverPin = 16;
124 // Definisi pin server
125 const int serverPin = 16;
126 // Definisi pin server
127 const int serverPin = 16;
128 // Definisi pin server
129 const int serverPin = 16;
130 // Definisi pin server
131 const int serverPin = 16;
132 // Definisi pin server
133 const int serverPin = 16;
134 // Definisi pin server
135 const int serverPin = 16;
136 // Definisi pin server
137 const int serverPin = 16;
138 // Definisi pin server
139 const int serverPin = 16;
140 // Definisi pin server
141 const int serverPin = 16;
142 // Definisi pin server
143 const int serverPin = 16;
144 // Definisi pin server
145 const int serverPin = 16;
146 // Definisi pin server
147 const int serverPin = 16;
148 // Definisi pin server
149 const int serverPin = 16;
150 // Definisi pin server
151 const int serverPin = 16;
152 // Definisi pin server
153 const int serverPin = 16;
154 // Definisi pin server
155 const int serverPin = 16;
156 // Definisi pin server
157 const int serverPin = 16;
158 // Definisi pin server
159 const int serverPin = 16;
160 // Definisi pin server
161 const int serverPin = 16;
162 // Definisi pin server
163 const int serverPin = 16;
164 // Definisi pin server
165 const int serverPin = 16;
166 // Definisi pin server
167 const int serverPin = 16;
168 // Definisi pin server
169 const int serverPin = 16;
170 // Definisi pin server
171 const int serverPin = 16;
172 // Definisi pin server
173 const int serverPin = 16;
174 // Definisi pin server
175 const int serverPin = 16;
176 // Definisi pin server
177 const int serverPin = 16;
178 // Definisi pin server
179 const int serverPin = 16;
180 // Definisi pin server
181 const int serverPin = 16;
182 // Definisi pin server
183 const int serverPin = 16;
184 // Definisi pin server
185 const int serverPin = 16;
186 // Definisi pin server
187 const int serverPin = 16;
188 // Definisi pin server
189 const int serverPin = 16;
190 // Definisi pin server
191 const int serverPin = 16;
192 // Definisi pin server
193 const int serverPin = 16;
194 // Definisi pin server
195 const int serverPin = 16;
196 // Definisi pin server
197 const int serverPin = 16;
198 // Definisi pin server
199 const int serverPin = 16;
200 // Definisi pin server
201 const int serverPin = 16;
202 // Definisi pin server
203 const int serverPin = 16;
204 // Definisi pin server
205 const int serverPin = 16;
206 // Definisi pin server
207 const int serverPin = 16;
208 // Definisi pin server
209 const int serverPin = 16;
210 // Definisi pin server
211 const int serverPin = 16;
212 // Definisi pin server
213 const int serverPin = 16;
214 // Definisi pin server
215 const int serverPin = 16;
216 // Definisi pin server
217 const int serverPin = 16;
218 // Definisi pin server
219 const int serverPin = 16;
220 // Definisi pin server
221 const int serverPin = 16;
222 // Definisi pin server
223 const int serverPin = 16;
224 // Definisi pin server
225 const int serverPin = 16;
226 // Definisi pin server
227 const int serverPin = 16;
228 // Definisi pin server
229 const int serverPin = 16;
230 // Definisi pin server
231 const int serverPin = 16;
232 // Definisi pin server
233 const int serverPin = 16;
234 // Definisi pin server
235 const int serverPin = 16;
236 // Definisi pin server
237 const int serverPin = 16;
238 // Definisi pin server
239 const int serverPin = 16;
240 // Definisi pin server
241 const int serverPin = 16;
242 // Definisi pin server
243 const int serverPin = 16;
244 // Definisi pin server
245 const int serverPin = 16;
246 // Definisi pin server
247 const int serverPin = 16;
248 // Definisi pin server
249 const int serverPin = 16;
250 // Definisi pin server
251 const int serverPin = 16;
252 // Definisi pin server
253 const int serverPin = 16;
254 // Definisi pin server
255 const int serverPin = 16;
256 // Definisi pin server
257 const int serverPin = 16;
258 // Definisi pin server
259 const int serverPin = 16;
260 // Definisi pin server
261 const int serverPin = 16;
262 // Definisi pin server
263 const int serverPin = 16;
264 // Definisi pin server
265 const int serverPin = 16;
266 // Definisi pin server
267 const int serverPin = 16;
268 // Definisi pin server
269 const int serverPin = 16;
270 // Definisi pin server
271 const int serverPin = 16;
272 // Definisi pin server
273 const int serverPin = 16;
274 // Definisi pin server
275 const int serverPin = 16;
276 // Definisi pin server
277 const int serverPin = 16;
278 // Definisi pin server
279 const int serverPin = 16;
280 // Definisi pin server
281 const int serverPin = 16;
282 // Definisi pin server
283 const int serverPin = 16;
284 // Definisi pin server
285 const int serverPin = 16;
286 // Definisi pin server
287 const int serverPin = 16;
288 // Definisi pin server
289 const int serverPin = 16;
290 // Definisi pin server
291 const int serverPin = 16;
292 // Definisi pin server
293 const int serverPin = 16;
294 // Definisi pin server
295 const int serverPin = 16;
296 // Definisi pin server
297 const int serverPin = 16;
298 // Definisi pin server
299 const int serverPin = 16;
300 // Definisi pin server
301 const int serverPin = 16;
302 // Definisi pin server
303 const int serverPin = 16;
304 // Definisi pin server
305 const int serverPin = 16;
306 // Definisi pin server
307 const int serverPin = 16;
308 // Definisi pin server
309 const int serverPin = 16;
310 // Definisi pin server
311 const int serverPin = 16;
312 // Definisi pin server
313 const int serverPin = 16;
314 // Definisi pin server
315 const int serverPin = 16;
316 // Definisi pin server
317 const int serverPin = 16;
318 // Definisi pin server
319 const int serverPin = 16;
320 // Definisi pin server
321 const int serverPin = 16;
322 // Definisi pin server
323 const int serverPin = 16;
324 // Definisi pin server
325 const int serverPin = 16;
326 // Definisi pin server
327 const int serverPin = 16;
328 // Definisi pin server
329 const int serverPin = 16;
330 // Definisi pin server
331 const int serverPin = 16;
332 // Definisi pin server
333 const int serverPin = 16;
334 // Definisi pin server
335 const int serverPin = 16;
336 // Definisi pin server
337 const int serverPin = 16;
338 // Definisi pin server
339 const int serverPin = 16;
340 // Definisi pin server
341 const int serverPin = 16;
342 // Definisi pin server
343 const int serverPin = 16;
344 // Definisi pin server
345 const int serverPin = 16;
346 // Definisi pin server
347 const int serverPin = 16;
348 // Definisi pin server
349 const int serverPin = 16;
350 // Definisi pin server
351 const int serverPin = 16;
352 // Definisi pin server
353 const int serverPin = 16;
354 // Definisi pin server
355 const int serverPin = 16;
356 // Definisi pin server
357 const int serverPin = 16;
358 // Definisi pin server
359 const int serverPin = 16;
360 // Definisi pin server
361 const int serverPin = 16;
362 // Definisi pin server
363 const int serverPin = 16;
364 // Definisi pin server
365 const int serverPin = 16;
366 // Definisi pin server
367 const int serverPin = 16;
368 // Definisi pin server
369 const int serverPin = 16;
369
```

Gambar 4. Program pengujian sistem ESP32

Pengujian Mikrokontroler ESP32 ini bertujuan untuk memastikan bahwa sistem Mikrokontroler ESP32 yang digunakan pada penelitian berjalan lancar. Sehingga program dapat berjalan mengontrol lampu LED sesuai harapan

### Hasil Rangkaian Sistem Kendali Lampu

Pengujian sistem kendali lampu dilakukan dengan merangkai komponen seperti pada Gambar 5, yaitu menggunakan mikrokontroler ESP32 dan dikendalikan melalui aplikasi Blynk. ESP32 menerima sinyal digital dari aplikasi dan mengaktifkan empat buah lampu LED yang terhubung ke pin digital GPIO 18, 19, 23, dan 5. Setiap lampu dikontrol melalui virtual pin (V0–V4), termasuk fitur kontrol serentak (V3) dan animasi *welcome light* yang akan aktif secara otomatis saat ESP32 terhubung ke jaringan WiFi. Pengujian dilakukan dengan mencoba menyalakan dan mematikan lampu dari aplikasi serta memastikan animasi *welcome light* berjalan normal setiap kali sistem dinyalakan dan berhasil terhubung ke WiFi.

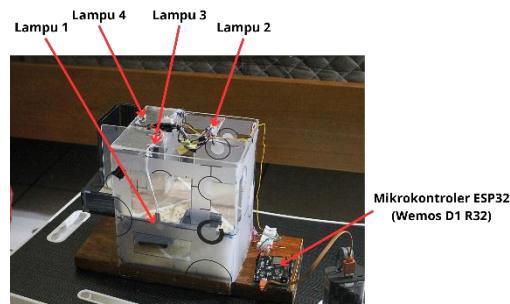


Gambar 5. Desain rangkaian sistem kendali lampu

### Hasil Implementasi Sistem Lampu Multi-Ruangan

secara umum prototype sistem terdiri dari rangkaian mikrokontroler ESP32 dan lampu LED, yang terhubung melalui pin output digital dan dikendalikan melalui koneksi Wi-Fi. Sistem ini memungkinkan kontrol jarak jauh terhadap perangkat lampu menggunakan aplikasi berbasis web atau mobile, serta mendemonstrasikan fungsi dasar dari integrasi mikrokontroler dan aktuator dalam sistem otomatisasi sederhana. Hasil rangkaian sistem kendali lampu multi-ruangan dapat dilihat pada gambar...

## Hasil Desain Antarmuka Blynk

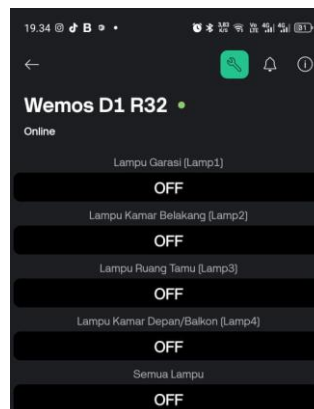


**Gambar 6. Hasil implementasi rangkaian sistem kendali lampu**

Dalam perancangan sistem kontrol lampu multi-ruangan ini, dibutuhkan sebuah antarmuka yang memudahkan pengguna dalam mengakses dan mengendalikan perangkat secara fleksibel melalui perangkat mobile. Untuk memenuhi kebutuhan tersebut, digunakan aplikasi Blynk sebagai media antarmuka berbasis Android.

Aplikasi ini dipilih karena mendukung integrasi langsung dengan mikrokontroler ESP32 melalui koneksi internet, serta menyediakan tampilan antarmuka yang dapat dikustomisasi tanpa perlu membuat aplikasi dari awal. Pengguna cukup menambahkan elemen seperti tombol, switch, dan indikator sesuai jumlah dan fungsi lampu yang dikendalikan.

Sebelum digunakan, pengguna harus mengunduh aplikasi Blynk melalui layanan Google Play Store. Setelah instalasi selesai, pengguna dapat melakukan konfigurasi project dengan memilih jenis perangkat (ESP32) dan memasukkan authentication token yang dihubungkan ke program mikrokontroler.



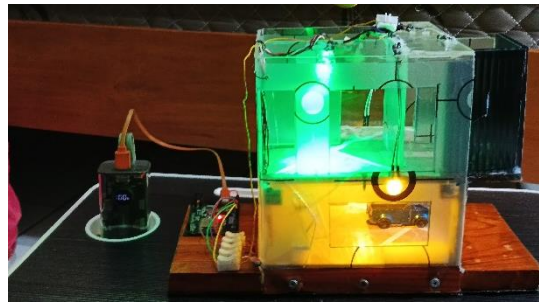
**Gambar 7. Desain antarmuka kendali lampu multi ruangan**

Dengan adanya aplikasi ini, sistem menjadi lebih praktis, karena seluruh pengendalian dapat dilakukan melalui satu perangkat tanpa memerlukan tambahan hardware kendali manual.

## Hasil pengujian perangkat keseluruhan

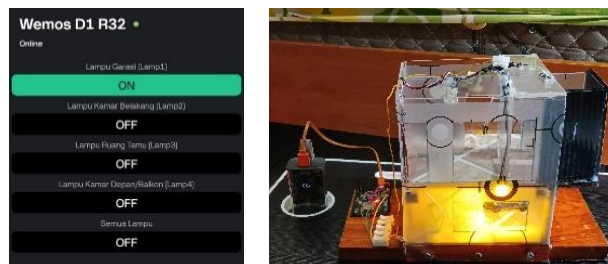
Hasil dari penelitian ini adalah menguji perangkat keras dan perangkat lunak sistem kendali lampu multi-ruangan menggunakan aplikasi Blynk. Pengujian dilakukan untuk memastikan bahwa setiap komponen sistem berfungsi dengan baik, mulai dari koneksi antara mikrokontroler ESP32 dan jaringan Wi-Fi, respon sistem terhadap perintah dari aplikasi, hingga efektivitas kendali terhadap masing-masing lampu pada ruangan yang berbeda.

Selama proses pengujian, sistem berhasil merespons perintah dari aplikasi secara real-time dengan tingkat keakuratan yang tinggi. Lampu dapat dinyalakan dan dimatikan sesuai tombol virtual yang ditekan pada aplikasi, serta koneksi antara perangkat dan server Blynk tetap stabil selama pengujian berlangsung. Hasil ini menunjukkan bahwa sistem yang dirancang telah bekerja sesuai dengan fungsinya dan siap untuk diterapkan dalam skala ruang kerja sebenarnya.

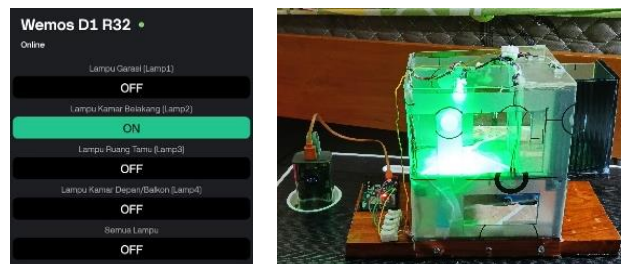


**Gambar 8. Kondisi saat terhubung ke WiFi**

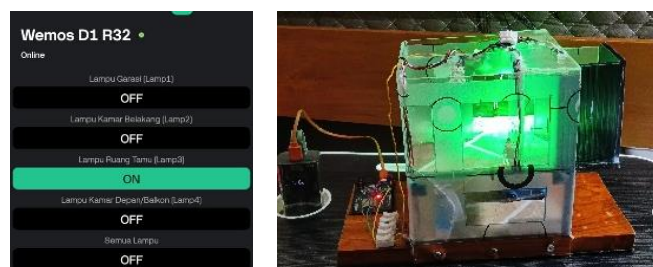
Pada saat Arduino terhubung ke Internet (Gambar 8) maka akan ada seperti animasi (*Welcome Light*) menandakan bahwa Arduino terhubung ke Internet dan sudah dapat dikendalikan menggunakan aplikasi Blynk.



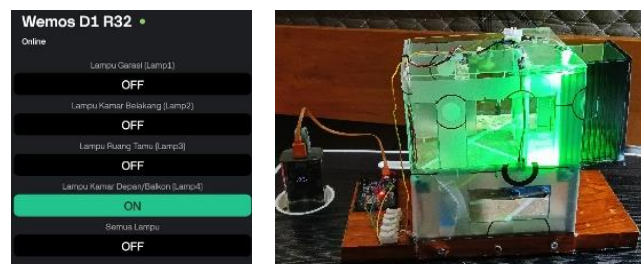
**Gambar 9. Hasil kendali lampu multi ruangan melalui Blynk**



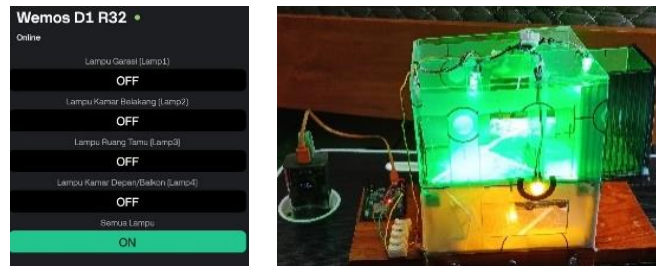
**Gambar 10. Hasil kendali lampu multi ruangan 1 melalui Blynk**



**Gambar 11. Hasil kendali lampu multi ruangan 2 melalui Blynk**



**Gambar 12. Hasil kendali lampu multi ruangan 3 melalui Blynk**



Gambar 13. Hasil kendali lampu multi ruangan 4 melalui Blynk

## SIMPULAN

Hasil penelitian menunjukkan bahwa sistem prototipe kendali lampu multi-ruangan berbasis mikrokontroler ESP32 Wemos D1 R32 dan aplikasi Blynk berhasil diimplementasikan sesuai fungsi yang dirancang, mampu mengontrol tiga lampu LED secara nirkabel melalui smartphone dengan respons yang baik. Sistem ini tidak terbatas pada jaringan WiFi lokal, karena Blynk memungkinkan kendali dilakukan dari mana saja selama perangkat terhubung ke internet. Sistem ini sederhana, hemat biaya, dan edukatif karena memanfaatkan GPIO secara langsung tanpa relay untuk prototipe berdaya rendah, sehingga cocok dijadikan dasar pengembangan sistem otomatisasi rumah sederhana berbasis IoT.

Untuk pengembangan lebih lanjut pada skala rumah tangga nyata, disarankan penggunaan modul relay demi keamanan, penambahan fitur autentikasi dan enkripsi, serta integrasi sensor seperti PIR, LDR, dan suhu, termasuk modul RTC, agar fungsionalitas semakin optimal. Selain itu, penggunaan WiFi repeater atau teknologi komunikasi lain seperti LoRa atau NB-IoT dapat meningkatkan stabilitas koneksi dan memperluas jangkauan, sehingga sistem lebih aplikatif dalam skenario smart lighting berskala besar.

## DAFTAR PUSTAKA

- Fitriansyah, A., & Suryanto, M. R. (2021). Teknologi Kontrol Lampu dan Kunci Rumah Berbasis IoT. *Jurnal Teknologi Informatika dan Komputer*, 7(1), 88–96.
- Fuady, S., Khaira, U., Hais, Y. R., & Sitanggang, R. H. (2021). Penerapan Model Terdistribusi untuk Sistem Smarthome Menggunakan Multi-Sensor Berbasis Internet of Things (IoT). *Journal of Electrical Power Control and Automation (JEPCA)*, 4(2), 68.
- Ibrahim, A. M., & Setiyadi, D. (2021). PROTOTYPE PENGENDALIAN LAMPU DAN AC JARAK JAUH DENGAN JARINGAN INTERNET MENGGUNAKAN APLIKASI TELEGRAM BERBASIS NODEMCU ESP8266. *Infotech: Journal of Technology Information*, 7(1), 27–34.
- Mamahit, C. (2024). Rumah Pintar dengan Lampu Kontrol Suara Menggunakan Arduino Uno R3. *Electrician: Jurnal Rekayasa dan Teknologi Elektro*, 18(2), 144–152.
- Supiyana, D. (2023). PERANCANGAN DAN IMPLEMENTASI KENDALI LAMPU RUANG BERBASIS IOT MENGGUNAKAN NODEMCU ESP32, 1(3).
- Tenriawaru, A., Saputra, R. A., & Yusril, M. (2023). Sistem Kendali Lampu Otomatis Multisensor Menggunakan Metode Fuzzy Logic Control Inferensi Sugeno Berbasis Mikrokontroler. *Jurnal Eksplora Informatika*, 13(1), 12–23.
- Zannatiamo, A. C., & Jakaria, D. A. (2024). Perancangan Sistem Kendali Otomatis pada Prototype Smart Home dengan Menggunakan Metode Fuzzy Logic Control, 17(1).