**Praktikum Rangkaian LED dengan Tiga Tombol Input**

*Maradu Denny S. Hutahaean*

*Fakultas Vokasi , Universitas Brawijaya*

*Email :* [*maradudenny@student.ub.ac.id*](mailto:maradudenny@student.ub.ac.id)

**Abstrak :**

Praktikum ini membahas simulasi sistem lampu lalu lintas berbasis Internet of Things (IoT) menggunakan platform Wokwi sebagai alat simulasi utama. Eksperimen dilakukan dengan merancang dan mengimplementasikan kode Arduino untuk mengatur perubahan warna lampu lalu lintas berdasarkan waktu yang telah ditentukan. Setiap lampu akan menyala dalam durasi tertentu sebelum beralih ke warna berikutnya, sesuai dengan logika pengendalian lalu lintas. Selain pemrograman, eksperimen ini juga mencakup integrasi komponen virtual, seperti mikrokontroler ESP32, LED, dan resistor, guna mereplikasi sistem fisik dalam lingkungan simulasi. Dengan memanfaatkan Wokwi, sistem ini dapat diuji dan dianalisis secara interaktif tanpa memerlukan perangkat keras fisik. Hasil dari eksperimen ini diharapkan dapat memberikan pemahaman mendalam mengenai konsep pengendalian lampu lalu lintas berbasis IoT serta meningkatkan keterampilan dalam pemrograman mikrokontroler dan simulasi sistem tertanam.

**Kata Kunci :** *Arduino, ESP32, Wokwi, Internet Of Things.*

**Abstract:**

This practicum discusses the simulation of an Internet of Things (IoT)-based traffic light system using the Wokwi platform as the main simulation tool. The experiment was carried out by designing and implementing Arduino code to adjust the color change of the traffic light based on a predetermined time. Each light will illuminate for a certain duration before switching to the next color, according to the logic of traffic control. In addition to programming, the experiment also includes the integration of virtual components, such as ESP32 microcontrollers, LEDs, and resistors, to replicate physical systems in a simulated environment. By utilizing Wokwi, these systems can be tested and analyzed interactively without the need for physical hardware. The results of this experiment are expected to provide an in-depth understanding of the concept of IoT-based traffic light control as well as improve skills in microcontroller programming and embedded system simulation.

**Kata Kunci :** *Arduino, ESP32, Wokwi, Internet Of Things.*

**PENDAHULUAN**

Sistem kendali berbasis mikrokontroler atau rangkaian elektronik sering digunakan untuk mengontrol perangkat seperti lampu LED berdasarkan input dari tombol. Pada praktikum ini, kita akan merancang dan mengimplementasikan sebuah sistem dengan tiga tombol sebagai input dan tiga LED sebagai output. Setiap tombol akan menghasilkan pola nyala-mati (kedip) LED yang berbeda sesuai dengan tugas yang telah ditentukan. Penggunaan tombol sebagai pemicu LED yang berkedip dengan pola tertentu merupakan konsep dasar dalam sistem otomasi dan kendali digital. Praktikum ini bertujuan untuk memahami logika kontrol sederhana, pemrograman mikrokontroler atau perangkat pengendali, serta penerapan rangkaian elektronik dasar.

Adapun tujuan dari praktikum ini adalah memahami cara kerja tombol sebagai input dalam sistem elektronik, mempelajari cara mengontrol LED berdasarkan sinyal input dari tombol, serta mengimplementasikan logika pemrograman untuk menghasilkan pola kedipan LED sesuai dengan spesifikasi berikut: Tombol 1 menyebabkan LED merah berkedip sebanyak 5 kali, Tombol 2 menyebabkan LED merah dan hijau berkedip secara bergantian, dan Tombol 3 menyebabkan LED merah, kuning, dan hijau berkedip secara bergantian. Dengan melakukan praktikum ini, diharapkan peserta dapat memahami dasar-dasar pengendalian perangkat elektronik sederhana dan meningkatkan keterampilan dalam pemrograman sistem berbasis tombol dan LED.

**METODOLOGI**

1. **Alat dan Bahan**

Mikrokontroler ESP32, 3 buah LED (merah, kuning, hijau), 3 buah resistor, 3 buah button ,kabel, software Visual Studio Code, PlatformIo, dan platform Wokwi .

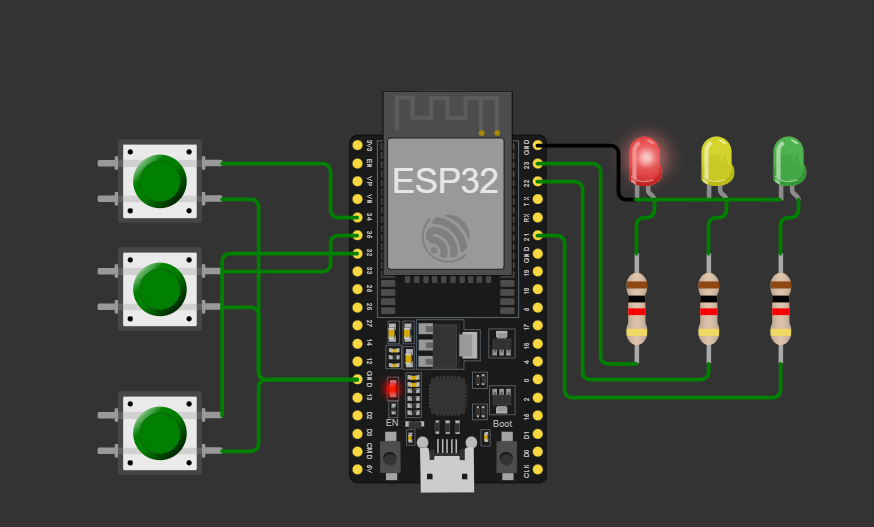
1. **Langkah Perancangan**
2. **Pilih ESP32** sebagai mikrokontroler pada situs [Wokwi](https://wokwi.com/).
3. **Tambahkan komponen pendukung** seperti resistor, tombol dan lampu LED berwarna merah, kuning, dan hijau untuk menyusun rangkaian yang lengkap. Pastikan seluruh komponen tersambung dengan benar ke ESP32.
4. **Unduh dan pasang PlatformIO IDE serta Wokwi Simulator** melalui ekstensi pada Visual Studio Code. Selanjutnya, buat proyek baru dengan memilih **Board DOIT ESP32 DEVKIT V1** dan **Framework Arduino** sebagai basis pengembangannya.
5. **Kompilasi proyek** yang telah dibuat dengan menekan tombol **Compile** di bagian kiri bawah aplikasi untuk memastikan tidak ada kesalahan dalam kode sebelum menjalankan simulasi.
6. Buat file diagram pada project dan copy script diagram.json di website.
7. {
8. "version": 1,
9. "author": "Maradu denny",
10. "editor": "wokwi",
11. "parts": [
12. { "type": "board-esp32-devkit-c-v4", "id": "esp", "top": 76.8, "left": -62.36, "attrs": {} },
13. {
14. "type": "wokwi-led",
15. "id": "led1",
16. "top": -99.6,
17. "left": -53.8,
18. "attrs": { "color": "limegreen" }
19. },
20. {
21. "type": "wokwi-led",
22. "id": "led2",
23. "top": -99.6,
24. "left": 13.4,
25. "attrs": { "color": "yellow" }
26. },
27. { "type": "wokwi-led", "id": "led3", "top": -99.6, "left": 51.8, "attrs": { "color": "red" } },
28. {
29. "type": "wokwi-resistor",
30. "id": "r1",
31. "top": 24,
32. "left": 47.45,
33. "rotate": 90,
34. "attrs": { "value": "1000" }
35. },
36. {
37. "type": "wokwi-resistor",
38. "id": "r2",
39. "top": 24,
40. "left": -58.15,
41. "rotate": 90,
42. "attrs": { "value": "1000" }
43. },
44. {
45. "type": "wokwi-resistor",
46. "id": "r3",
47. "top": 24,
48. "left": 18.65,
49. "rotate": 90,
50. "attrs": { "value": "1000" }
51. },
52. {
53. "type": "wokwi-pushbutton",
54. "id": "btn1",
55. "top": 169.4,
56. "left": -182.4,
57. "attrs": { "color": "green", "xray": "1" }
58. },
59. {
60. "type": "wokwi-pushbutton",
61. "id": "btn2",
62. "top": 111.8,
63. "left": -182.4,
64. "attrs": { "color": "green", "xray": "1" }
65. },
66. {
67. "type": "wokwi-pushbutton",
68. "id": "btn3",
69. "top": 54.2,
70. "left": -182.4,
71. "attrs": { "color": "green", "xray": "1" }
72. },
73. { "type": "wokwi-junction", "id": "j1", "top": 158.4, "left": -216, "attrs": {} }
74. ],
75. "connections": [
76. [ "esp:TX", "$serialMonitor:RX", "", [] ],
77. [ "esp:RX", "$serialMonitor:TX", "", [] ],
78. [ "led1:C", "esp:GND.2", "black", [ "v0" ] ],
79. [ "led2:C", "esp:GND.2", "black", [ "v86.4", "h-76.4" ] ],
80. [ "led3:C", "esp:GND.2", "black", [ "v0" ] ],
81. [ "led1:A", "r2:1", "cyan", [ "v0" ] ],
82. [ "r2:2", "esp:19", "cyan", [ "h0", "v37.2" ] ],
83. [ "r3:2", "esp:18", "green", [ "h0", "v46.8" ] ],
84. [ "r1:2", "esp:17", "yellow", [ "h0", "v66" ] ],
85. [ "led3:A", "r1:1", "yellow", [ "v0" ] ],
86. [ "led2:A", "r3:1", "green", [ "v0" ] ],
87. [ "btn3:1.r", "esp:33", "green", [ "v0", "h38.6", "v76.8" ] ],
88. [ "esp:35", "btn2:1.r", "green", [ "h-28.65", "v-28.8", "h-9.6" ] ],
89. [ "esp:32", "btn1:1.r", "green", [ "h-19.05", "v19.2" ] ],
90. [ "btn1:2.r", "esp:GND.1", "green", [ "h19.4", "v29" ] ],
91. [ "btn3:2.r", "j1:J", "green", [ "h19.4", "v19.4", "h-105.6" ] ],
92. [ "j1:J", "esp:GND.1", "green", [ "v0" ] ],
93. [ "btn2:2.r", "j1:J", "green", [ "h9.8", "v19.4" ] ]
94. ],
95. "dependencies": {}
96. }
97. **Ubah nama file diagram** menjadi **diagram.json** agar gambar rangkaian dapat ditampilkan dengan benar.
98. Buka file main.cpp di dalam folder src untuk menulis kode yang mengontrol sistem lampu lalu lintas. Atur agar lampu merah menyala selama 30 detik, lampu kuning selama 5detik, dan lampu hijau selama 20 detik. Pastikan program berjalan sesuai dengan konfigurasi pada file diagram.json untuk mensimulasikan rangkaian secara optimal.
99. #include <Arduino.h>
100. // Deklarasi pin tombol
101. const int tombol1 = 33;
102. const int tombol2 = 35;
103. const int tombol3 = 32;
104. // Deklarasi pin LED
105. const int lampuMerah = 17;
106. const int lampuKuning = 18;
107. const int lampuHijau = 19;
108. void setup() {
109. Serial.begin(115200);
110. Serial.println("Program LED dan Tombol ESP32");
111. // Konfigurasi tombol sebagai input dengan pull-up internal
112. pinMode(tombol1, INPUT\_PULLUP);
113. pinMode(tombol2, INPUT\_PULLUP);
114. pinMode(tombol3, INPUT\_PULLUP);
115. // Konfigurasi LED sebagai output
116. pinMode(lampuMerah, OUTPUT);
117. pinMode(lampuKuning, OUTPUT);
118. pinMode(lampuHijau, OUTPUT);
119. }
120. void loop() {
121. if (digitalRead(tombol1) == LOW) {
122. // Tombol 1: Lampu merah berkedip 5 kali
123. for (int i = 0; i < 5; i++) {
124. digitalWrite(lampuMerah, HIGH);
125. Serial.println("Merah Menyala");
126. delay(500);
127. digitalWrite(lampuMerah, LOW);
128. Serial.println("Merah Padam");
129. delay(500);
130. }
131. }
132. if (digitalRead(tombol2) == LOW) {
133. // Tombol 2: Lampu merah dan hijau berkedip bergantian
134. for (int i = 0; i < 5; i++) {
135. digitalWrite(lampuMerah, HIGH);
136. digitalWrite(lampuHijau, LOW);
137. Serial.println("Merah Menyala");
138. delay(500);
139. digitalWrite(lampuMerah, LOW);
140. digitalWrite(lampuHijau, HIGH);
141. Serial.println("Hijau Menyala");
142. delay(500);
143. }
144. digitalWrite(lampuHijau, LOW); // Matikan LED setelah selesai
145. }
146. if (digitalRead(tombol3) == LOW) {
147. // Tombol 3: Lampu merah, kuning, dan hijau berkedip bergantian
148. for (int i = 0; i < 5; i++) {
149. digitalWrite(lampuMerah, HIGH);
150. Serial.println("Merah Menyala");
151. delay(500);
152. digitalWrite(lampuMerah, LOW);
153. digitalWrite(lampuKuning, HIGH);
154. Serial.println("Kuning Menyala");
155. delay(500);
156. digitalWrite(lampuKuning, LOW);
157. digitalWrite(lampuHijau, HIGH);
158. Serial.println("Hijau Menyala");
159. delay(500);
160. digitalWrite(lampuHijau, LOW);
161. }
162. }
163. }

**HASIL DAN PEMBAHASAN**

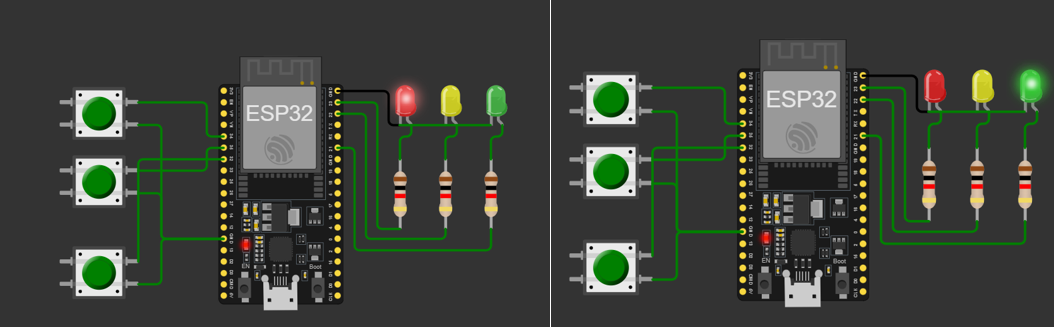
Pada praktikum ini, sistem dengan tiga tombol sebagai input dan tiga LED sebagai output telah berhasil diimplementasikan sesuai dengan spesifikasi yang ditentukan. Berdasarkan pengujian yang dilakukan, setiap tombol mampu mengontrol LED dengan pola kedipan yang sesuai. Saat Tombol 1 ditekan, LED merah berkedip sebanyak lima kali dengan interval waktu yang stabil. Saat Tombol 2 ditekan, LED merah dan hijau berkedip secara bergantian dengan durasi yang sama, menunjukkan bahwa sistem dapat menangani pergantian nyala antara dua LED. Selanjutnya, ketika Tombol 3 ditekan, LED merah, kuning, dan hijau berkedip secara bergantian dalam urutan yang benar, memastikan bahwa logika kontrol telah diterapkan dengan baik.

Pembahasan dari hasil yang diperoleh menunjukkan bahwa sistem berjalan sesuai dengan rancangan, di mana tombol berfungsi sebagai input digital yang membaca sinyal tekan dari pengguna, kemudian mengaktifkan LED sesuai dengan logika yang telah diprogram. Pola kedipan LED dikontrol

1. Tombol 1 menghidupkan lampu merah selama 5 kedipan.



1. Tombol 2 menghidupakan lampu merah dan hijau secara bergantian.



1. Tombol 3 menghidupakan lampu merah, kuning dan hijau secara bergantian.

