

SISTEM KONTROL PENERANGAN RUMAH OTOMATIS BERBASIS IoT MENGUNAKAN APLIKASI BLYNK IoT DAN MIKROKONTROLER ESP32

Yudiansyah Fauzi¹⁾, Rifky Aditia²⁾, Endro Andiono³⁾

^{1,2)}Rekayasa Keamanan Siber Politeknik Piksi Input Serang, Serang, Indonesia

³⁾Manajemen Informatika Politeknik Piksi Input Serang, Serang, Indonesia

*Corresponding author: yudiansyahfauzi@piksiinputserang.ac.id¹⁾, rifkyaditia@piksiinputserang.ac.id²⁾, endro.andiono@gmail.com³⁾

Article history

Received:

dd-mm-yyyy

Accepted:

dd-mm-yyyy

Published:

dd-mm-yyyy

Copyright © 2024
Jurnal Manajemen
dan Teknik

Open Access

Abstrak

Listrik dan sistem penerangan merupakan kebutuhan vital dalam kehidupan rumah tangga modern. Namun, kelalaian pengguna dalam mematikan lampu sering kali memicu pemborosan energi dan peningkatan biaya operasional. Penelitian ini bertujuan untuk merancang dan mengimplementasikan prototipe sistem kontrol penerangan rumah otomatis berbasis Internet of Things (IoT) yang ekonomis dan efisien. Sistem ini menggunakan mikrokontroler ESP32 sebagai pusat kendali utama yang memanfaatkan konektivitas Wi-Fi terintegrasi untuk menerima instruksi digital secara nirkabel. Aplikasi Blynk IoT diterapkan sebagai antarmuka pengguna pada smartphone, memungkinkan pengendalian lampu melalui tombol widget yang terhubung dengan virtual pin V1. Perintah dari aplikasi ditransmisikan melalui jaringan internet menuju ESP32, yang kemudian mengaktifkan modul relay untuk memutus atau menyambung arus listrik pada lampu. Hasil penelitian menunjukkan bahwa prototipe ini berhasil mengontrol perangkat penerangan dari jarak jauh tanpa batasan geografis selama terdapat koneksi internet. Selain itu, pemindahan pin kendali relay dari GPIO 25 ke GPIO 26 terbukti menyelesaikan kendala lampu yang menyala secara otomatis (flicker) saat sistem melakukan booting awal. Sistem ini menawarkan solusi alternatif smart home yang sederhana, berbiaya rendah, serta efektif dalam menekan pemborosan energi akibat faktor kelalaian manusia.

Kata Kunci: Blynk IoT, ESP32, Internet of Things, Pemborosan Energi, Relay, Sistem Penerangan

Abstract

Electricity and lighting systems have become vital necessities in modern household life. However, user negligence in turning off lights often leads to energy waste and increased operational costs. This study aims to design and implement an economical and efficient prototype of an Internet of Things (IoT)-based automatic home lighting control system. The system utilizes the ESP32 microcontroller as the central control unit, leveraging its integrated Wi-Fi connectivity to receive wireless digital instructions. The Blynk IoT application is implemented as the user interface on a smartphone, enabling light control via button widgets connected to virtual pins V1. Commands from the application are transmitted over the internet to the ESP32, which subsequently activates the relay module to connect or disconnect the electrical current to the lights. The results demonstrate that the prototype successfully controls the lighting devices remotely without geographical limitations, provided an internet connection is available. Furthermore, migrating the relay control pin from GPIO 25 to GPIO 26 successfully resolved the issue of the light automatically turning on (flickering) during the initial system booting process. This system offers an alternative smart home solution that is simple, low-cost, and effective in reducing energy waste caused by human error.

Keywords: Blynk IoT, Energy Waste, ESP32, Internet of Things, Lighting System, Relay

1.0 PENDAHULUAN

Listrik dan sistem penerangan sudah menjadi kebutuhan penting dalam kehidupan rumah tangga modern. Namun, penggunaan lampu rumah masih sering menimbulkan pemborosan energi karena kelalaian pengguna, misalnya lupa mematikan lampu ketika meninggalkan rumah atau membiarkan lampu tetap menyala pada saat tidak dibutuhkan. Kondisi tersebut dapat meningkatkan biaya penggunaan listrik dan membuat pemakaian energi menjadi kurang efisien. Penghematan energi tidak hanya berdampak pada berkurangnya tagihan finansial bulanan tetapi juga berkontribusi positif terhadap penurunan emisi karbon global.

Perkembangan teknologi Internet of Things (IoT) memberikan peluang untuk mengendalikan perangkat elektronik dari jarak jauh melalui jaringan internet. Dengan teknologi ini, sakelar manual dapat dikembangkan menjadi sakelar digital yang dapat dikontrol melalui smartphone. Pengguna tidak harus berada di dekat lampu untuk menyalakan atau mematikannya, karena perintah dapat dikirim melalui aplikasi yang terhubung ke perangkat pengendali secara real-time. Smart home merupakan bentuk otomatisasi rumah yang memungkinkan perangkat seperti lampu, sakelar, dan aktuator dikontrol melalui smartphone, mikrokontroler, dan koneksi internet, sehingga fungsi rumah dapat dipantau atau dikendalikan dari lokasi berbeda (Domb, 2019).

Pada penelitian ini digunakan mikrokontroler ESP32 sebagai pusat kendali karena perangkat ini telah memiliki kemampuan komunikasi Wi-Fi dan dapat diprogram untuk membaca perintah dari aplikasi secara efisien. Aplikasi Blynk IoT digunakan sebagai antarmuka kontrol karena menyediakan widget tombol (button) yang dapat dihubungkan dengan mudah menggunakan virtual pin. Perintah dari Blynk dikirim melalui jaringan internet menuju ESP32, kemudian ESP32 mengaktifkan modul relay untuk mengendalikan kondisi fisik lampu rumah. Penggabungan ESP32, modul relay, dan Blynk IoT menawarkan alternatif sistem kontrol penerangan yang sederhana, ekonomis, dan mudah dipahami dalam skala prototipe dibandingkan dengan sistem rumah pintar komersial yang mahal dan kompleks.

1.1 Ruang Lingkup Penelitian

Untuk menjaga fokus penelitian, ruang lingkup dan batasan sistem didefinisikan sebagai berikut:

- **Batasan Perangkat Keras:** Perangkat keras dibatasi pada penggunaan mikrokontroler ESP32 Dev Module, modul relay 1-channel, lampu AC beserta fitting, kabel jumper, dan catu daya micro-USB.

- **Batasan Perangkat Lunak:** Menggunakan Arduino IDE 2.x untuk pengembangan kode program dan pengunggahan firmware, serta platform Blynk IoT (aplikasi seluler) dengan datastream Virtual Pin V1 sebagai antarmuka kendali.

- **Batasan Jaringan dan Data:** Komunikasi data bergantung sepenuhnya pada ketersediaan jaringan Wi-Fi 2.4 GHz lokal untuk ESP32 dan koneksi internet seluler untuk smartphone yang terhubung dengan Blynk Cloud. Penelitian ini tidak menganalisis keamanan jaringan mendalam maupun pengukuran konsumsi daya lampu secara kuantitatif.

1.2 Rumusan Masalah

1. Bagaimana merancang sistem kontrol penerangan rumah otomatis berbasis IoT menggunakan ESP32 dan modul relay?
2. Bagaimana mengintegrasikan aplikasi Blynk IoT dengan ESP32 agar dapat mengendalikan lampu melalui smartphone?
3. Bagaimana hasil pengujian fungsional sistem saat menerima perintah ON dan OFF dari aplikasi Blynk IoT?

1.3 Tujuan Penelitian

1. Merancang prototipe sistem kontrol penerangan rumah otomatis berbasis IoT menggunakan mikrokontroler ESP32 dan modul relay.
2. Membangun antarmuka kontrol lampu pada aplikasi Blynk IoT dengan memanfaatkan Virtual Pin V1.

Pengujian fungsionalitas sistem dalam menyalakan dan mematikan lampu secara nirkabel melalui perintah dari aplikasi Blynk IoT.

1.4 TINJAUAN TEORITIS

Sistem dapat dipahami sebagai kumpulan komponen yang saling berhubungan dan bekerja bersama untuk

mencapai tujuan tertentu. Dalam penelitian ini, sistem terdiri dari pengguna, aplikasi Blynk IoT, jaringan internet, mikrokontroler ESP32, modul relay, dan lampu. Setiap komponen memiliki peran berbeda, tetapi keseluruhannya membentuk satu kesatuan fungsi kontrol penerangan (Rizal et al., 2023). Sistem tidak hanya dipahami sebagai perangkat tunggal, tetapi sebagai susunan terpadu perangkat keras, perangkat lunak, dan infrastruktur komunikasi (Ismail, 2019).

Internet of Things (IoT) merupakan konsep penghubungan perangkat fisik ke jaringan internet agar perangkat tersebut dapat mengirim, menerima, dan memproses data secara otomatis. Pada sistem kontrol penerangan ini, konsep IoT diterapkan pada ESP32 yang terhubung ke internet untuk menerima perintah dari Blynk Cloud. Perintah tersebut kemudian diproses menjadi sinyal kontrol untuk mengaktifkan atau menonaktifkan relay. Komunikasi jaringan pada ESP32 didukung oleh fitur Wi-Fi bawaan yang andal untuk mentransmisikan data perangkat IoT (Espressif Systems, 2024c). Arsitektur IoT yang mencakup perangkat keras, sensor/aktuator, jaringan komunikasi, platform IoT, dan aplikasi pengguna selaras dengan struktur prototipe yang dirancang dalam penelitian ini (Rizal et al., 2023).

Blynk IoT adalah platform populer yang digunakan untuk membuat antarmuka kontrol perangkat IoT melalui smartphone secara praktis. Blynk menyediakan dashboard, template, dan pengaturan datastream yang dapat dikustomisasi (Blynk, 2024a). Datastream berfungsi sebagai saluran komunikasi digital virtual untuk mengirim dan menerima nilai logika antara aplikasi Blynk dan ESP32 (Blynk, 2024b). Pada penelitian ini, widget button dikonfigurasi sebagai switch untuk mengendalikan Virtual Pin V1. Ketika tombol ditekan, status logika 1 (ON) atau 0 (OFF) akan dikirimkan ke cloud (Blynk, 2024c). Penerapan Blynk 2.0 sebagai sistem kontrol penerangan terbukti memiliki latensi rendah dan stabilitas koneksi yang memadai untuk kebutuhan automasi rumah tangga (Herlina dkk., 2022).

Mikrokontroler ESP32 adalah papan pengendali mikro berbiaya rendah dengan performa tinggi yang dilengkapi dengan modul Wi-Fi dan Bluetooth terintegrasi (Espressif Systems, 2024a). ESP32 bertanggung jawab mengolah data masukan dari Blynk dan menghasilkan keluaran logika pada pin GPIO untuk mengontrol aktuator berupa relay. Modul relay merupakan komponen sakelar elektromagnetik yang berfungsi menyambung atau memutuskan aliran daya AC pada lampu berdasarkan sinyal kendali dari ESP32 (Domb, 2019). Relay yang digunakan pada penelitian ini adalah model active-low, yang berarti relay akan aktif dan

menutup sirkuit beban apabila pin pemicu menerima logika LOW dari mikrokontroler. Seluruh kode logika kendali ini ditulis dan dikompilasi menggunakan perangkat lunak Arduino IDE (Arduino, 2024a; Hidayatullah dkk., 2024).

2.0 METODE

Penelitian ini menerapkan metode rekayasa prototipe yang mencakup beberapa tahapan sistematis. Tahapan dimulai dari identifikasi kebutuhan komponen, perancangan skema kelistrikan, konfigurasi platform Blynk IoT, pembuatan kode program pada Arduino IDE, perakitan fisik alat, hingga pengujian fungsionalitas sistem secara langsung.

Tabel 1. Daftar Komponen Perangkat Keras dan Perangkat Lunak

No	Komponen Perangkat	Fungsi dan Kegunaan Utama
1	ESP32 Dev Module	Mikrokontroler pengendali utama, pemroses data, dan penghubung Wi-Fi ke Blynk Cloud.
2	Modul Relay 1-Channel	Sakelar elektronik isolasi listrik AC 220V berbasis sinyal logika DC dari ESP32.
3	Lampu & Fitting AC	Beban keluaran listrik AC yang dikontrol sebagai representasi penerangan rumah.
4	Kabel Jumper & USB	Kabel penghubung antar pin komponen dan jalur catu daya perangkat keras.
5	Aplikasi Blynk IoT	Antarmuka grafis (GUI) berbasis smartphone untuk kontrol nirkabel pengguna.
6	Arduino IDE 2.x	Software editor kode, kompiler, dan uploader firmware ke dalam ESP32.
7	Jaringan Wi-Fi/Internet	Media transmisi data protokol TCP/IP penghubung aplikasi, cloud, dan alat.

2.1 Desain dan Skema Hubungan Perangkat

Skema fungsional hubungan perangkat keras dirancang secara langsung untuk mengintegrasikan ESP32 dengan modul relay aktif-rendah (active-low). Untuk memastikan keandalan sistem saat dihidupkan, pin pemicu relay dihubungkan ke GPIO 26 pada ESP32. Pin catu daya VIN dari ESP32 menyalurkan daya DC sebesar 5V menuju pin VCC pada modul relay, dan pin Ground (GND) keduanya saling terhubung. Di sisi sirkuit beban, modul relay dipasang secara seri pada kabel fasa sirkuit AC menuju lampu. Terminal Common (COM) pada relay dihubungkan ke sumber fasa listrik AC 220V,

sementara terminal Normally Open (NO) disambungkan ke fitting lampu. Desain ini memastikan sirkuit beban lampu tetap terbuka (mati) saat relay tidak aktif.

Alur kerja operasional (proses bisnis) berjalan dari interaksi manusia dengan smartphone hingga perubahan fisik lampu. Ketika pengguna berinteraksi dengan widget switch pada aplikasi Blynk IoT, nilai datastream Virtual Pin V1 berubah (1 untuk ON, 0 untuk OFF). Aplikasi Blynk akan langsung mengirim data tersebut ke Blynk Cloud Server melalui jaringan internet. ESP32, yang secara konstan mendengarkan aktivitas cloud melalui koneksi Wi-Fi lokal, menangkap perubahan nilai tersebut melalui instruksi BLYNK_WRITE(V1). Kode program di dalam ESP32 selanjutnya menerjemahkan nilai tersebut dan mengubah status logika pin GPIO 26. Logika LOW akan memicu koil relay untuk menarik kontak sakelar sehingga lampu menyala, sedangkan logika HIGH akan mematikan relay.

Tabel 2. Rencana Pengujian Sistem Kontrol

No	Aspek Pengujian	Metodologi Pengujian	Indikator Keberhasilan
1	Koneksi ESP32 ke Blynk	Menghidupkan ESP32 dan memeriksa status konektivitas pada Blynk Web Dashboard.	ESP32 berhasil daring dan ikon status perangkat berwarna hijau.
2	Aksi Tombol ON	Menekan widget button di aplikasi Blynk IoT ke posisi aktif (nilai V1 = 1).	Modul relay merespons aktif, sirkuit menutup, dan lampu AC menyala.
3	Aksi Tombol OFF	Menekan widget button di aplikasi Blynk IoT ke posisi nonaktif (nilai V1 = 0).	Modul relay merespons nonaktif, sirkuit membuka, dan lampu AC padam.
4	Kestabilan Booting Pin	Melakukan booting ulang catu daya ESP32 dan mengamati kondisi lampu.	Lampu tidak berkedip (flicker) atau menyala sendiri selama proses inisialisasi boot.

3.0 HASIL DAN PEMBAHASAN

Implementasi antarmuka pada aplikasi Blynk IoT dikonfigurasi melalui pembuatan Datastream Virtual Pin V1 yang diberi nama 'Lampu'. Datastream ini diatur dengan tipe data integer dengan jangkauan nilai minimum 0 dan maksimum 1. Pada Blynk Mobile App, sebuah widget Button ditambahkan ke dalam ruang kerja dashboard kontrol dan dihubungkan ke datastream Lampu tersebut. Mode operasi widget diatur sebagai 'Switch' (sakelar) sehingga status kondisi ON atau

OFF dapat bertahan pada posisinya secara stabil setelah ditekan oleh pengguna, meniadakan perlunya penekanan berulang-ulang.

Di sektor perangkat keras, perakitan fisik dipusatkan pada integrasi modul relay dengan ESP32. Uji fungsionalitas awal menggunakan pin default GPIO 25 memicu kendala operasional yang serius: lampu menyala secara otomatis saat modul ESP32 pertama kali dinyalakan. Masalah ini disebabkan oleh kondisi status pin GPIO 25 yang mengalami floating state (keadaan mengambang) atau memberikan sinyal transien rendah selama proses inisialisasi boot ESP32. Karena modul relay bekerja secara aktif-rendah (active-low), transien logika rendah ini salah diterjemahkan sebagai sinyal aktivasi. Solusi penyelesaian dilakukan dengan memindahkan pin kendali relay ke GPIO 26, yang terbukti mempertahankan kondisi impedansi tinggi (High-Z) yang stabil selama proses booting, serta secara eksplisit menuliskan logika HIGH pada rutin setup() kode program.

Tabel 3. Spesifikasi Koneksi Pin ESP32 dan Modul Relay

No	Koneksi Pin ESP32	Terminal Tujuan Relay	Deskripsi Fungsional Sirkuit
1	VIN (5V DC Output)	Pin VCC Relay	Menyuplai daya operasional sirkuit elektromagnet koil relay.
2	GND (Ground)	Pin GND Relay	Jalur referensi massa bersama (common ground reference).
3	GPIO 26 (Digital Output)	Pin IN Relay	Saluran sinyal kontrol logika untuk memicu transistor relay.
4	Tidak Terhubung Fisik	COM dan NO Relay	Dipasang seri memotong jalur fasa sirkuit AC 220V lampu rumah.

3.1 Implementasi Kode Program Firmware

Pemrograman mikrokontroler ESP32 ditulis menggunakan bahasa C++ pada lingkungan pengembangan Arduino IDE. Kode program berikut mengintegrasikan pustaka BlynkSimpleEsp32.h dan mengonfigurasi logika active-low secara presisi untuk menjamin kestabilan dan keamanan operasional sistem saat booting awal.

```

// Arduino IDE: File -> Preferences
// Arduino IDE: Tools -> Board -> Arduino Uno
// Arduino IDE: Tools -> Port -> COM1

// Pin definitions
const int V1 = 1; // Virtual Pin V1
const int LED = 13; // LED pin
const int RELAY = 26; // Relay pin

// Variables
int V1_state = 0; // V1 state
int LED_state = 0; // LED state
int RELAY_state = 0; // Relay state

// Setup function
void setup() {
  // Initialize V1 pin as input
  pinMode(V1, INPUT);

  // Initialize LED pin as output
  pinMode(LED, OUTPUT);

  // Initialize Relay pin as output
  pinMode(RELAY, OUTPUT);

  // Set initial states
  digitalWrite(LED, LOW); // LED OFF
  digitalWrite(RELAY, LOW); // Relay OFF
}

// Loop function
void loop() {
  // Read V1 state
  V1_state = digitalRead(V1);

  // Control LED and Relay based on V1 state
  if (V1_state == HIGH) {
    // V1 is HIGH (1)
    digitalWrite(LED, HIGH); // LED ON
    digitalWrite(RELAY, LOW); // Relay OFF
  } else {
    // V1 is LOW (0)
    digitalWrite(LED, LOW); // LED OFF
    digitalWrite(RELAY, HIGH); // Relay ON
  }
}

```

4.2 Hasil Analisis Pengujian Fungsional

Pengujian fungsional prototipe dilakukan secara empiris untuk mengevaluasi respons sistem terhadap perintah perubahan status yang dikirim dari smartphone melalui aplikasi Blynk IoT. Respons waktu sirkuit diukur dari saat tombol ditekan hingga koil fisik relay merespons secara mekanis. Berdasarkan data pengujian, seluruh instruksi berhasil dieksekusi secara andal dengan waktu tunda respon rata-rata di bawah satu detik, bergantung pada latensi jaringan internet lokal. **Tabel 4. Matriks Hasil Pengujian Fungsional Sistem**

No	Tombol Blynk	Nilai V1	Logika GPIO 26	Status Relay	Kondisi Lampu	Hasil Uji
1	Ditekan ON	1	LOW (0V)	Aktif / Menutup	Lampu Menyala	Sukses
2	Ditekan OFF	0	HIGH (3.3V)	Nonaktif / Terbuka	Lampu Padam	Sukses

4.0 KESIMPULAN

4.1 Simpulan

1. Sistem kontrol penerangan rumah otomatis berbasis Internet of Things (IoT) telah berhasil dirancang dandiimplementasikan secara sukses menggunakan mikrokontroler ESP32 Dev Module sebagai unit kendali pusat dan modul relay 1-channel sebagai sakelar pemutus daya AC.
2. Integrasi antara aplikasi Blynk IoT dan ESP32 berjalan dengan andal memanfaatkan konektivitas Wi-Fi terintegrasi dan datastream Virtual Pin V1. Aplikasi Blynk pada smartphone mampu mengirimkan nilai kontrol digital yang langsung disinkronkan oleh firmware ESP32 secara real-time.
3. Pengujian fungsional membuktikan efektivitas sistem: perintah ON ($V1 = 1$)

sukses menghasilkan logika LOW pada GPIO 26 yang mengaktifkan relay active-low dan menyalakan lampu, sedangkan perintah OFF ($V1 = 0$) menghasilkan logika HIGH pada GPIO 26 yang menonaktifkan relay dan mematikan lampu.

4. Solusi pemindahan pin kendali dari GPIO 25 ke GPIO 26 berhasil mengatasi masalah ketidakstabilan (lampu menyala otomatis/flicker) saat sistem melakukan proses booting awal, menghasilkan operasi inisialisasi sistem yang aman dan stabil.

4.2 Saran

1. Sistem kontrol penerangan dapat dikembangkan lebih lanjut dengan menambahkan sensor intensitas cahaya lingkungan seperti LDR (Light Dependent Resistor) untuk mewujudkan otomatisasi hibrida (kontrol nirkabel aplikasi dan kontrol otomatis berdasarkan parameter cahaya sekitar).
2. Penelitian berikutnya disarankan menerapkan modul relay multi-channel (misalnya 4-channel atau 8-channel) untuk mengendalikan beberapa sirkuit kelompok lampu rumah secara terpisah (ruang tamu, kamar tidur, dapur, halaman luar) dalam satu antarmuka dashboard Blynk.
3. Meningkatkan faktor keamanan akses sistem kontrol pada platform Blynk dengan menerapkan enkripsi SSL/TLS atau membatasi hak akses otorisasi pengguna smartphone yang terhubung agar sistem terhindar dari penyalahgunaan pihak tidak bertanggung jawab.
4. Disarankan untuk melakukan analisis pengukuran data latensi respons waktu eksekusi secara kuantitatif dalam berbagai variasi kondisi jaringan seluler dan Wi-Fi guna memahami performa fungsionalitas sistem pada beban jaringan yang fluktuatif.

DAFTAR PUSTAKA

Arduino. (2024). Arduino IDE 2 documentation. Arduino Documentation. <https://docs.arduino.cc/software/ide-v2>

- Arduino. (2024). Arduino language reference. Arduino Documentation. <https://docs.arduino.cc/language-reference/>
- Blynk. (2024). Blynk IoT documentation. Blynk Documentation. <https://docs.blynk.io/>
- Blynk. (2024). Datastreams. Blynk Documentation. <https://docs.blynk.io/en/concepts/datastreams>
- Blynk. (2024). Button widget. Blynk Documentation. <https://docs.blynk.io/en/blynk.apps/widgets-controllers/button>
- Domb, M. (2019). Smart home systems based on Internet of Things. Dalam Y. Ismail (Ed.), *Internet of Things (IoT) for Automated and Smart Applications*. IntechOpen. <https://doi.org/10.5772/intechopen.84894>
- Espressif Systems. (2024). ESP32 series datasheet. https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf
- Espressif Systems. (2024). ESP-IDF programming guide: Wi-Fi. <https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-guides/wifi.html>
- Herlina, A., Syahbana, M. I., Gunawan, M. A., & Rizqi, M. M. (2022). Sistem kendali lampu berbasis IoT menggunakan aplikasi Blynk 2.0 dengan modul NodeMCU ESP8266. *INSANtek*, 3(2), 61-66. <https://doi.org/10.31294/instk.v3i2.1532>
- Hidayatullah, R. N., Ramdhan, N. A., & Khamid, A. (2024). Pengembangan kendali lampu menggunakan mikrokontroler NodeMCU ESP32 dan Arduino IDE berbasis Internet of Things (IoT). *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(4), 7762-7767. <https://doi.org/10.36040/jati.v8i4.10461>
- Ismail, Y. (Ed.). (2019). *Internet of Things (IoT) for Automated and Smart Applications*. IntechOpen. <https://doi.org/10.5772/intechopen.77404>
- Rahayu, P., Sularno, & Sari, I. U. (2025). Perancangan sistem smart home berbasis IoT menggunakan ESP32 dan aplikasi Blynk untuk otomatisasi perangkat rumah tangga. *Jurnal Sistem Informasi dan Informatika*, 3(2), 95-100. <https://doi.org/10.47233/jiska.v3i2.2137>
- Rizal, M., Sondakh, D. E., Ashari, I. F., Suryawan, M. A., Mahmudi, A. A., Hidayat M., W., Lontaan, R. J., Agussalim, Rifqie, D. M., Hazriani, Sirmayanti, Lisa, N. P., Supriyanto, B. F., Syamil, A., Hasiri, E. M., Tambi, & Simarmata, J. (2023). *Konsep dan Implementasi Internet of Things*. Yayasan Kita Menulis.
- <https://sipora.polije.ac.id/37500/1/FullBook%20Konsep%20dan%20Implementasi%20Internet%20of%20Things.pdf>