

LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

**(Simulasi Traffic Light Berbasis IoT dengan ESP32 Menggunakan Platform
Wokwi yang Diintegrasikan dengan VS Code)**



Muhammad Kadavi
Fakultas Vokasi, Universitas Brawijaya
Email: kadavi2945@student.ub.ac.id

Abstrak

Tujuan dari praktikum ini adalah untuk mensimulasikan sistem lampu lalu lintas dan mengetahui bagaimana logika lampu lalu lintas bekerja dengan Internet of Things (IoT) menggunakan mikrokontroler ESP32 dan display TFT-LCD ILI9341 2.8 inci sebagai display perhitungan mundur untuk setiap warna pada lampu lalu lintas. Platform Wokwi digunakan untuk mengembangkan Simulasi, dimana ekstensi PlatformIo sebagai penghubung antara wokwi yang diintegrasikan dengan VS Code untuk mempermudah pemrograman dan debugging. Sistem ini dirancang untuk beroperasi dalam urutan waktu yang telah ditentukan dalam syntax, sehingga dapat meniru bagaimana lalu lintas berjalan di dunia nyata. Penggunaan LED sebagai indikator lampu lalu lintas dan sebagai fitur display perhitungan mundur, di mana setiap warna memiliki waktu yang berbeda untuk menyala. Hasil eksperimen menunjukkan bahwa sistem bekerja sesuai dengan algoritma yang ditulis dalam syntax. Setiap perubahan status lampu terjadi pada interval waktu yang terjadwal dan stabil. Eksperimen ini menunjukkan bahwa mikrokontroler dengan logika pemrograman dan algoritma yang tepat dapat secara efektif mereplikasi ide pengendalian lampu lalu lintas.

Kata kunci—Internet of Things, ESP32, Lampu Lalu Lintas, Wokwi, VS Code

Pendahuluan

Lampu lalu lintas didefinisikan dalam UU no. 22 tahun 2009 sebagai perangkat atau sistem yang mengontrol arus lalu lintas di persimpangan jalan, tempat pejalan kaki lewat, dan lokasi lalu lintas yang banyak kendaraan melintas. Kegunaan dari lampu ini bisa terlihat dari tiga komponen warna didalamnya ada merah yang menunjukkan untuk berhenti, kuning sebagai peringatan hati-hati, dan hijau untuk menandakan jalan, dari ketiga komponen warna tersebut tentunya untuk menandakan kapan kendaraan harus berjalan dan berhenti secara bergantian dari berbagai arah. Di dalam sistemnya tersebut sudah diatur durasi kapan warna merah harus menyala, lampu kuning harus menyala, dan lampu hijau harus menyala, sehingga dari pengaturan durasi waktu yang sudah di set sangat mempengaruhi kelancaran arus lalu lintas serta untuk menurunkan kemungkinan kecelakaan yang sering berlangsung belakangan ini.

Praktikum dilakukan untuk memahami bagaimana logika dan algoritma lampu lalu lintas diterapkan dalam dunia nyata. Dalam praktiknya, sistem lampu lalu lintas bekerja berdasarkan pengaturan waktu yang telah ditentukan untuk setiap fase lampu. Durasi tiap warna harus disesuaikan dengan kondisi lalu lintas agar kendaraan dapat bergerak dengan aman dan efisien.

Untuk mensimulasikan sistem ini, digunakan mikrokontroler ESP32 sebagai unit kendali utama dan display TFT-LCD ILI9341 2.8 inci sebagai penghitung mundur untuk setiap perubahan warna lampu lalu lintas. Simulasi dikembangkan menggunakan platform Wokwi dan diprogram melalui Visual Studio Code (VS Code) dengan bantuan ekstensi PlatformIO. Sistem ini dirancang untuk meniru cara kerja lampu lalu lintas di dunia nyata, di mana setiap perubahan status lampu dilakukan sesuai dengan interval waktu yang telah ditentukan dalam program.

Melalui praktikum ini, diharapkan dapat diperoleh pemahaman yang lebih baik mengenai cara kerja lampu lalu lintas, bagaimana algoritma pengendalian waktu diterapkan, serta bagaimana logika pemrograman digunakan untuk mengatur perubahan warna lampu secara otomatis.

Latar Belakang

Setiap hari, kita pasti melewati lampu lalu lintas saat berkendara atau berjalan kaki di jalan raya. Lampu ini berfungsi untuk mengatur arus kendaraan supaya tertib dan tidak terjadi kecelakaan. Kalau tidak ada lampu lalu lintas, bisa dibayangkan bagaimana kacanya jalanan, karena setiap pengendara mungkin ingin jalan duluan tanpa aturan yang jelas.

Lampu lalu lintas bekerja dengan sistem yang sudah diatur waktunya. Warna merah berarti berhenti, hijau berarti jalan, dan kuning sebagai tanda untuk bersiap. Semua ini berjalan dengan bantuan sebuah sistem yang memastikan pergantian warna terjadi pada waktu yang tepat. Tapi, bagaimana sebenarnya sistem ini bekerja? Bagaimana lampu bisa berganti warna sendiri tanpa ada orang yang menekan tombol atau mengendalikannya satu per satu?

Untuk memahami hal itu, praktikum ini dilakukan dengan membuat simulasi lampu lalu lintas menggunakan ESP32. Dengan simulasi ini, kita bisa melihat bagaimana logika dan algoritma pengaturan lampu lalu lintas bekerja, termasuk bagaimana sistem menghitung waktu dan menentukan kapan lampu harus berubah. Selain itu, digunakan juga display TFT-LCD sebagai penghitung mundur, sehingga kita bisa melihat dengan jelas berapa detik lagi lampu akan berganti.

Melalui praktikum ini, kita akan lebih paham bagaimana lampu lalu lintas yang kita lihat setiap hari sebenarnya diatur dan dikendalikan, serta bagaimana teknologi dapat digunakan untuk membuatnya bekerja secara otomatis sesuai aturan yang sudah ditentukan.

Tujuan

Eksperimen ini dilakukan untuk memahami bagaimana sistem lampu lalu lintas bekerja, termasuk logika dan algoritma yang mengatur pergantian warna lampu secara otomatis. Dengan menggunakan ESP32 dan display TFT-LCD, eksperimen ini mensimulasikan cara kerja lampu lalu lintas di dunia nyata, sehingga dapat dipahami bagaimana waktu setiap lampu diatur agar lalu lintas berjalan lancar.

Serta juga memahami bagaimana cara menggunakan wokwi yang diintegrasikan dengan vscode tanpa harus parktik dengan hardwarenya secara lansung

Metodelogi

Eksperimen ini dilaksanakan dengan mensimulasikan sistem lampu lalu lintas menggunakan mikrokontroler ESP32. Lampu lalu lintas dan lampu LED yang berganti warna sesuai waktu yang sudah dituliskan di dalam program.display TFT-LCD digunakan untuk menampilkan hitungan mundur setiap pergantian lampu.

Penulisan syntax dilakukan di Visual Studio Code menggunakan ekstensi PlatformIO, sementara simulasi dijalankan di Wokwi untuk menguji apakah logika dan algoritma pengaturan lampu berjalan sesuai yang diharapkan.

Software & Hardware

Software	Hardware
Wokwi	Esp32 (virtual)
Extension PlatformIO	Lampu led 3 buah (merah,kuning,hijau) (virtual)
VSCode	ILI9341 2.8" TFT-LCD display (virtual)
	Laptop

2.2 Implementation Steps (Langkah Implementasi)

1. Persiapan

1. **Buka Wokwi** (<https://wokwi.com>)
2. **Buat proyek baru dengan ESP32**, lalu tambahkan komponen:
 - **3 LED (Merah, Kuning, Hijau)**
 - **ILI9341 2.8" TFT-LCD Display**
3. **Hubungkan komponen virtual ke ESP32** menggunakan diagram JSON.

2. Instalasi VS Code dan PlatformIO

1. **Download dan Install VS Code** dari <https://code.visualstudio.com/>
2. **Buka VS Code**, lalu buka **Extensions Marketplace** (Ctrl+Shift+X).
3. **Cari "PlatformIO"**, lalu **install extension PlatformIO**.
4. **Restart VS Code** setelah instalasi selesai.

3. Membuat Proyek di PlatformIO

1. **Buka VS Code** dan **klik ikon PlatformIO** (di sidebar kiri).
2. **Buat proyek baru** dengan:
 - **Board:** DOIT ESP32 DEVKIT V1
 - **Framework:** Arduino
 - **Nama Proyek:** TrafficLightESP32
3. **Tambahkan library yang diperlukan** dengan perintah:

```
pio lib install "Adafruit GFX Library"
pio lib install "Adafruit ILI9341"
```

4. Menulis Kode ESP32 di VS Code

1. **Buka file src/main.cpp** dan masukkan kode berikut:

```
#include <Arduino.h>
#include <SPI.h>
#include <Adafruit_GFX.h>
#include <Adafruit_ILI9341.h>

// Definisi pin untuk TFT ILI9341
#define TFT_CS 5
#define TFT_RST 4
#define TFT_DC 2

// Inisialisasi layar TFT
Adafruit_ILI9341 tft = Adafruit_ILI9341(TFT_CS, TFT_DC, TFT_RST);

// Definisi pin untuk lampu lalu lintas
const int lampuMerah = 27;
const int lampuKuning = 26;
const int lampuHijau = 33;

void setup() {
  Serial.begin(115200);
  Serial.println("ESP32 Traffic Light dengan TFT ILI9341");

  // Konfigurasi pin output untuk lampu lalu lintas
  pinMode(lampuMerah, OUTPUT);
  pinMode(lampuKuning, OUTPUT);
  pinMode(lampuHijau, OUTPUT);

  // Inisialisasi TFT
  tft.begin();
  tft.setRotation(0); // Atur orientasi portrait
  tft.fillRect(ILI9341_BLACK); // Bersihkan layar awal
}

void tampilkanCountdown(int waktu, uint16_t warna) {
  for (int i = waktu; i >= 0; i--) {
    tft.fillRect(ILI9341_BLACK);
    tft.setTextSize(5);
    tft.setTextColor(warna);
    tft.setCursor(50, 50); // Posisi tengah
    tft.print(i);
    delay(1000);
  }
}

void loop() {
  // Lampu merah (30 detik)
  digitalWrite(lampuMerah, HIGH);
  digitalWrite(lampuKuning, LOW);
  digitalWrite(lampuHijau, LOW);
  tampilkanCountdown(30, ILI9341_RED);

  // Lampu kuning (5 detik)
  digitalWrite(lampuMerah, LOW);
  digitalWrite(lampuKuning, HIGH);
  digitalWrite(lampuHijau, LOW);
  tampilkanCountdown(5, ILI9341_YELLOW);
}
```

```

// Lampu hijau (20 detik)
digitalWrite(lampuMerah, LOW);
digitalWrite(lampuKuning, LOW);
digitalWrite(lampuHijau, HIGH);
tampilkanCountdown(20, ILI9341_GREEN);
}

```

5. Menjalankan Simulasi di Wokwi

1. Buka proyek Wokwi yang sudah dibuat.
2. Tambahkan diagram JSON untuk menghubungkan ESP32 dengan LED dan ILI9341.
3. Klik tombol "RUN" untuk menjalankan simulasi dan melihat hasilnya.

6. Mengupload Kode ke ESP32

1. Hubungkan ESP32 fisik (jika ada) ke laptop.
2. Jalankan perintah upload di PlatformIO:

```
pio run --target upload
```

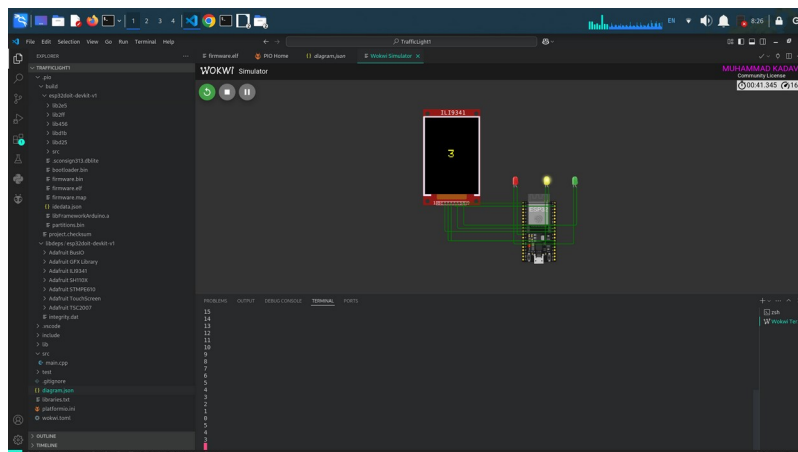
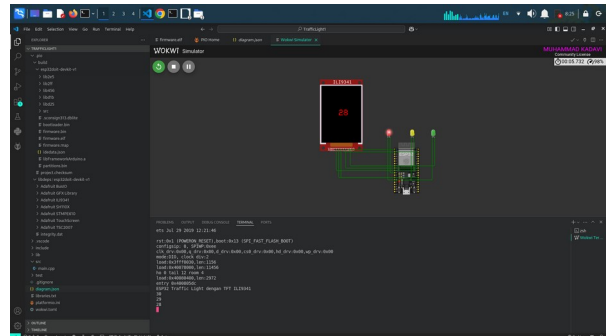
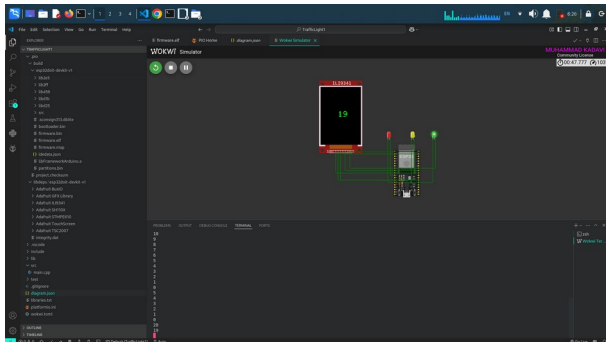
3. Buka Serial Monitor untuk melihat output dengan:

```
pio device monitor
```

3. Results and Discussion (Hasil dan Pembahasan)

Hasil dari simulasi menunjukkan bahwa sistem lampu lalu lintas bekerja sesuai dengan urutan perubahan warna merah, kuning, dan hijau dengan durasi yang telah ditentukan. Lampu merah menyala selama 30 detik, diikuti lampu kuning selama 5 detik, dan lampu hijau selama 20 detik. Selama setiap fase, tampilan countdown pada ILI9341 TFT-LCD menunjukkan angka mundur yang sesuai dengan warna lampu lalu lintas yang sedang menyala.

3.1 Experimental Results (Hasil Eksperimen)



4. Appendix (Lampiran, jika diperlukan)

```
1 #include <Arduino.h>
2 #include <SPI.h>
3 #include <Adafruit_GFX.h>
4 #include <Adafruit_ILI9341.h>
5 >
6 // Definisi pin untuk TFT ILI934
7 #define TFT_CS 5
8 #define TFT_RS 4
9 #define TFT_DC 2
10 #define TFT_MOSI 23
11 #define TFT_SC 18
12 #define KFT_MISO 19 // Tidak digunakan untuk output ke laya
13 e r
14 // Inisialisasi layar TF
15 Adafruit_ILI9341 tft = Adafruit_ILI9341(TFT_CS, TFT_DC, TFT_RS );
16 T
17 // Definisi pin untuk lampu lalu lint
18 const int lampuMerah = 27;
19 const int lampuKuning = 26;
20 const int lampuHijau = 33;
21
22 void setup() {
23   Serial.begin(115200);
24   Serial.println("ESP32 Traffic Light dengan TFT ILI934 ");
25   n 1"
26   // Konfigurasi pin output untuk lampu lalu lint
27 s pinMode (lampuMerah, OUTPUT);
28 pinMode (lampuKuning, OUTPUT);
29 pinMode (lampuHijau, OUTPUT);
30 e
31 // Inisialisasi TF
32 T tft.begin();
33 tft.setRotation(0); // Sesuaikan orientasi layar jika perl
34 tft.fillRect(ILI9341_BLAC ); // Bersihkan layar awa
35 } K l
36
37 // Fungsi untuk menampilkan angka countdown dengan warna tertent
38 void tampilkanCountdown (int waktu, uint16_t warna) {
39   for (int i = waktu; i >= 0; i--) {
40     tft.fillRect(ILI9341_BLAC ); // Bersihkan laya
41     tft.setTextSize(5); // Ukuran angka besa
42     tft.setTextCoLo (warna);
43     r
44     // Posisi angka di tengah laya
45 r int xPos = (tft.width() - 30) / 2; // Menyesuaikan ukuran fon
46 int yPos = (tft.height() - 30) / 2;
47
48     tft.setCursor(xPos, yPos);
49     tft.print(i);
50     Serial.println (String(i));
51     delay(1000);
52   }
53 }
54
55 void loop() {
56   // Lampu merah (30 deti
57 k) digitalWrite (lampuMerah, HIGH);
58 digitalWrite (lampuKuning, LOW);
59 digitalWrite (lampuHijau, LOW);
60 tampilkanCountdown (30, ILI9341_RED);
61 n
62 // Lampu kuning (5 deti
63 k) digitalWrite (lampuMerah, LOW);
64 digitalWrite (lampuKuning, HIGH);
65 digitalWrite (lampuHijau, LOW);
66 tampilkanCountdown (5, ILI9341_YELLOW );
67 n W
68 // Lampu hijau (20 deti
69 k) digitalWrite (lampuMerah, LOW);
70 digitalWrite (lampuKuning, LOW);
71 digitalWrite (lampuHijau, HIGH);
72 tampilkanCountdown (20, ILI9341_GREE );
73 } n N
74
```