

**RANCANGAN PENGENDALIAN LAMPU
BERBASIS INTERNET OF THINGS
DI BANDAR UDARA INTERNASIONAL SOEKARNO-HATTA**

TUGAS AKHIR

Diajukan untuk Melengkapi Proyek Tugas Akhir dan
Memenuhi Salah Satu Syarat Kelulusan Program Strata 1 (Satu)
Pada Jurusan Teknik Elektro

Disusun Oleh:
DWIKI ADITYA WIBOWO
2114237021



**PROGRAM STUDI TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS SANGGA BUANA YPKP
BANDUNG
2025**

	UNIVERSITAS SANGGA BUANA	FORMULIR LEMBAR PENGESAHAN	
	JL PH.H Mustofa No 68 Bandung 40124	No. Revisi	
		Berlaku Efektif	21 Februari 2025

LEMBAR PENGESAHAN

SKRIPSI

RANCANGAN PENGENDALIAN LAMPU BERBASIS INTERNET OF THINGS DI BANDAR UDARA INTERNASIONAL SOEKARNO-HATTA

LIGHT CONTROL DESIGN BASED ON INTERNET OF THINGS AT SOEKARNO-HATTA INTERNATIONAL AIRPORT

Disusun oleh:

DWIKI ADITYA WIBOWO

2114237021

Telah disetujui dan disahkan sebagai Skripsi Program S1 Teknik Elektro Fakultas

Teknik Universitas Sangga Buana YPKP

Bandung, 21 Februari 2025

Disahkan oleh :

Pembimbing

Dr. Hartuti Mistialustina, S.T., M.

NIK. 432.200.166

	UNIVERSITAS SANGGA BUANA	FORMULIR LEMBAR PENGESAHAN	
	JL PH.H Mustofa No 68 Bandung 40124	No. Revisi	
		Berlaku Efektif	21 Februari 2025

Penguji 1

Penguji 2



Ivany Syarief S.T., M.T.
NIK. 432.200.122



Muhammad Farhan Maulana S.T., M.T.
NIK. 990.000.151

Mengetahui,
Ketua Program Studi



Ivany Sarief, S.T., M.T.
NIK. 432.200.122

	UNIVERSITAS SANGGA BUANA	FORMULIR LEMBAR PENGESAHAN	
	JL PH.H Mustofa No 68 Bandung 40124	No. Revisi	
		Berlaku Efektif	21 Februari 2025

LEMBAR PERNYATAAN ORISINALITAS

Nama : Dwiki Aditya Wibowo
 NPM : 2114237021
 Alamat : Jl. Arwana Blok A5/37 Perumahan Wisma Sooko Indah
 E-mail : adityadwiki456@gmail.com

Menyatakan bahwa Skripsi ini merupakan karya orisinal saya sendiri, dengan judul:

RANCANGAN PENGENDALIAN LAMPU BERBASIS INTERNET OF THINGS DI BANDAR UDARA INTERNASIONAL SOEKARNO-HATTA

LIGHT CONTROL DESIGN BASED ON INTERNET OF THINGS AT SOEKARNO-HATTA INTERNATIONAL AIRPORT

Atas pernyataan ini, saya siap menanggung resiko/sanksi yang dijatuhkan kepada saya apabila di kemudian hari ditemukan adanya pelanggaran terhadap kejujuran akademik atau etika keilmuan dalam karya ini, atau ditemukan bukti yang menunjukkan ketidak aslian.



Bandung, 21 Februari
 2025.

DWIKI ADITYA WIBOWO

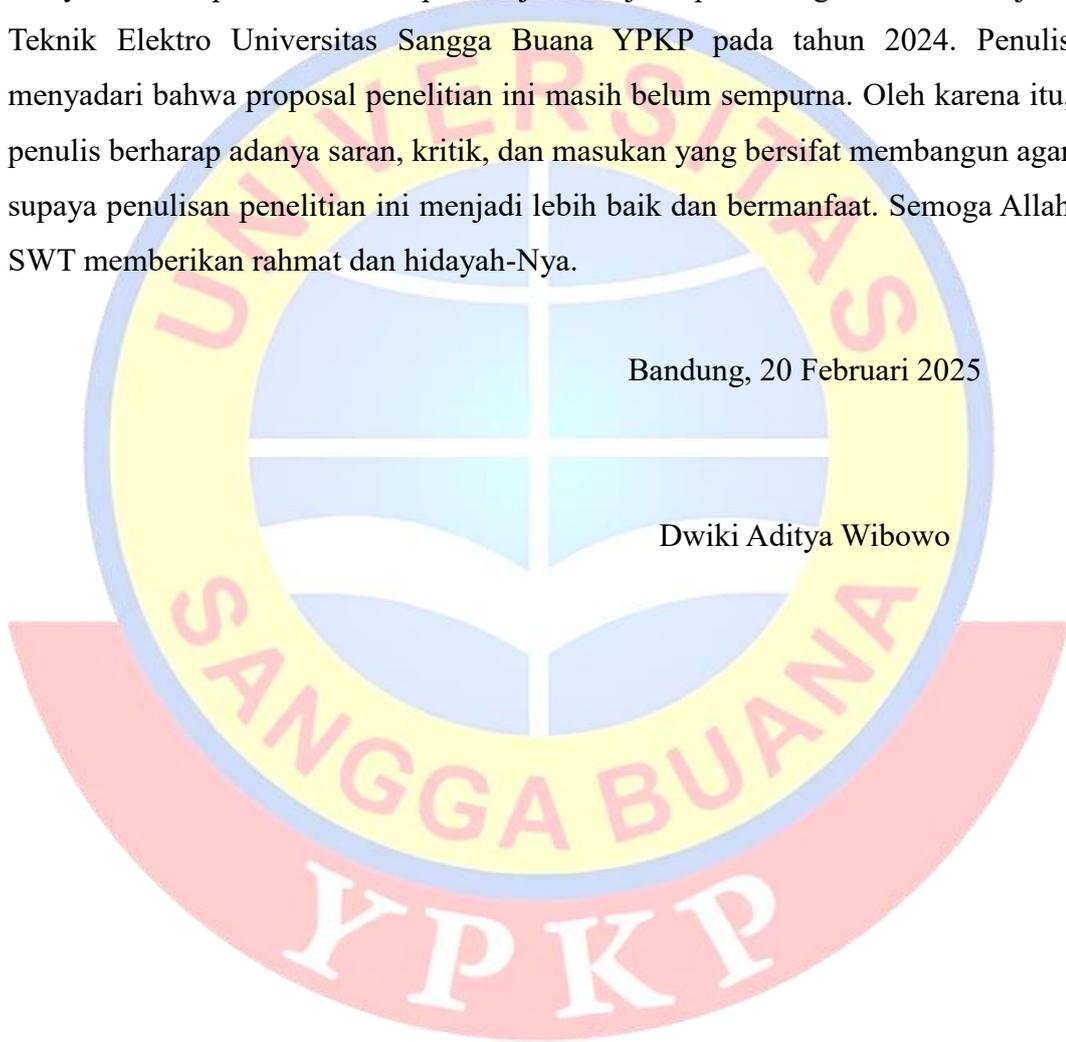
NPM. 2114237021

KATA PENGANTAR

Puji dan syukur ke hadirat ALLAH SWT, atas rahmat dan hidayah – Nya penulis dapat menyelesaikan Proposal Tugas akhir dengan judul “Rancangan Pengendalian Lampu Berbasis *Internet of Things* Di Bandar Udara Internasional Soekarno-Hatta”. Penulis menyusun tugas akhir ini sebagai syarat dalam menyelesaikan pendidikan tahap kemajuan sarjana pada Program Studi Sarjana Teknik Elektro Universitas Sangga Buana YPKP pada tahun 2024. Penulis menyadari bahwa proposal penelitian ini masih belum sempurna. Oleh karena itu, penulis berharap adanya saran, kritik, dan masukan yang bersifat membangun agar supaya penulisan penelitian ini menjadi lebih baik dan bermanfaat. Semoga Allah SWT memberikan rahmat dan hidayah-Nya.

Bandung, 20 Februari 2025

Dwiki Aditya Wibowo



UCAPAN TERIMAKASIH

Dalam proses penyusunan proposal penelitian ini peneliti banyak mendapatkan bantuan, arahan, dan bimbingan dari berbagai pihak. Peneliti mengucapkan terimakasih kepada:

1. Bapak Ivany Sarief, S.T., M.T. Selaku Dosen Wali dan Ketua Program Studi Teknik Elektro di Fakultas Teknik Elektro di Universitas Sangga Buana.
2. Bapak Ketut Abimanyu Munastha, S.T., M.T. Kepala Biro Akademik di Universitas Sangga Buana.
3. Ibu Dr. Hartuti Mistialustina, S.T., M.T. Selaku Dosen Pembimbing, yang selalu memberikan arahan dan masukan untuk selama penulisan ini berlangsung.
4. Dengan penuh rasa hormat dan cinta, saya ingin mengucapkan terima kasih yang sebesar-besarnya kepada kedua orang tua tercinta, yang selalu memberikan doa, dukungan, kasih sayang, dan semangat yang tiada henti dalam setiap langkah perjuangan saya.
5. Istri tercinta Atik Atul Hidayah, kehadirannya adalah sumber inspirasi dan motivasi terbesar bagi saya untuk terus berjuang dan menyelesaikan studi ini
6. Sahabat saya Haikal Islami S.T dan Rekan kerja saya yang telah memberikan dukungan, bantuan, serta motivasi selama proses penelitian ini. Kerjasama, diskusi, dan kebersamaan yang kita jalani telah menjadi bagian yang sangat berarti dalam perjalanan akademik saya.

ABSTRAK

Bandara Soekarno-Hatta merupakan salah satu bandara tersibuk di Asia, yang menghadapi tantangan dalam pengelolaan konsumsi energi, terutama pada sistem penerangan yang beroperasi hampir 24 jam. Penggunaan teknologi *Internet of Things* (IoT) pada sistem kendali penerangan jarak jauh merupakan ide yang cemerlang berdampak untuk solusi ini meningkatkan kinerja dan mengurangi komplain penumpang terkait sistem kendali penerangan di terminal 2 Bandar udara internasional soekarno hatta yang awalnya hanya bisa dikendalikan lewat PC tetapi dengan inovasi ini bisa menjadi solusi jika teknisi sedang tidak ada di ruang *standby* listrik terminal 2 dengan pengelolaan secara efisien hanya melalui layar *smartphone*. Dalam konteks ini, pengembangan sistem kendali penerangan jarak jauh berbasis ESP32. sangat relevan untuk diterapkan di bandara guna mengurangi komplain penumpang di saat cuaca yang tidak pasti entah mendung maupun terang dan meningkatkan efektivitas operasional. Penelitian ini bertujuan untuk merancang dan mengembangkan sistem Kendali penerangan jarak jauh berbasis IoT menggunakan mikrokontroler ESP32, yang dapat mengendalikan penerangan secara *mobile* berdasarkan kondisi cuaca di lapangan. Sistem ini diharapkan dapat meningkatkan efisiensi energi di Bandara Soekarno-Hatta, dengan titik incar penggunaan lampu di area-area kritis seperti terminal penumpang, ruang tunggu, dan area *boarding*. Selain itu, penelitian ini bertujuan untuk mengembangkan sistem yang dapat dipantau dan dikendalikan dari jarak jauh, memudahkan pengelolaan penerangan di area yang luas dan *mobile*. Dan hasil yang dikendalikan dapat lebih responsif yakni tidak sampai 1 detik saat dikendalikan dari *smartphone*.

Kata Kunci:

Sistem Kendali, *Internet of Things* (IoT), mikrokontroler *ESP32*, *mobile*, *smartphone*, lampu.

ABSTRACT

Soekarno-Hatta Airport is one of the busiest airports in Asia, which faces challenges in managing energy consumption, especially the lighting system which operates almost 24 hours. The use of Internet of Things (IoT) technology in remote lighting control systems is a brilliant idea that has an impact on solutions to improve performance and reduce passenger complaints regarding the lighting control system at Terminal 2 of Soekarno Hatta International Airport which initially could only be controlled via PC, but with this innovation could be a solution if the technician is not in the terminal 2 electrical standby room by managing it efficiently just via a smartphone screen. In this context, the development of a remote lighting control system based on ESP32. very relevant to be implemented at airports to reduce passenger complaints during uncertain weather whether cloudy or clear and increase operational effectiveness.

This research aims to design and develop an IoT-based remote lighting control system using an ESP32 microcontroller, which can control lighting mobilely based on weather conditions in the field. This system is expected to increase energy efficiency at Soekarno-Hatta Airport, by targeting the use of lights in critical areas such as the passenger terminal, waiting room and boarding area. In addition, this research aims to develop a system that can be monitored and controlled remotely, making it easier to manage lighting in large areas and mobile. And the controlled results can be more responsive, namely less than 1 second when controlled from a smartphone.

Keywords:

Control System, Internet of Things (IoT), microcontroller ESP32, mobile, smartphone. Lamp.

DAFTAR ISI

UCAPAN TERIMA KASIH	vi
ABSTRAK	vii
ABSTRACT	viii
DAFTAR ISI	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL	xii
BAB I	1
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	3
1.5 Metode Penelitian	4
BAB II	5
LANDASAN TEORI	5
2.1 State Of The Art	5
2.2 Sistem Kendali	7
2.3 ESP 32	7
2.4 Relay	11
2.4.1 Dasar Relai	12
2.4.2 Prinsip Kerja	12
2.4.3 Cara Kerja	14
2.4.4 Fungsi Relai	14
2.5 Internet of Things (IoT)	16
2.5.1 Teknologi Internet of Things (IoT)	17
2.5.2 Prinsip Kerja Internet of Things (IoT)	18
2.5.3 Perangkat IoT (Embedded Device)	18
2.6 Arduino IDE	19
2.7 Blynk	20
BAB III METODE PENELITIAN	21

3.1	Gambaran Umum	21
3.2	Tahapan Penelitian	21
3.3.	Blok Diagram	22
3.4	Diagram Skematik dan Model 3D Sistem.....	23
3.5	Perancangan Hardware.....	25
3.6	Perancangan <i>Software</i>	27
BAB IV HASIL DAN PEMBAHASAN.....		31
4.1	Pengujian Aplikasi <i>Blynk</i>	31
4.1.1	Konektivitas Aplikasi <i>Blynk</i>	31
4.1.2	Respon Waktu Aplikasi <i>Blynk</i>	32
4.1.3	Tampilan Antarmuka Aplikasi <i>Blynk</i>	34
4.2	Pengujian Relai dan Lampu	35
4.1.4	Pengukuran Tegangan pada Setiap Lampu	37
BAB V KESIMPULAN		40
5.1	Kesimpulan	40
5.2	Saran.....	40
DAFTAR PUSTAKA		42
LAMPIRAN		43
DOKUMENTASI PENELITIAN		43
PROGRAM YANG DIGUNAKAN.....		45

DAFTAR GAMBAR

Gambar 1.1(a) dan (b) pengontrolan lampu pada wilayah terminal 2 Bandara Soetta	2
Gambar 2. 1 Diagram State Of The Art.....	5
Gambar 2. 2 Modul ESP32.....	8
Gambar 2. 3 Data Serial ESP32 Dan ESP8266	9
Gambar 2. 4 Kaki-kaki ESP 32.....	10
Gambar 2. 5 Komponen Dalam Relai	13
Gambar 2. 6 Relay.....	13
Gambar 2. 7 Cara Kerja Relay	14
Gambar 2. 8 Topologi Internet Of Things.....	17
Gambar 2. 9 <i>Software</i> Arduino IDE	20
Gambar 3. 1 Tahapan Penelitian	22
Gambar 3. 2 Diagram Blok Sistem.....	22
Gambar 3. 3 Skematik Rangkaian 2D.....	24
Gambar 3. 4 Skematik Rangkaian 2D.....	25
Gambar 3. 5 Maket Gedung Terminal 2 Bandara Soetta Tampak Atas	26
Gambar 3. 6 <i>Flowchart</i> Sistem	28
Gambar 3. 7 Tampilan <i>Blynk</i> di <i>Smartphone</i>	29
Gambar 3. 8 Tampilan <i>Blynk</i> di PC.....	30
Gambar 4. 1 Tampilan Connected dan Disconnected to <i>Blynk</i>	32
Gambar 4. 2 Tampilan <i>Blynk</i> Dekstop Mengikuti Kontrol dari <i>Smartphone</i> .	34

DAFTAR TABEL

Tabel 2. 1 Referensi Jurnal Yang Digunakan	6
Tabel 2. 2 Data Integrasi ESP32	10
Tabel 3. 1 Pemosisian Lampu P dan NP Pada Maket.....	27
Tabel 4. 1 Respon Waktu Nyala/Mati Dari <i>Blynk</i> Ke Lampu	33
Tabel 4. 2 Pengujian nyala lampuan.....	35
Tabel 4. 3 Tabel Pengukuran Tegangan pada LED.....	37



BAB I

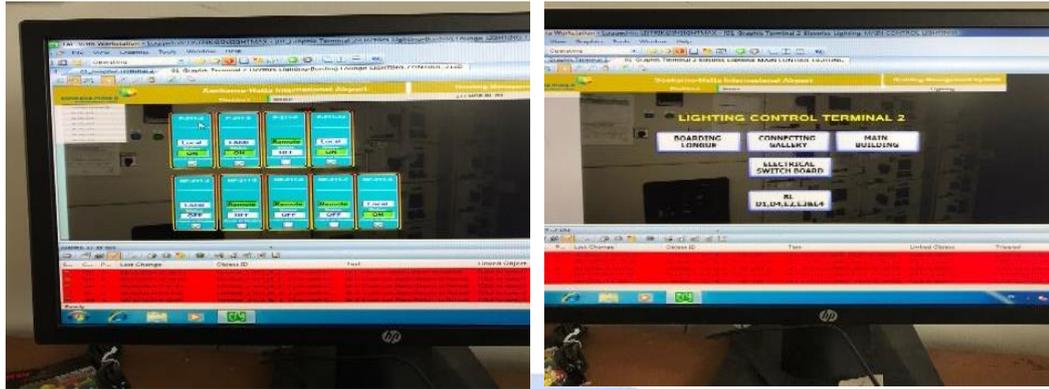
PENDAHULUAN

1.1 Latar Belakang

Bandar Udara Soekarno-Hatta (Bandara Soetta) adalah gerbang udara utama Indonesia, menghubungkan jutaan penumpang domestik dan internasional setiap tahunnya. Sebagai salah satu bandara tersibuk di Asia Tenggara, bandara ini beroperasi hampir 24 jam sehari, 7 hari seminggu. Dalam operasionalnya, bandara ini sangat bergantung pada infrastruktur yang berfungsi secara terus menerus, termasuk sistem penerangan yang tersebar di berbagai area kritis seperti terminal, area *boarding lounge*, parkir area, jalan akses, hingga area keamanan. Penerangan ini tidak hanya berfungsi untuk kenyamanan penumpang, tetapi juga merupakan elemen penting dalam menjaga keamanan dan kelancaran aktivitas sehari-hari di bandara.

Namun, skala besar dan kebutuhan operasional yang tinggi menyebabkan tantangan tersendiri dalam pengelolaan sistem penerangan di Bandara Soetta. Salah satu masalah utama yang dihadapi adalah banyak komplain dari para penumpang akibat kurangnya intensitas cahaya begitu pun sebaliknya akibat dari cuaca yang tidak menentu di lapangan. Penerangan di bandara sering kali tetap menyala sepanjang waktu terutama menjelang malam hari bahkan area-area krusial yang tetap harus menyala 24 jam, seperti area *arrival canopy* atau tempat penumpang keluar, penerangan area apron yang dominan secara langsung terkena cahaya matahari, dan parkir yang jarang digunakan atau bagian terminal yang tidak selalu ramai.

Dalam kondisi saat ini untuk pengontrolan lampu pada wilayah terminal 2 Bandara Soetta ini masih tergolong konvensional karena untuk sistem kendali lampu dikendalikan melalui PC yang ada di ruang *standby* listrik. dengan memakai pc, dan kabel *fiber optic* untuk mengirim perintah ke panel *direct digital control* sebagai kendali lampu-lampu di wilayah terminal 2 Bandara Soetta



(a)

(b)

Gambar 1.1(a) dan (b) pengontrolan lampu pada wilayah terminal 2 Bandara Soetta

Dari gambar 1.1, wilayah kendali lampu terminal 2 sendiri terdapat wilayah *boarding lounge*, *main building* dan lainnya, di mana lampu dikendalikan dengan metode *priority* maupun *non priority*. Contoh kecil jika wilayah *boarding* terdapat 100 lampu. Yang dikendalikan untuk *priority* hanya 40 lampu dan sisanya 60 lampu merupakan *non priority* tergantung dengan cuaca di lapangan untuk lampu wilayah tersebut. Dengan begitu Penerapan pengendalian lampu berbasis *Internet of Things* (IoT) pada seluruh wilayah khususnya di terminal 2 di Bandara Soekarno-Hatta dapat memberikan solusi terkini untuk masalah ini. Dengan sistem berbasis IoT, dalam pengendalian lampu melalui *smartphone* yang dibagi menjadi 2 yaitu lampu *priority* maupun *non priority* yang dapat dikendalikan secara langsung tanpa harus adanya operator ataupun teknisi di ruang *standby* listrik. Dengan adanya perangkat digital yaitu *smartphone* yang memungkinkan tim teknis mendapatkan akses yang mudah dan secara langsung jika terjadi cuaca yang tidak menentu. Hal ini akan sangat membantu dalam mencegah adanya komplain baik dari penumpang maupun pihak atasan, mengoptimalkan penggunaan daya, dan menjaga efisiensi operasional bandara secara keseluruhan. Selain itu, sistem berbasis IoT juga memungkinkan pengelolaan energi yang lebih cerdas dan berkelanjutan, sejalan dengan tuntutan lingkungan dan efisiensi energi di fasilitas-fasilitas besar seperti fasilitas area terminal 2 Bandara Soetta. Dengan pengendalian yang cepat dengan memanfaatkan jaringan internet, risiko komplain dan kurangnya intensitas cahaya di lapangan yang dapat mengganggu operasional bandara dapat dikurangi, serta memastikan kenyamanan dan keamanan bagi semua pengguna jasa bandara. Penelitian ini

bertujuan untuk merancang pengendalian lampu berbasis IoT di terminal 2 Bandara Soetta. Dengan adanya sistem kendali tersebut, diharapkan dapat meningkatkan keandalan dan efisiensi operasional pada sistem penerangan lampu di mana pun operator berada saat bertugas dinas.

1.2 Rumusan masalah

1. Bagaimana cara pengontrolan penerangan dari jarak jauh?
2. Apakah metode pengontrolan dengan menggunakan *smartphone* dapat meningkatkan efisiensi penggunaan pada sistem penerangan?
3. Dengan metode apakah sistem pengontrolan penerangan tersebut dapat dilakukan?

1.3 Batasan Masalah

1. Inovasi penelitian hanya untuk pengontrolan penerangan wilayah sisi krusial gedung terminal 2 Bandar Udara Soekarno Hatta
2. Media pengendalian lampu dilakukan dengan *smartphone*
3. Pengontrolan jarak jauh dari layar *smartphone* difokuskan jika tidak ada teknisi di ruang *standby* listrik.
4. Prototipe ini bisa menjadi opsi ke 2 jika kendali penerangan melalui tidak bekerja dengan efektif.
5. Pengontrolan *prototype* ini difokuskan hanya pengendalian ON/OFF lampu.

1.4 Tujuan Penelitian

Sistem pengontrolan tersebut dapat memudahkan para teknisi listrik jika berada di luar ruangan bisa mengontrol lewat *smartphone* dengan memanfaatkan teknologi dari *Blynk* dengan memanfaatkan jaringan internet sehingga bisa lebih efisien dalam bekerja dan dapat mengurangi komplain dalam segala aspek

1. Memberikan kemudahan dalam akses kendali jarak jauh.
2. Meningkatkan dan mengoptimalkan pemakaian energi.
3. Meningkatkan efisiensi dalam peningkatan kegiatan operasional.
4. Memudahkan petugas lapangan dalam melakukan kegiatan operasional.

1.5 Metode Penelitian

Metode yang digunakan dalam penyusunan Tugas Akhir ini terdiri dari: 1. Metode Perpustakaan Dengan mencari referensi buku, jurnal yang berhubungan dengan tugas akhir sesuai dengan judul yang dirujuk. 2. Metode Analisa Merancang, membangun dan menganalisis alat yang dibuat. 3. Metode Eksperimen Melakukan percobaan pengukuran dan menganalisis hasil pengendalian lampu berbasis IoT.

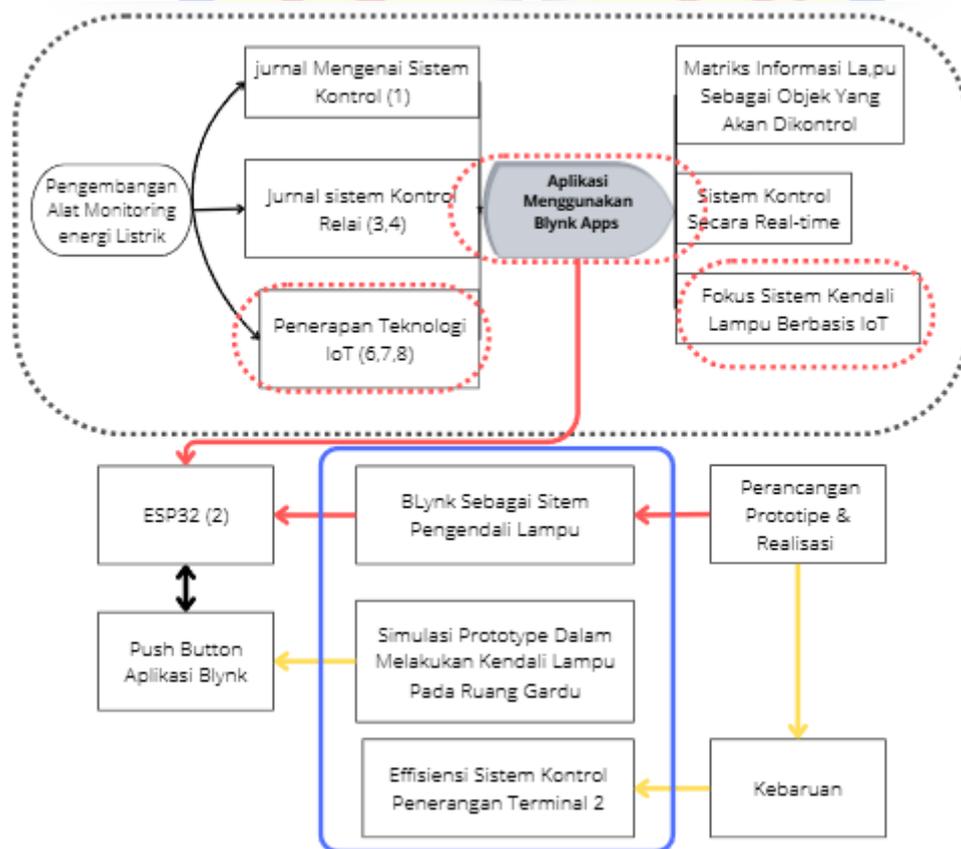


BAB II

LANDASAN TEORI

2.1 State Of The Art

State of the art dalam konteks penelitian berfungsi sebagai elemen penting untuk memahami dan merumuskan posisi penelitian Anda dalam ranah ilmiah yang lebih luas. *State of the art* terlihat pada penggunaan jurnal terkait sistem kendali, penerapan teknologi IoT dan sistem pengendalian penerangan pada terminal 2 Bandara Soetta menggunakan aplikasi *Blynk*.



Gambar 2. 1 Diagram *State Of The Art*

Tabel 2. 1 Referensi Jurnal Yang Digunakan

Judul Jurnal	Penulis	Tahun Terbit	Metode Penelitian	Sistem Kerja
Sistem Pantau dan Kendali Peralatan Listrik Rumah Berbasis Internet of Things (IoT)	F. Farid and N. S. Salahuddin	2024	Kualitatif	teknologi IoT untuk memonitor dan mengontrol peralatan listrik secara jarak jauh melalui perangkat terhubung ke internet, seperti <i>smartphone</i> atau komputer
Pendeteksi Kehadiran Menggunakan Esp32 Untuk Sistem Pengunci Kakitu Otomatis	A. Prafanto, E. Budiman, P. P. Widagdo, G. Mahendra Putra, R. Wardhana, and U. Mulawarman	2021	Kuantitatif	mengontrol kunci kakitu secara manual dari jarak jauh.
Sistem Kendali Lampu pada Smart Home Berbasis IoT (Internet of Things)	D. Susilo, C. Sari, and G. W. Krisna	2021	Kualitatif	Mengendalikan Lampu pada <i>Smart Home</i> Berbasis IoT (Internet of Things)
Rancang Bangun Sistem Keamanan Rumah Menggunakan Relai	M. Saleh and M. Haryanti,	2017	Kualitatif	Sistem keamanan rumah berbasis relai dirancang untuk mengontrol perangkat keamanan seperti kunci kakitu, alarm, atau kamera pengawas secara otomatis.
Penerapan Internet of Things (IoT) Pada Sistem Kendali Lampu Berbasis Mobile	R. Muzawi, W. Joni Kurniawan	2018	Kualitatif	<i>Platform</i> IoT memperbarui informasi di aplikasi mobile sehingga pengguna dapat memantau status lampu secara real-time.
Internet of Thing Sistem Pengendalian Lampu Jarak Jauh Menggunakan Nodemcu Amica CP2102 berbasis Mobile	R. Apriza Dini	2021	Kualitatif	Sistem pengendalian lampu jarak jauh berbasis Internet of Things (IoT) menggunakan NodeMCU Amica CP2102 dan aplikasi mobile bekerja dengan memanfaatkan teknologi IoT untuk mengontrol lampu secara remote melalui perangkat mobile.
Aplikasi Arduino Ide Pada Mata Kuliah Sistem Digital	U. Mahanin Tyas, A. Apri Buckhari, P.	2023	Kualitatif	pengembangan perangkat lunak yang digunakan untuk memprogram mikroendalier Arduino
Perancangan Sistem Monitoring Dan	N. Imamah and D. Sagara Andika	2021	Kualitatif	mengontrol lampu secara otomatis berdasarkan

Judul Jurnal	Penulis	Tahun Terbit	Metode Penelitian	Sistem Kerja
Pengendalian Lampu Menggunakan Sensor Gerak Dan Sensor Cahaya Dilengkapi Internet Of Things (Iot)				deteksi gerak dan intensitas cahaya di sekitarnya
Rancangan pengendalian lampu berbasis internet of things di bandar udara internasional soekarno-hatta	Dwiky Aditya Wibowo	2024	Kuantitatif	Simulasi prototype teknologi IoT untuk sistem monitoring daya dan kendali lampu dan menguji efisiensi sistem kendali dan monitoring dalam aplikasi nyata.

2.2 Sistem Kendali

Sistem kendali adalah suatu sistem yang di gunakan untuk mengatur atau mengubah suatu sistem jaringan listrik yang sudah ada menjadi sistem jaringan listrik yang di inginkan. Sistem kendali di bagi menjadi dua, yaitu sistem kendali secara manual dan secara otomatis. Berikut penjelasannya:

1. Sistem Kendali Manual

Sistem kendali manual adalah sistem pengendalian dengan subjek makhluk hidup, contoh oleh manusia. Biasanya sistem ini di pakai pada beberapa proses-proses yang tidak banyak mengalami perubahan beban (*load*) atau pada proses yang tidak kritis.

2 Sistem Kendali *Remote*

Sistem kendali otomatis adalah sistem pengendalian di mana subjek digantikan oleh suatu alat yang disebut kendalier. Di mana tugas untuk membuka dan menutup *valve* tidak lagi dikerjakan oleh operator, tetapi atas perintah kendalier.

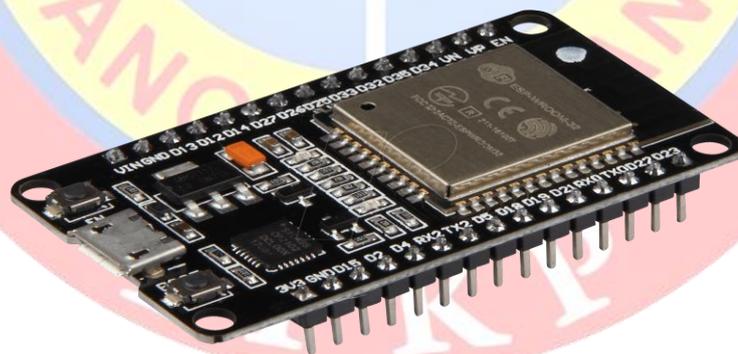
2.3 ESP 32

Node MCU ESP32 merupakan sebuah *opensource platform IoT* dan pengembangan kit yang menggunakan bahasa pemrograman *Lua* untuk membantu *programmer* dalam membuat *prototype* produk *IoT* atau bisa dengan memakai *sketch* dengan Arduino IDE. Pengembangan *Kit* ini didasarkan pada modul ESP32,

yang mengintegrasikan GPIO, PWM (*Pulse Width Modulation*), IIC , *1-Wire* dan ADC (*Analog to Digital Converter*) semua dalam satu *board*.

Keunikan dari Node MCU ESP32 ini sendiri yaitu memiliki ukuran yang sangat kecil yaitu panjang 4.83cm, lebar 2.54cm, dan dengan berat 7 gram. Tapi walaupun ukurannya yang kecil, *board* ini sudah dilengkapi dengan fitur *wifi* dan *firmware*nya yang bersifat *opensource*. Penggunaan Node MCU ESP32 lebih menguntungkan dari segi biaya maupun efisiensi tempat, karena Node MCU ESP32 yang ukurannya kecil, lebih praktis dan harganya jauh lebih murah dibandingkan dengan Arduino Uno.

Arduino Uno sendiri merupakan salah satu jenis mikrokontroler yang banyak diminati dan memiliki bahasa pemrograman C++ sama seperti Node MCU ESP32 namun Arduino Uno belum memiliki modul *wifi* dan belum berbasis *IoT*. Untuk dapat menggunakan *wifi* Arduino Uno memerlukan perangkat tambahan berupa *wifi shield*. Node MCU ESP32 merupakan salah satu produk yang mendapatkan hak khusus dari Arduino untuk dapat menggunakan aplikasi Arduino sehingga bahasa pemrograman yang digunakan sama dengan board Arduino pada umumnya [2].



Gambar 2. 2 Modul ESP32

ESP32 merupakan penerus dari *module* ESP8266. Pada ESP32 terdapat inti CPU serta *Wifi* yang lebih cepat, GPIO yang lebih, dan mendukung *Bluetooth* 4.2 konsumsi daya yang rendah. Tabel di bawah menunjukkan perbedaan utama antara ESP 32 dan ESP8266.

VARIANS	ESP8266	ESP32
PICTURE MODULE		
MCU	Xtensa Single-core 32-bit L106	Xtensa Dual-Core 32-bit LX6 with 600 DMIPS
Wi-Fi	802.11 b/g/n tipe HT20	802.11 b/g/n tipe HT40
Bluetooth	Tidak ada	tipe 4.2 dan BLE
Typical Frequency	80 MHz	160 MHz
SRAM	Tidak ada	ada
Total GPIO	17	36
Total SPI-UART-I2C-I2S	2-2-1-2	4-2-2-2
Resolusi ADC	10 bit	12 bit
Suhu operasional Kerja	-40°C to 125°C	-40°C to 125°C
Sensor di dalam module	Tidak ada	touch sensor, temperature sensor, hall effect sensor
Harga di pasaran	Rp. 30.000 – 350.000	Rp. 100.000 – 650.000

Gambar 2. 3 Data Serial ESP32 Dan ESP8266

Dikarenakan pada Node MCU ESP32 memiliki lebih banyak kaki GPIO dibanding dengan ESP8266, maka dapat diputuskan kaki mana yang akan difungsikan sebagai UART, I2C, atau SPI sesuai dengan kode program yang akan dibuat. Selain itu memungkinkan karena terdapat fitur *multiplexing* pada *chip* Node MCU ESP32 yang memungkinkan untuk menggunakan beberapa fungsi pada kaki yang sama. Akan tetapi jika tidak diprogram untuk fungsi khusus, maka kaki GPIO berfungsi secara *default*, seperti gambar 2.4. Untuk dapat mengatur sinyal PWM pada GPIO mana pun dengan frekuensi yang dikonfigurasi dan siklus yang diatur pada program kode. Pada Node MCU ESP32 juga mendukung dua saluran DAC 8-bit. Selain itu, Node MCU ESP32 berisi 10 GPIO penginderaan kapasitif, yang mendeteksi sentuhan dan dapat digunakan untuk memicu peristiwa, atau mengaktifkan ESP32 dari *mode sleep* Untuk spesifikasi lengkap dari Node MCU ESP32 dapat dilihat dari tabel dibawah ini.

Node MCU ESP32 mengintegrasikan dua ADC 12-bit SAR (“Register Berurutan”) (Pengkonversi *Analog* ke *Digital*) dan mendukung pengukuran pada 18 saluran (kaki berkemampuan *analog*). Beberapa kaki ini dapat digunakan untuk membangun *amplifier* yang dapat diprogram yang digunakan untuk pengukuran sinyal analog kecil. API driver ADC saat ini hanya mendukung ADC1 (9 saluran, dilampirkan ke GPIO 32-39).

Tegangan skala penuh adalah tegangan yang sesuai dengan pembacaan maksimum (tergantung pada lebar bit yang dikonfigurasi ADC1, nilai ini adalah: 4095 untuk 12-bit, 2047 untuk 11-bit, 1023 untuk 10-bit, 511 untuk 9 bit.) Nilai yang dibaca pada ADC ESP32 adalah 12 bit (kisaran 0-4095).

2.4 Relai

Relai adalah suatu peranti yang bekerja berdasarkan elektromagnetik untuk menggerakkan sejumlah kontaktor yang tersusun atau sebuah saklar elektronis yang dapat dikendalikan dari rangkaian elektronik lainnya dengan memanfaatkan tenaga listrik sebagai sumber energinya. Kontaktor akan tertutup (menyala) atau terbuka (mati) karena efek induksi magnet yang dihasilkan kumparan (induktor) ketika dialiri arus listrik. Berbeda dengan saklar, pergerakan kontaktor (*on* atau *off*) dilakukan manual tanpa perlu arus listrik.

Relai yang paling sederhana ialah relai elektro mekanis yang memberikan pergerakan mekanis saat mendapatkan energi listrik. Secara sederhana relai elektro mekanis ini didefinisikan sebagai berikut.

1. Alat yang menggunakan gaya elektromagnetik untuk menutup atau membuka kontak saklar.
2. Saklar yang digerakkan secara mekanis oleh daya atau energi listrik.

Sebagai komponen elektronika, relai mempunyai peran penting dalam sebuah sistem rangkaian elektronika dan rangkaian listrik untuk menggerakkan sebuah perangkat yang memerlukan arus besar tanpa terhubung langsung dengan perangkat pengendali yang mempunyai arus kecil. Dengan demikian relai dapat berfungsi sebagai pengaman.

Relai terdiri dari 3 bagian utama, yaitu:

1. *Common*, merupakan bagian yang tersambung dengan *Normally Close* (dalam keadaan normal).
2. *Coil* (kumparan), merupakan komponen utama relai yang digunakan untuk menciptakan medan magnet.
3. Kontak, yang terdiri dari *Normally Close* dan *Normally Open*.

2.4.1 Dasar Relai

Penemu relai pertama kali adalah Joseph Henry pada tahun 1835. Dalam pemakaiannya biasanya relai yang digerakkan dengan arus DC dilengkapi dengan sebuah dioda yang diparalel dengan lilitannya dan dipasang terbalik yaitu anoda pada tegangan (-) dan katoda pada tegangan (+). Ini bertujuan untuk mengantisipasi sentakan listrik yang terjadi pada saat relai berganti posisi dari *on* ke *off* agar tidak merusak komponen di sekitarnya.

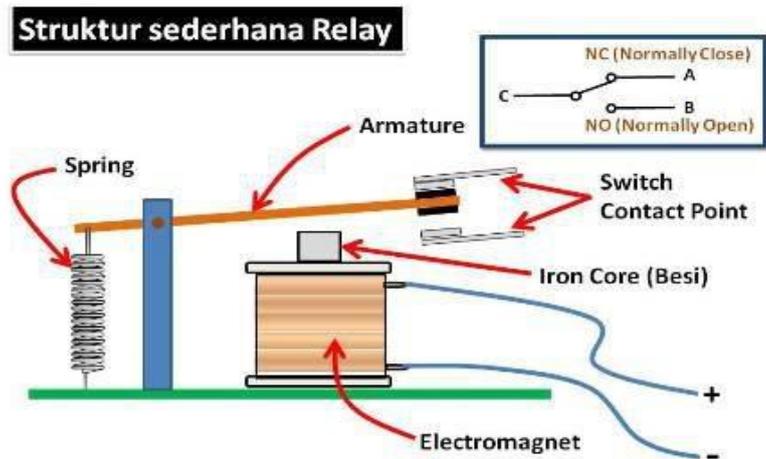
Penggunaan relai perlu memperhatikan tegangan pengontrolnya serta kekuatan relai men-*switch* arus/tegangan. Biasanya ukurannya tertera pada *body* relai. Misalnya relai 12 VDC/4 A 220V, artinya tegangan yang diperlukan sebagai pengontrolnya adalah 12 Volt DC dan mampu men-*switch* arus listrik (maksimal) sebesar 4 ampere

ada tegangan 220 Volt. Sebaiknya relai difungsikan 80% saja dari kemampuan maksimalnya agar aman, lebih rendah lagi lebih aman. Relai jenis lain ada yang namanya *reedswitch* atau relai lidi. Relai jenis ini berupa batang kontak terbuat dari besi pada tabung kaca kecil yang dililitin kawat. Pada saat lilitan kawat dialiri arus, kontak besi tersebut akan menjadi magnet dan saling menempel sehingga menjadi saklar yang *on*. Ketika arus pada lilitan dihentikan medan magnet hilang dan kontak kembali terbuka (*off*).

2.4.2 Prinsip Kerja

Relai merupakan komponen listrik yang memiliki prinsip kerja magnet dengan induksi listrik. Relai terdiri atas bagian-bagian utama sebagai berikut.

1. *Coil* atau Kumparan, merupakan gulungan kawat yang mendapat arus listrik. adalah sejenis saklar yang pergerakannya tergantung dari ada tidaknya arus listrik di *coil*.



Gambar 2. 5 Komponen Dalam Relai

Contact atau Penghubung, adalah sejenis saklar yang pergerakannya tergantung dari ada tidaknya arus listrik di *coil*. *Contact* ada 2 jenis: *Normally Open* (kondisi awal sebelum diaktifkan open), dan *Normally Closed* (kondisi awal sebelum diaktifkan close). Prinsip kerja relai adalah sebagai berikut :

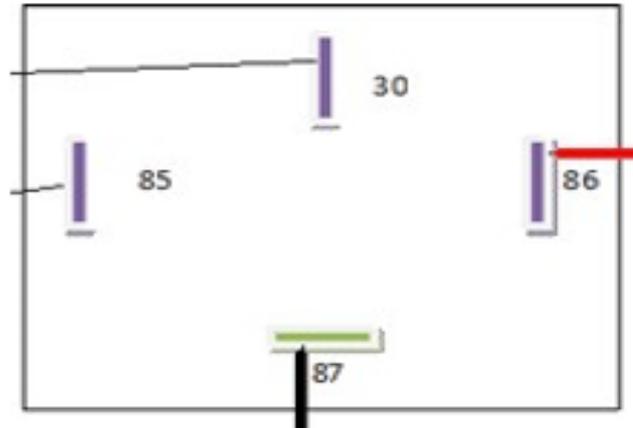


Gambar 2. 6 Relai

Saat *coil* mendapatkan energi listrik (*energized*) akan menimbulkan gaya elektromagnetik. Gaya magnet yang ditimbulkan akan menarik plat/lengan kontak (*armature*) berpegas (bersifat berlawanan), sehingga menghubungkan 2 titik *contact*.^[4]

2.4.3 Cara Kerja

Cara kerja relai sangat sederhana. Di sini kita akan membahas relai pada umumnya.



Gambar 2. 7 Cara Kerja Relai

Relai terdiri dari 2 terminal *trigger*, 1 terminal input dan 1 terminal output.

1. Terminal *trigger* : yaitu terminal yang akan mengaktifkan relai. Seperti alat *electronic* lainnya relai akan aktif apabila di aliri arus + dan arus. Pada contoh relai yang kita gunakan terminal *trigger* ini adalah 85 dan 86.
2. Terminal *input* : yaitu terminal tempat kita memberikan masukan pada contoh adalah terminal 30
3. Terminal *output* : yaitu tempat keluarnya *output* pada contoh adalah terminal 87

2.4.4 Fungsi Relai

Fungsi atau kegunaan relai dalam dunia elektronika sebenarnya juga sama seperti dalam teknik listrik. Hanya saja kebanyakan relai yang digunakan dalam teknik elektronika adalah relai dengan voltase kecil seperti 6 Volt, 12 Volt, 24 Volt berbeda dengan teknik listrik yang memakai relai 220 Volt dan 110 Volt. Namun ada juga dalam teknik elektronika yang memakai relai dengan voltase tinggi. Walau ada perbedaan pemakaian voltase pada relai, sebenarnya relai memiliki fungsi atau kegunaan yang sama yakni sebagai alat pengganti saklar yang bekerja untuk mengontrol atau membagi arus listrik ataupun sinyal lain ke sirkuit rangkaian lainnya.

Secara garis besar, fungsi relai adalah sebagai berikut.

- A. Kendali tegangan tinggi rangkaian dengan sinyal bertegangan rendah, seperti dalam beberapa jenis modem atau *audio amplifier*.
- B. Kendali sebuah rangkaian arus tinggi dengan sinyal arus rendah, seperti pada *solenoid starter* dari sebuah mobil.
- C. Mendeteksi dan mengisolasi kesalahan pada jalur transmisi dan distribusi dengan membuka dan menutup pemutus rangkaian (perlindungan relai).
- D. Sebuah kumparan relai DPDT AC dengan kemasan "*ice cube*".
- E. Isolasi mengendalikan rangkaian dari rangkaian yang dikendali ketika kedua berada pada potensi yang berbeda, misalnya ketika mengendalikan sebuah perangkat bertenaga utama dari tegangan rendah *switch*. Yang terakhir ini sering digunakan untuk mengontrol pencahayaan kantor sebagai kawat tegangan rendah dapat dengan mudah diinstal di partisi, yang dapat dikakidahkan sesuai kebutuhan sering berubah. Mereka mungkin juga akan dikendalikan oleh hunian kamar detektor dalam upaya untuk menghemat energi.
- F. Logika fungsi. Sebagai contoh, DAN fungsi *boolean* direalisasikan dengan menghubungkan relai normal kontak terbuka secara seri, maka fungsi atau dengan menghubungkan normal kontak terbuka secara paralel. Perubahan atas atau Formulir C kontak melakukan XOR fungsi. Fungsi yang sama untuk NAND dan NOR yang dicapai dengan menggunakan kontak normal tertutup. Tangga bahasa pemrograman yang sering digunakan untuk merancang jaringan logika relai.
- G. Awal komputasi, Sebelum tabung vakum dan transistor, relai digunakan sebagai unsur-unsur logis dalam komputer digital.
- H. *Safety* logika kritis. Karena relai jauh lebih tahan daripada semikonduktor radiasi nuklir, mereka banyak digunakan dalam keselamatan logika kritis, seperti panel kendali penanganan limbah radioaktif mesin.
- I. Waktu tunda fungsi. Relai dapat dimodifikasi untuk menunda pembukaan atau penutupan menunda satu set kontak. Yang sangat singkat (sepersekian detik) penundaan ini akan menggunakan tembaga *disk* antara angker dan bergerak blade perakitan. Arus yang mengalir dalam *disk* mempertahankan medan

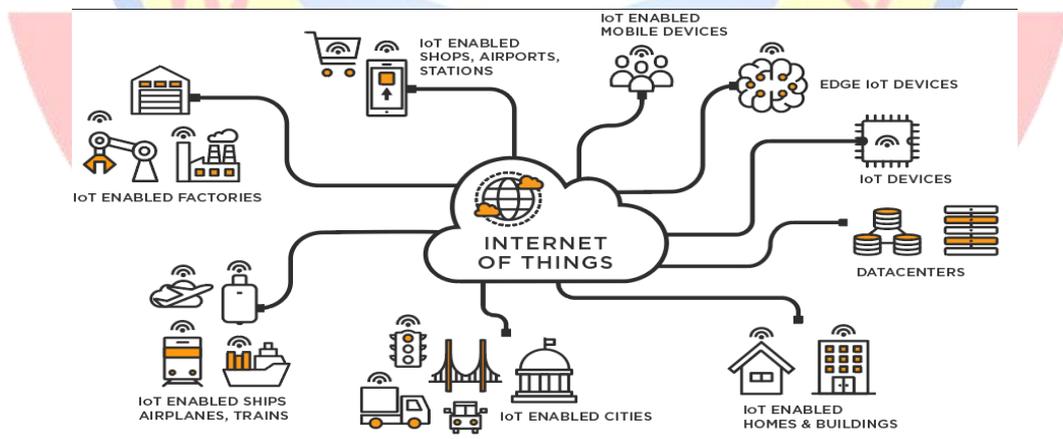
magnet untuk waktu yang singkat, memperpanjang waktu rilis. Untuk sedikit lebih lama (sampai satu menit) keterlambatan, sebuah *dashpot* digunakan. Sebuah *dashpot* adalah sebuah piston diisi dengan cairan yang diperbolehkan untuk melarikan diri perlahan lahan. Jangka waktu dapat divariasikan dengan meningkatkan atau menurunkan laju aliran. Untuk jangka waktu lebih lama, mesin jam mekanik timer diinstal. [5]

2.5 Internet of Things (IoT)

Internet of Things (IOT) adalah struktur di mana objek, orang disediakan dengan identitas eksklusif dan kemampuan untuk kakidah data melalui jaringan tanpa memerlukan dua arah antara manusia ke manusia yaitu sumber ke tujuan atau interaksi manusia ke komputer. *Internet of Things* merupakan perkembangan keilmuan yang sangat menjanjikan untuk mengoptimalkan kehidupan berdasarkan sensor cerdas dan peralatan kakitar yang bekerjasama melalui jaringan internet. Sejak mulai dikenalnya internet pada tahun 1989, mulai banyak hal kegiatan melalui internet, Pada tahun 1990 John Romkey menciptakan 'perangkat', pemanggang roti yang bisa dinyalakan dan dimatikan melalui Internet. *WearCam* diciptakan pada tahun 1994 oleh Steve Mann. Pada tahun 1997 Paul Saffo memberikan penjelasan singkat pertama tentang sensor dan masa depan. Tahun 1999 Kevin Ashton menciptakan *The Internet of Things*, direktur eksekutif Auto IDCentre, MIT. Mereka juga menemukan peralatan berbasis RFID (*Radio Frequency Identification*) global yang sistem identifikasi pada tahun yang sama. Penemuan ini disebut sebagai sebuah lompatan besar dalam *commercializing IoT*. Tahun 2000 LG mengumumkan rencananya menciptakan kulkas kakitar yang akan menentukan sendiri apakah bisa atau tidak makanan yang tersimpan di dalamnya diisi ulang. Pada tahun 2003 RFID mulai ditempatkan pada tingkat besar besaran di militer AS di Program Savi mereka. Pada tahun yang sama melihat raksasa ritel Walmart untuk menyebarkan RFID di semua toko di seluruh dunia untuk lebih besar batas. Pada tahun 2005 arus publikasi utama seperti *The Guardian*, Amerika ilmiah dan Boston Globe mengutip banyak artikel tentang *IoT*.

Pada tahun 2008 kelompok perusahaan meluncurkan IPSO Alliance untuk mempromosikan penggunaan *Internet Protocol (IP)* dalam jaringan dari "*Smart object*" dan untuk mengaktifkan *Internet of Things*. Pada tahun 2008 FCC

menyetujui penggunaan “*white space spectrum*”. Akhirnya peluncuran IPv6 di tahun 2011 memicu pertumbuhan besar di bidang *Internet of Things*, perkembangan ini didukung oleh perusahaan raksasa seperti Cisco, IBM, Ericson mengambil inisiatif banyak dari pendidikan dan komersial dengan *IoT* teknologi dapat hanya dijelaskan sebagai hubungan antara manusia dan komputer. Perkembangan *Internet of Things*, semua peralatan yang kita gunakan dalam kehidupan kita sehari hari dapat dikendalikan dan dipantau menggunakan *IoT*. Mayoritas proses dilakukan dengan bantuan sensor di *IoT*. Sensor dikerahkan di mana mana dan sensor ini mengkonversi data fisik mentah menjadi sinyal digital dan mengirimkan mereka ke pusat kendali. Dengan cara ini kita bisa memonitor perubahan lingkungan jarak jauh dari setiap bagian dari dunia melalui internet. Arsitektur sistem ini akan didasarkan pada konteks operasi dan proses dalam skenario *real-time*. Di otomasi rumah setiap kotak saklar listrik akan terhubung dengan ponsel kakitar (atau kadang-kadang *remote*) sehingga itu bisa dioperasikan dari jarak jauh. Tapi skenario seperti itu tidak perlu prosesor dan perangkat penyimpanan dipasang di setiap kotak saklar. Hanya dibutuhkan sensor untuk menangkap sinyal dan proses itu (kebanyakan beralih *ON / OFF*). Jadi arsitektur sistem ini bervariasi tergantung pada konteks penerapannya”.



Gambar 2. 8 Topologi Internet Of Things

2.5.1 Teknologi Internet of Things (IoT)

Internet of Things menggunakan beberapa teknologi yang secara garis besar digabungkan menjadi satu kesatuan diantaranya sensor sebagai pembaca data, koneksi internet dengan beberapa macam topologi jaringan, *radio frequency*

identification (RFID), *wireless sensor network* dan teknologi yang terus akan bertambah sesuai dengan kebutuhan. [7]

2.5.2 Prinsip Kerja *Internet of Things* (IoT)

Dasar prinsip kerja perangkat *IoT* adalah, benda di dunia nyata diberikan identitas unik dan dapat dikenali di sistem komputer dan dapat di representasikan dalam bentuk data di sebuah sistem komputer. Pada awal-awal gagasan *IoT* pengenalan yang digunakan agar benda dapat diidentifikasi dan dibaca oleh komputer adalah dengan menggunakan kode batang (*Barcode*), Kode QR (*QR Code*) dan Identifikasi Frekuensi Radio (*RFID*). dalam perkembangannya sebuah benda dapat diberi pengenalan berupa IP address dan menggunakan jaringan internet untuk bisa berkomunikasi dengan benda lain yang memiliki pengenalan *IP address*.

Cara Kerja *Internet of Things* yaitu dengan memanfaatkan sebuah argumentasi pemrograman yang dimana tiap-tiap perintah argumennya itu menghasilkan sebuah interaksi antara sesama mesin yang terhubung secara otomatis tanpa campur tangan manusia dan dalam jarak berapapun. Internetlah yang menjadi penghubung di antara kedua interaksi mesin tersebut, sementara manusia hanya bertugas sebagai pengatur dan pengawas bekerjanya alat tersebut secara langsung. [8]

2.5.3 Perangkat *IoT* (Embedded Device)

Barang apapun dapat dikatakan sebagai *IoT Device* jika telah terpasang *IoT module/embedded device*, *IoT Module* pada umumnya terdiri dari 4 komponen penting di antaranya :

1. Sensor

Sensor berfungsi sebagai penerima/pengoleksi informasi tentang apa yang ingin dimonitor, misalnya sensor suhu untuk mendapatkan informasi suhu, kamera, microphone, dll.

2. CPU/Komputer

Komputer di jaman ini tidak harus berbentuk laptop. Khusus untuk membuat perangkat *IoT* ada perangkat komputer kecil atau sering disebut *single board computer* seperti *raspberry pi* atau *arduino*. Perangkat komputer kecil inilah yang diprogram untuk mengolah informasi dari sensor yang terpasang dan menentukan tindakan berdasarkan informasi yang diperoleh dari sensor.

CPU juga bertugas sebagai pengolah data yang nantinya akan dikirim ke perangkat lain untuk diolah.

3. Sistem Operasi

Embedded Device untuk perangkat *IoT* memerlukan sistem operasi khusus karena perangkat *IoT* berukuran kecil/*portable* dan memiliki spesifikasi yang minim. Sistem operasi inilah yang menjadi nyawa dari perangkat/*module/embedded device/perangkat IoT/Module IoT*

4. Jalur Komunikasi

Setelah sensor mengoleksi informasi dan CPU mengolah dan menentukan tindakan berdasarkan informasi yang diterima maka perangkat *IoT* memerlukan jalur komunikasi untuk mengirim data yang telah diolah nya ke *user* atau bahkan ke *server* pusat. media komunikasi disini bisa berupa *bluetooth*, *wifi*, dan untuk mengirim informasi dari tempat yang jauh tanpa batasan rung dan waktu maka perangkat *IoT* akan menggunakan media internet.

5. Keluaran

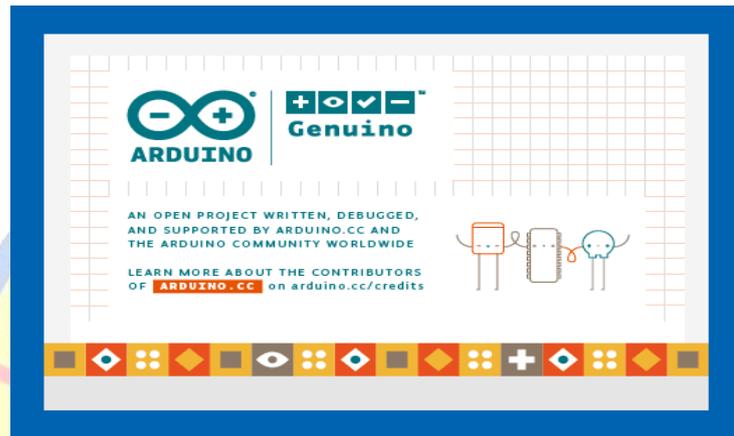
Keluaran disini merupakan action dari program yang terpasang di CPU seperti mengirim informasi ke pusat server jika memenuhi kondisi tertentu.

2.6 Arduino IDE

Arduino IDE merupakan kependekan dari *Integrated Development Enviroenment*, atau secara bahasa mudahnya merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Disebut sebagai lingkungan karena melalui *Software* inilah Arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang dibenamkan melalui sintaks pemrograman. Arduino menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C.

Bahasa pemrograman Arduino (*Sketch*) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Sebelum dijual ke pasaran, IC mikroendalier Arduino telah ditanamkan suatu program bernama *Bootlader* yang berfungsi sebagai penengah antara *compiler* Arduino dengan mikroendalier. Arduino IDE dibuat dari bahasa pemrograman JAVA. Arduino IDE juga dilengkapi dengan *library C/C++* yang biasa disebut *wiring* yang membuat operasi *input* dan *output* menjadi lebih mudah. Arduino IDE ini dikembangkan dari *Software processing* yang dirombak menjadi

Arduino IDE khusus untuk pemrograman dengan Arduino. Program yang ditulis dengan menggunakan *Arduino Software (IDE)* disebut sebagai *sketch*. *Sketch* ditulis dalam suatu editor teks dan disimpan dalam file dengan ekstensi. Teks editor pada *Arduino Software* memiliki fitur seperti *cutting/paste* dan *seraching/replacing* sehingga mempermudah dalam menulis kode program.[9]



Gambar 2. 9 Software Arduino IDE

2.7 *Blynk*

Blynk merupakan aplikasi berbasis layanan yang dapat digunakan sebagai pengontrol mikrokontroler berbasis internet. Dalam aplikasi *Blynk* terdapat beberapa widget yang harus disusun sesuai dengan kebutuhan. *Blynk* menjadi interface antara mikrokontroler dan *smartphone* melalui koneksi internet. Aplikasi ini tidak berbayar jika widget yang digunakan sedikit. Untuk penggunaan aplikasi yang membutuhkan widget banyak maka ada beberapa widget yang berbayar. *Blynk* dapat merekam dan menyimpan data sesuai dengan pengaturan yang telah ditanamkan. *Blynk* merupakan platform baru yang memudahkan peneliti untuk menghubungkan perangkat keras dengan tampilan pada *smartphone*. Waktu yang dibutuhkan untuk memprogram pada aplikasi *Blynk* relatif cepat [10].

BAB III METODE PENELITIAN

3.1 Gambaran Umum

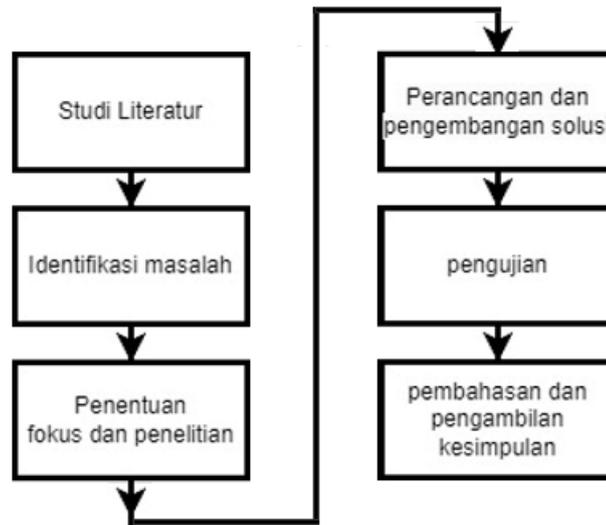
Rancangan prototipe ini terdiri dari 2 komponen utama, yakni *hardware* (perangkat keras) dan *Software* (perangkat lunak). Pembahasan yang digunakan dalam perangkat keras terdiri dari komponen fisik seperti modul ESP32, kabel penghubung, modul relai 10 *channel*, lampu LED, *smartphone*, dan adaptor untuk modul ESP32. Sedangkan untuk perangkat lunaknya sendiri menggunakan *Software* dari aplikasi *Blynk smartphone*

Sistem kendali penerangan jarak jauh ini menggunakan ESP32 sebagai pusat kendali yang menghubungkan berbagai perangkat keras seperti modul relai, dan aplikasi pada *smartphone* untuk mengatur lampu secara *mobile*. Dengan modul relai, sistem ini dapat mengontrol beberapa lampu secara terpisah. Sedangkan aplikasi pada *smartphone* berfungsi sebagai platform kendali yang memungkinkan pengguna memantau dan mengelola penerangan dari jarak jauh melalui aplikasi berbasis IoT.

3.2 Tahapan Penelitian

Pendekatan metode penelitian yang dilakukan oleh penulis adalah pendekatan Studi Literatur, identifikasi masalah, penentuan fokus dan penelitian, perancangan dan pengembangan solusi, pengujian, pembahasan dan pengambilan kesimpulan. Bagan alir dari tahapan penelitian dapat dilihat pada gambar.

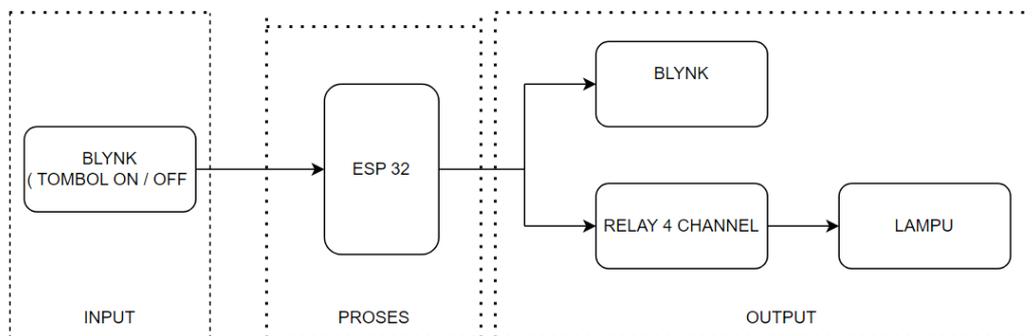
Dari awal mulai penerapan pengontrolan lampu di lapangan. Yaitu *connect Blynk* dengan internet atau Wifi agar pengontrolan bisa berjalan dengan optimal, kemudian setelah terkoneksi dengan jaringan internet maupun Wifi barulah dilakukan pengontrolan penerangan yaitu perintah antara lampu tersebut dalam kondisi menyala ataupun mati yang selanjutnya dikirim perintah kepada ESP 32 untuk mengirim perintah Kembali kepada relai sebagai saklar elektrik baik dalam kondisi mati ataupun hidup tergantung kondisi cuaca yang ada di lapangan secara gambar di atas tertera kondisi *on* maupun *off*. Setelah itu nantinya di *display Blynk* akan menampilkan status apakah penerangan sudah menyala ataupun tidak dan dilanjutkan dengan beban lampu tersebut dalam kondisi nyala / mati.



Gambar 3. 1 Tahapan Penelitian

3.3. Blok Diagram

Pada blok diagram perancangan sistem penerangan berbasis IoT ini menggambarkan hubungan antar komponen utama dalam rangkaian sistem. Sistem ini bekerja secara masif dan optimal untuk meningkatkan efisiensi penerangan di Bandar Udara Internasional Soekarno - Hatta. Sistem ini menggunakan modul ESP32 sebagai inti pengolahan data dan komunikasi, serta aplikasi *Blynk* pada *smartphone* untuk kendali jarak jauh. Gambar di bawah ini menunjukkan detail blok diagram tersebut:



Gambar 3. 2 Diagram Blok Sistem

1. Input

Rancangan rangkaian ESP32 dengan modul relai 10 *channel*. Dimulai dari adanya simbol *Blynk* yang merupakan akses aplikasi pada *smartphone* untuk

sebuah pengontrolan utama di antarmuka aplikasi tersebut. Di dalam tampilan aplikasi *Blynk* tersebut terdapat beberapa akses untuk mengontrol penerangan yang ada di lapangan dan memanfaatkan jaringan internet dan memberi perintah kepada ESP 32. Aplikasi *Blynk* tersebut memiliki di antaranya tombol *on* maupun *off* untuk pengontrolan,

2. Proses

Dari proses *input* yang ada dilanjutkan proses yang merupakan data perintah dari aplikasi *Blynk* tersebut diterima oleh ESP32 yang akan mengolah data kepada relai 10 channel dan mengaktifkan maupun menonaktifkan penerangan di lapangan serta mengirim data dan sinyal Kembali ke aplikasi *Blynk* tersebut.

3. Output

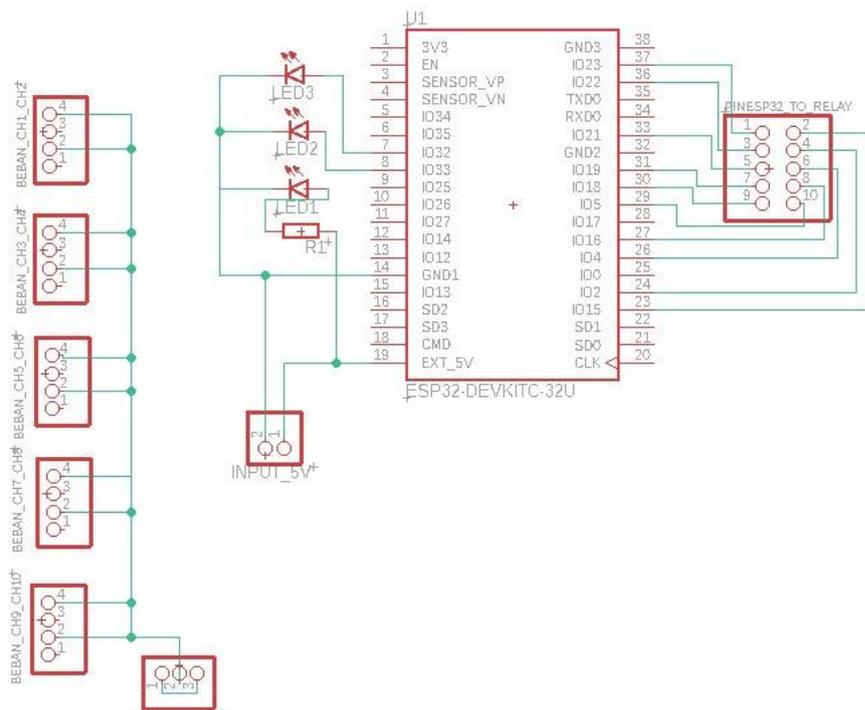
Yang terakhir yaitu merupakan *output* dari semua yang sudah dilewati dari input hingga proses. Dari proses *output* ini nantinya aplikasi *Blynk* menerima status jika sudah kondisi di lapangan untuk penerangannya *off* maupun *on*, dan relai nantinya sebagai saklar otomatis yang akan mengirim perintah kepada beban yaitu lampu. Dan lampu yang dikendali ada beberapa lampu nantinya yang akan dikendali karena luasnya wilayah di terminal 2 Bandar Udara Internasional Soekarno – Hatta.

3.4 Diagram Skematik dan Model 3D Sistem

Pada bagian ini, akan ditampilkan **diagram skematik 2D** dan **model 3D** dari sistem pengendalian lampu berbasis IoT. Diagram ini bertujuan untuk memvisualisasikan hubungan antara komponen-komponen utama sistem serta bagaimana perangkat keras disusun secara fisik.

• Skematik 2D Sistem

Berikut adalah diagram rangkaian listrik yang menggambarkan hubungan antara mikrokontroler, modul relai, dan lampu yang dikendalikan. Diagram ini menunjukkan bagaimana **ESP8266/ESP32** mengendalikan **10-channel relai** untuk mengatur **nyala/mati** lampu.

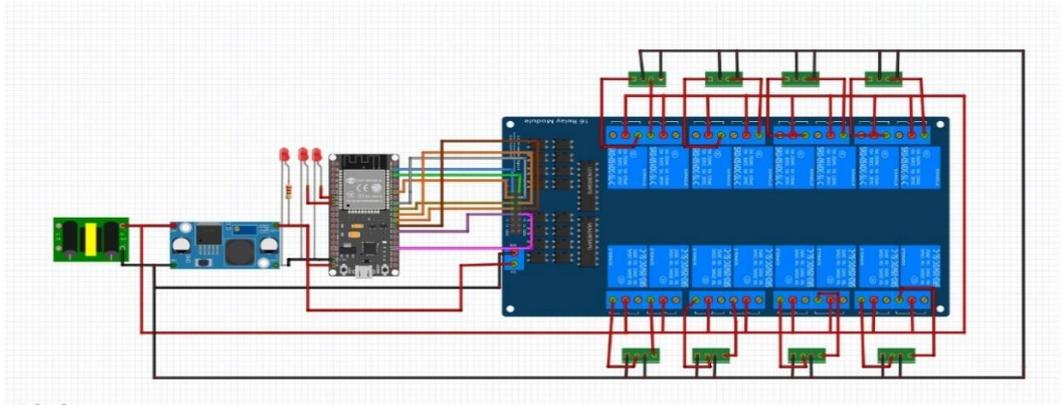


Gambar 3. 3 Skematik Rangkaian 2D

Keterangan: Pada gambar ini, setiap relai dikendalikan oleh pin I/O mikrokontroler, dan tiap relai mengendalikan lampu yang terhubung ke sumber daya 12V DC.

- **Model 3D Sistem**

Berikut adalah model 3D yang menunjukkan susunan fisik dari **mikrokontroler, relai, dan lampu** dalam sistem. Model ini memperlihatkan bagaimana komponen-komponen tersebut diintegrasikan dalam **box panel relai** dan bagaimana **ESP8266/ESP32** terhubung dengan relai dan aplikasi *Blynk*.



Gambar 3. 4 Skematik Rangkaian 2D

Keterangan: Pada gambar ini, model 3D memperlihatkan posisi fisik mikrokontroler dan relai dalam kotak, serta pengkabelan yang menghubungkan relai dengan lampu.

3.5 Perancangan Hardware

Sistem ini menggunakan 34 buah lampu yang dikendalikan oleh ESP8266/ESP32 dan *Blynk* Cloud melalui modul relai 10-channel. Perangkat keras utama terdiri dari:

- Mikrokontroler: ESP8266 / ESP32
- Modul Relai: 10-channel relai untuk mengontrol nyala/mati lampu
- Catu Daya: 12V DC untuk lampu LED, 5V DC untuk mikrokontroler

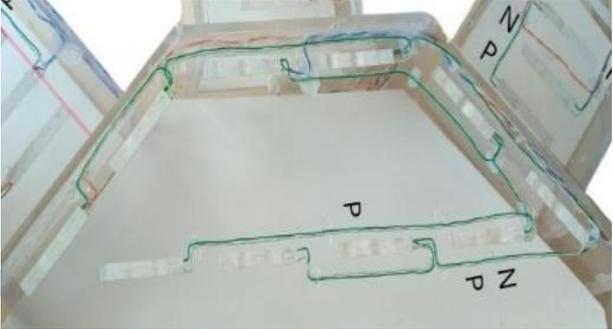
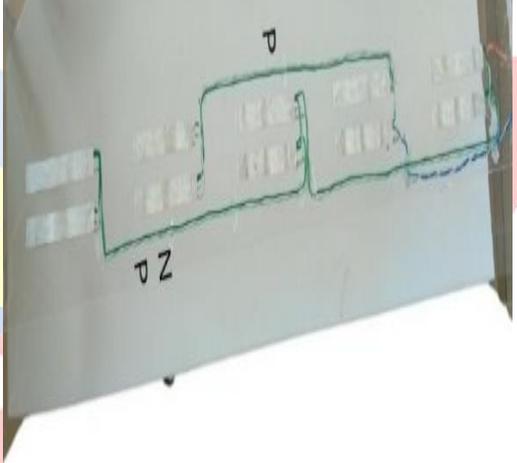
Perancangan dilakukan dengan membuat maket gedung terminal 2 bandara soetta menggunakan potongan akrilik, di mana terdapat 2 lantai bangunan dengan 3 sayap tambahan seperti ditunjukkan pada gambar 3.5



Gambar 3. 5 Maket Gedung Terminal 2 Bandara Soetta Tampak Atas

Maket dibuat menjadi 2 lantai, di mana lantai 1 menggunakan akrilik putih dan lantai 2 menggunakan akrilik transparan. Pada setiap lantai bangunan utama menggunakan 20 lampu dengan skema *Priority* (P) dan *Non-Priority* (NP) dengan 4 wilayah: *Arrival Area* lantai 1 bagian depan, *Pick Up Point* lantai 2 bagian depan, *Check in Counter* lantai 1 bagian belakang dan *Transit Area* lantai 2 bagian belakang. Sedangkan pada sayap bangunan kiri dan kanan atau disebut wilayah *Boarding Area* menggunakan 4 lampu dengan skema yang sama untuk lantai 1 dan 2 (2 P dan 2 NP), dan juga sayap depan (*boarding area*) menggunakan 6 lampu pada sayap depan (3 P dan 3 NP). Berikut merupakan penjelasan pemosisian skema P dan NP pada setiap lantai di maket:

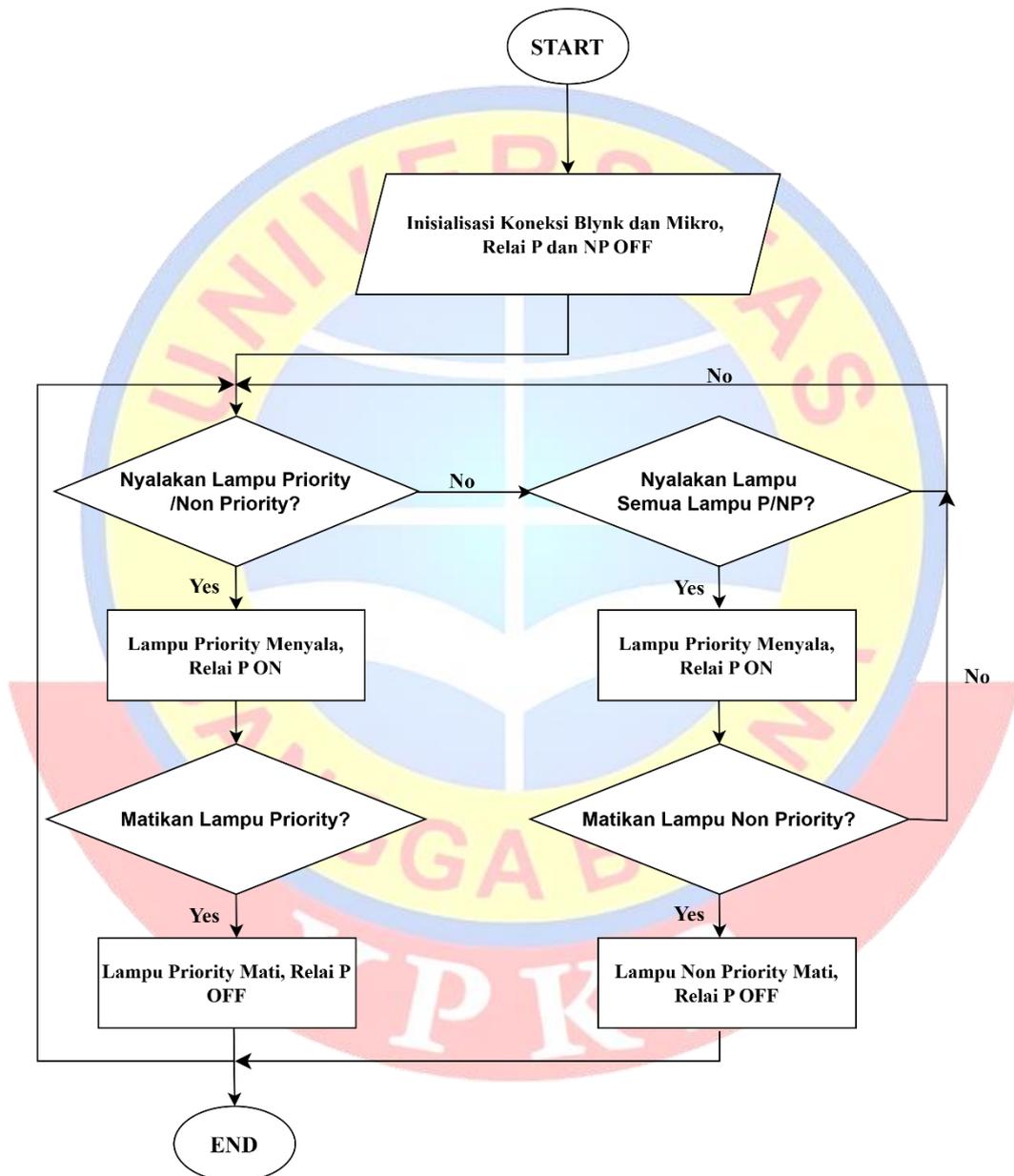
Tabel 3. 1 Pemosisian Lampu P dan NP Pada Maket

No.	Area	Priority (P) dan Non Priority (NP)
1	Arrival Area lantai 1 dan Pick Up Point lantai 2 (bangunan utama bagian depan)	
2	Boarding Area (Sayap depan, kiri dan kanan)	
3	Check In Counter lantai 1 dan Transit Area lantai 2 (gedung utama bagian belakang)	

3.6 Perancangan Software

Sistem menggunakan ESP32 sebagai pengendali utama dan dihubungkan ke aplikasi *Blynk* untuk menampilkan kontrol dan status dari lampu yang dikendalikan. Sistem menggunakan aplikasi *Blynk* pada *smartphone* sebagai pengirim perintah utama dan penampil status *on/off* saat dikendalikan. Pilihan yang muncul pada

kendali pada *Blynk* adalah *on* dan *off* pada lampu untuk setiap lampu yang terhubung ke relai. Apabila tombol *on* (baik lampu P atau NP) ditekan maka lampu akan menyala sesuai skema, dan bila tombol *off* ditekan, maka lampu (baik P atau NP) akan mati. Berikut merupakan *flowchart* sistem yang digunakan seperti ditunjukkan pada gambar 3.6



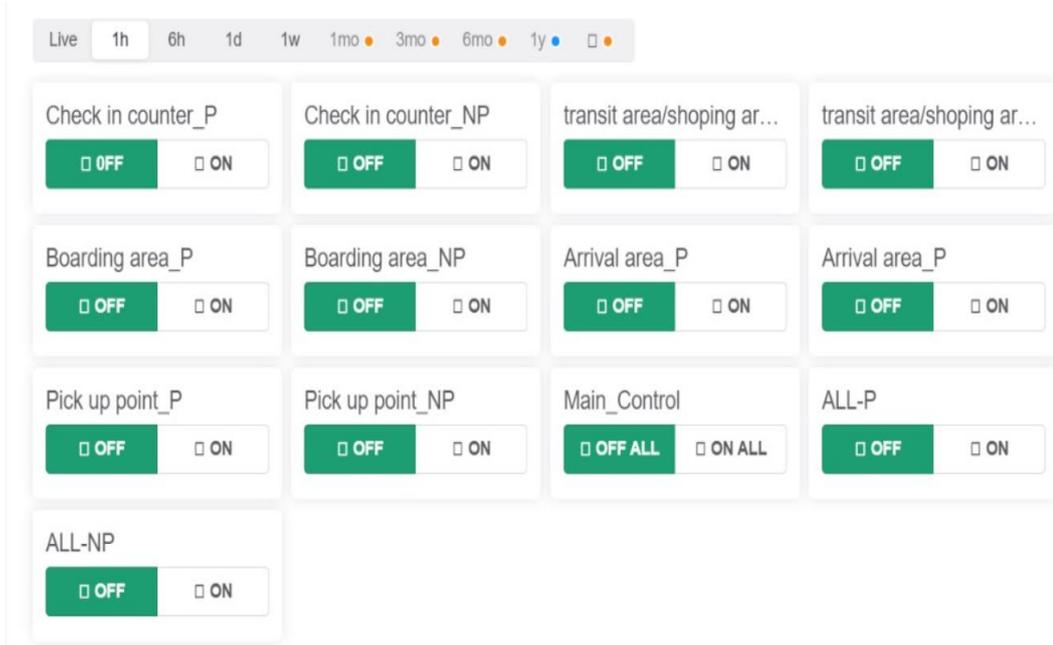
Gambar 3. 6 Flowchart Sistem

Tampilan *Blynk* dibuat pada *smartphone* dan PC menyesuaikan wilayah maket pada tabel 3.1, di mana skema P dan NP akan terbagi pada semua wilayah.

Selain itu dibuat juga kontrol utama pada untuk menyalakan/mematikan semua lampu sekaligus, baik P, NP ataupun seluruh lampu sekaligus, seperti ditunjukkan pada gambar 3.7 dan 3.8.



Gambar 3. 7 Tampilan *Blynk* di *Smartphone*



Gambar 3. 8 Tampilan *Blynk* di PC



BAB IV HASIL DAN PEMBAHASAN

Pengujian dibagi menjadi beberapa tahap, pertama pengujian konektivitas *Blynk* ke mikrokontroler, dilanjutkan dengan pengujian nyala lampu dari *Blynk* ke lampu disertai dengan pengujian respon waktu nyala/mati lampu. Terakhir dilakukan pengujian tegangan keluaran relai.

4.1 Pengujian Aplikasi *Blynk*

Pengujian ini dilakukan untuk memastikan bahwa aplikasi *Blynk* dapat berfungsi dengan baik dalam mengendalikan lampu melalui internet. Aplikasi ini berperan sebagai antarmuka utama bagi pengguna dalam menyalakan atau mematikan lampu secara *real-time*. Dengan adanya aplikasi ini, pengguna dapat dengan mudah mengontrol delapan buah lampu tanpa harus berada di lokasi yang sama dengan perangkat keras yang dikendalikan.

Aplikasi *Blynk* dipilih karena kemudahannya dalam menghubungkan perangkat IoT dengan mikrokontroler melalui jaringan internet. Pengguna hanya perlu meng-install aplikasi di perangkat *smartphone* dan melakukan konfigurasi yang sesuai agar dapat mengontrol sistem dengan lancar. Oleh karena itu, pengujian aplikasi ini menjadi sangat penting untuk memastikan bahwa sistem bekerja dengan baik dan tidak mengalami kendala dalam operasionalnya.

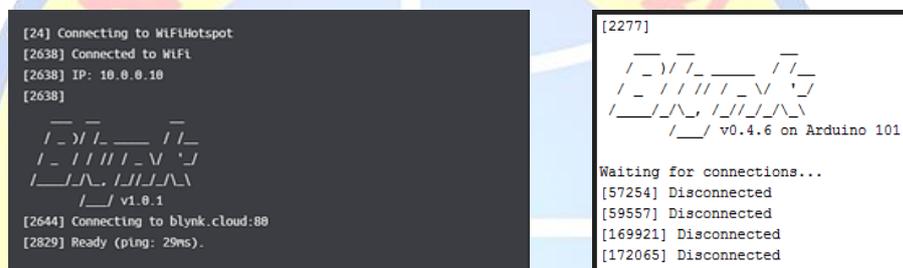
Pengujian aplikasi *Blynk* dilakukan dengan fokus pada beberapa aspek utama, yaitu konektivitas aplikasi ke server, kecepatan respon terhadap perintah ON/OFF, serta tampilan antarmuka yang digunakan untuk mengontrol lampu. Pengujian ini juga mempertimbangkan faktor-faktor yang dapat mempengaruhi performa aplikasi, seperti kualitas jaringan internet dan stabilitas sistem selama pengoperasian dalam jangka waktu tertentu.

4.1.1 Konektivitas Aplikasi *Blynk*

Salah satu faktor utama dalam sistem ini adalah koneksi antara aplikasi *Blynk* dengan mikrokontroler yang bertugas mengendalikan relai untuk menyalakan atau mematikan lampu. Jika koneksi tidak stabil, maka

sistem tidak akan bekerja secara optimal, dan pengguna mungkin mengalami keterlambatan atau kegagalan dalam mengontrol lampu.

Untuk memastikan konektivitas berjalan dengan baik, dilakukan beberapa tahap pengujian. Pertama, aplikasi *Blynk* dijalankan dan diperiksa apakah dapat terhubung ke server *Blynk*. Jika aplikasi berhasil terhubung, maka status yang ditampilkan pada dashboard akan menunjukkan "**Connected**", yang berarti aplikasi siap digunakan. Namun, jika koneksi tidak stabil atau ada kendala jaringan, maka status akan berubah menjadi "**Disconnected**", yang menunjukkan bahwa aplikasi tidak dapat berkomunikasi dengan mikrokontroler.



```
[24] Connecting to WiFiHotspot
[2638] Connected to WiFi
[2638] IP: 10.0.0.10
[2638]
  _ _ _ _ _
 / _ \ / _ \ / _ \
| _ | | | | | _ \ 'J
| _ \ / _ \ / _ \ / _ \
  _ _ _ _ _
  / _ \ v1.0.1
[2644] Connecting to blynk.cloud:80
[2829] Ready (ping: 29ms).
```

```
[2277]
  _ _ _ _ _
 / _ \ / _ \ / _ \ / _ \
| _ | | | | | _ \ 'J
| _ \ / _ \ / _ \ / _ \
  _ _ _ _ _
  / _ \ v0.4.6 on Arduino 101

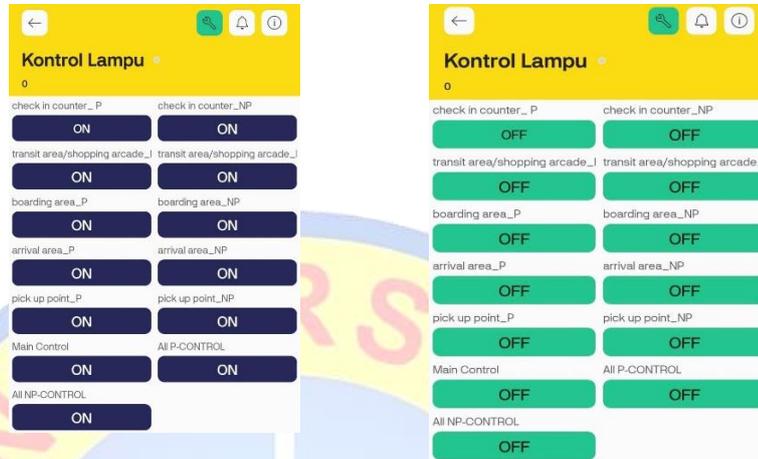
Waiting for connections...
[57254] Disconnected
[59557] Disconnected
[169921] Disconnected
[172065] Disconnected
```

Gambar 4. 1 Tampilan Connected dan Disconnected to *Blynk*

4.1.2 Respon Waktu Aplikasi *Blynk*

Selanjutnya, dilakukan pengujian dengan menekan tombol ON/OFF pada aplikasi untuk melihat apakah perintah dapat dikirimkan ke mikrokontroler dan dieksekusi dengan baik dan mengukur waktu nyala/mati menggunakan stopwatch. Jika perintah berhasil dikirim dan lampu merespons sesuai dengan yang diharapkan, maka dapat disimpulkan bahwa konektivitas antara aplikasi dan perangkat keras berjalan dengan baik. Namun, jika terjadi keterlambatan atau kegagalan dalam mengeksekusi perintah, maka kemungkinan terdapat masalah pada koneksi jaringan atau server *Blynk* itu sendiri. Selain itu, pengujian dilakukan dalam berbagai kondisi jaringan untuk melihat bagaimana aplikasi berfungsi dalam situasi yang berbeda. Pada koneksi internet yang stabil, aplikasi dapat bekerja dengan lancar tanpa kendala, dengan waktu respon yang sangat cepat. Namun, pada jaringan yang lebih lambat atau tidak stabil, kadang terjadi

keterlambatan dalam pengiriman perintah, yang menyebabkan lampu tidak segera menyala atau mati sesuai dengan instruksi dari aplikasi.



Gambar 4.2 Pengujian ON/OFF Aplikasi *Blynk*

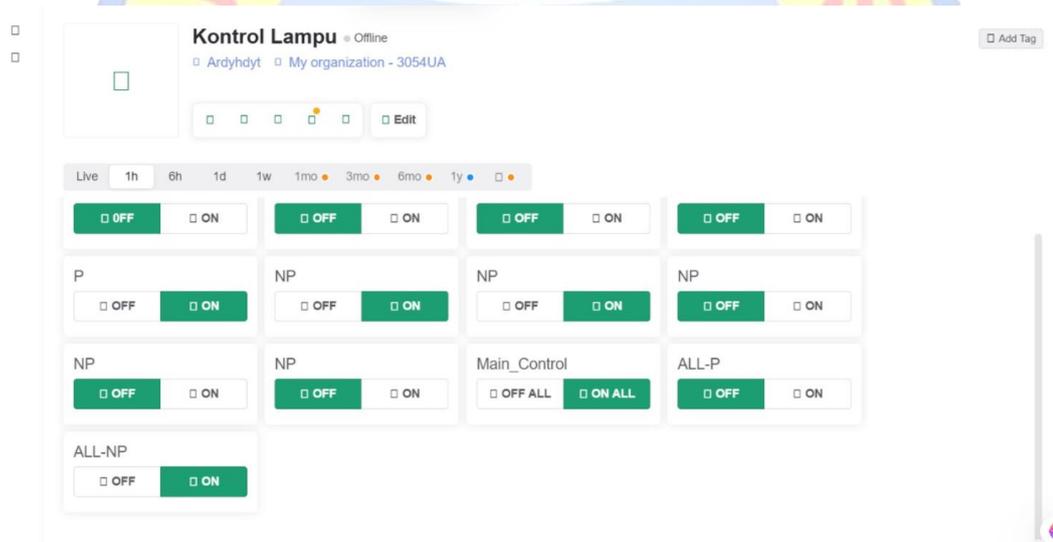
Tabel 4. 1 Respon Waktu Nyala/Mati Dari *Blynk* Ke Lampu

No.	Perintah	Interface <i>Blynk</i> (sec)
1	ALL Lamp on	0,652
2	All lamp off	0,6213
3	ALL on P	0,54
4	ALL off P	0,534
5	ALL on NP	0,532
6	ALL off NP	0,521
7	Manual on Boarding Area	0,5
8	Manual off Boarding Area	0,498
9	Manual on Arrival Area	0,51
10	Manual off Arrival Area	0,5
11	Manual on Check in Counter	0,5
12	Manual off Check in Counter	0,496
13	Manual on Transit Area	0,494
14	Manual off Transit Area	0,502
15	Manual on Pick Up Point	0,498
16	Manual off Pick Up Point	0,5

Dari pengujian ini, dapat disimpulkan bahwa aplikasi *Blynk* dapat terhubung dengan baik ke mikrokontroler selama jaringan internet dalam kondisi stabil. Namun, jika jaringan mengalami gangguan atau latensi tinggi, terdapat kemungkinan bahwa perintah yang dikirimkan mengalami keterlambatan atau gagal terkirim. Oleh karena itu, dalam implementasi sistem ini, penting untuk memastikan bahwa jaringan internet yang digunakan memiliki kecepatan dan stabilitas yang cukup baik agar aplikasi dapat berfungsi dengan optimal.

4.1.3 Tampilan Antarmuka Aplikasi *Blynk*

Aplikasi *Blynk* yang digunakan dalam sistem ini memiliki antarmuka sederhana dan mudah digunakan. Berikut tampilan antarmuka aplikasi yang digunakan:



Gambar 4. 2 Tampilan *Blynk* Dekstop Mengikuti Kontrol dari *Smartphone*

Pada tampilan aplikasi *Blynk*, terdapat beberapa fitur utama:

- **Tombol ON/OFF untuk setiap lampu:** Memungkinkan pengguna untuk menyalakan dan mematikan lampu secara individual.
- **Indikator status lampu:** Menampilkan kondisi lampu (menyala atau mati) secara real-time.
- **Status koneksi:** Menunjukkan apakah aplikasi *Blynk* berhasil terhubung ke mikrokontroler.

Hasil pengujian menunjukkan bahwa seluruh tombol dan indikator bekerja dengan baik, sehingga aplikasi dapat digunakan untuk mengendalikan lampu secara efisien.

4.2 Pengujian Relai dan Lampu

Setiap lampu terhubung ke relai yang dikendalikan oleh ESP8266/ESP32. Relai akan menyalurkan tegangan 12V ke lampu LED saat mendapatkan sinyal dari aplikasi *Blynk*. Berikut merupakan pengujian lampu P dan NP untuk semua area

Tabel 4. 2 Pengujian kontrol lampu dari masing – masing wilayah

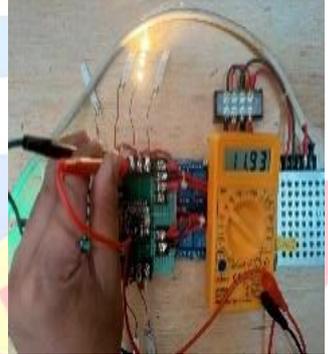
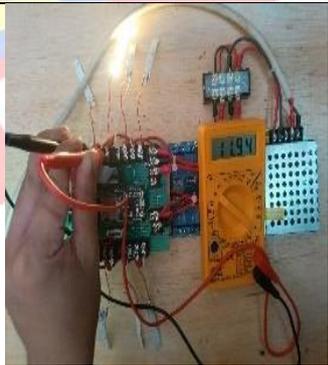
No.	Area	Priority (P)	Non Priority (NP)
1	Arrival Area (bangunan utama bagian depan lantai 1)		
2	Boarding Area lantai 1 dan 2 (Sayap depan, kiri dan kanan)		

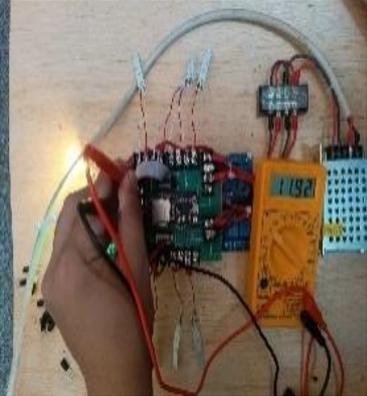
No.	Area	Priority (P)	Non Priority (NP)
3	Check In Counter (gedung utama bagian belakang lantai 2)		
4	Transit Area (Gedung utama bagian depan lantai 2)		
5	Pick Up Point (Gedung utama bagian belakang, lantai 1)		

4.1.4 Pengukuran Tegangan pada Setiap Lampu

Untuk memastikan bahwa sistem berfungsi dengan baik, dilakukan pengukuran tegangan pada setiap lampu saat relai diaktifkan. Pengujian dilakukan dengan memasang lampu LED pada *output* relai dan mengukur tegangan keluarannya. Berikut hasil pengukurannya:

Tabel 4. 3 Data Pengukuran Tegangan pada LED

No.	Lampu	Tegangan Pada LED (V)	Hasil Pengujian
1	Lampu 1 P Arrival Area (bangunan utama bagian depan lantai 1)	11.93 V	
2	Lampu 2 P Boarding Area lantai 1 dan 2 (Sayap depan, kiri dan kanan)	11.93 V	
3	Lampu 3 P Check In Counter (gedung utama bagian belakang lantai 2)	11.94 V	

No.	Lampu	Tegangan Pada LED (V)	Hasil Pengujian
4	Lampu 4 P Transit Area (Gedung utama bagian depan lantai 2)	11.93 V	
5	Lampu 5 P Pick Up Point (Gedung utama bagian belakang, lantai 1)	11.92 V	
6	Lampu 6 NP Check In Counter (gedung utama bagian belakang lantai 2)	11.93 V	

No.	Lampu	Tegangan Pada LED (V)	Hasil Pengujian
7	Lampu 7 NP Transit Area (Gedung utama bagian depan lantai 2)	11.93 V	
8	Lampu 8 NP Pick Up Point (Gedung utama bagian belakang, lantai 1)	11.94 V	

Hasil ini menunjukkan bahwa relai berfungsi normal, dengan tegangan input 5V DC yang sesuai dengan kebutuhan ESP8266/ESP32, dan tegangan output DC stabil di sekitar 11.93V

BAB V

KESIMPULAN

5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan mengenai pengendalian lampu berbasis IoT menggunakan aplikasi *Blynk*, beberapa kesimpulan dapat diambil sebagai berikut:

1. Sistem pengendalian lampu berbasis IoT berhasil diimplementasikan dengan menggunakan mikrokontroler dan aplikasi *Blynk* sebagai antarmuka pengguna. Hasil pengujian menunjukkan bahwa sistem dapat berfungsi dengan baik dalam menyalakan dan mematikan lampu secara real-time melalui jaringan internet.
2. Konektivitas antara aplikasi *Blynk* dan mikrokontroler sangat bergantung pada kualitas jaringan internet. Dalam kondisi jaringan stabil, sistem memiliki waktu respon yang cepat, yaitu berkisar antara 0,494 hingga 0,652 detik. Namun, pada jaringan yang kurang stabil respon waktu dapat bertambah lebih lama.

5.2 Saran

Untuk pengembangan dan peningkatan sistem ini di masa mendatang, beberapa saran yang dapat diberikan adalah sebagai berikut:

1. Peningkatan Stabilitas Koneksi Internet, Mengingat performa sistem sangat bergantung pada kualitas jaringan, maka disarankan untuk menggunakan jaringan internet yang lebih stabil, seperti WiFi berkecepatan tinggi atau koneksi berbasis jaringan seluler yang lebih andal.
2. Integrasi dengan Sensor Tambahan, Sistem dapat dikembangkan lebih lanjut dengan menambahkan sensor cahaya atau sensor gerak sehingga lampu dapat menyala dan mati secara otomatis berdasarkan kondisi lingkungan, tanpa perlu dikontrol secara manual.
3. Pengembangan Fitur pada Aplikasi *Blynk*, Menambahkan fitur seperti jadwal otomatis (timer) untuk menyalakan atau mematikan lampu pada waktu tertentu guna meningkatkan efisiensi penggunaan energi.
4. Penggunaan Mikrokontroler dengan Kapasitas Lebih Tinggi, Untuk meningkatkan kinerja sistem, dapat dipertimbangkan penggunaan mikrokontroler dengan spesifikasi lebih tinggi yang memiliki konektivitas lebih stabil, seperti ESP32 yang mendukung dual-band WiFi dan Bluetooth.

5. Pengujian pada Skala Lebih Besar, Pengujian lebih lanjut perlu dilakukan pada jumlah lampu yang lebih banyak dan dalam berbagai kondisi jaringan untuk memastikan sistem tetap berfungsi dengan baik dalam skala yang lebih luas.

Dengan adanya pengembangan lebih lanjut berdasarkan saran di atas, diharapkan sistem pengendalian lampu berbasis IoT ini dapat menjadi solusi yang lebih optimal, efisien, dan dapat diterapkan dalam berbagai skenario manajemen pencahayaan di lingkungan bandara atau fasilitas umum lainnya.

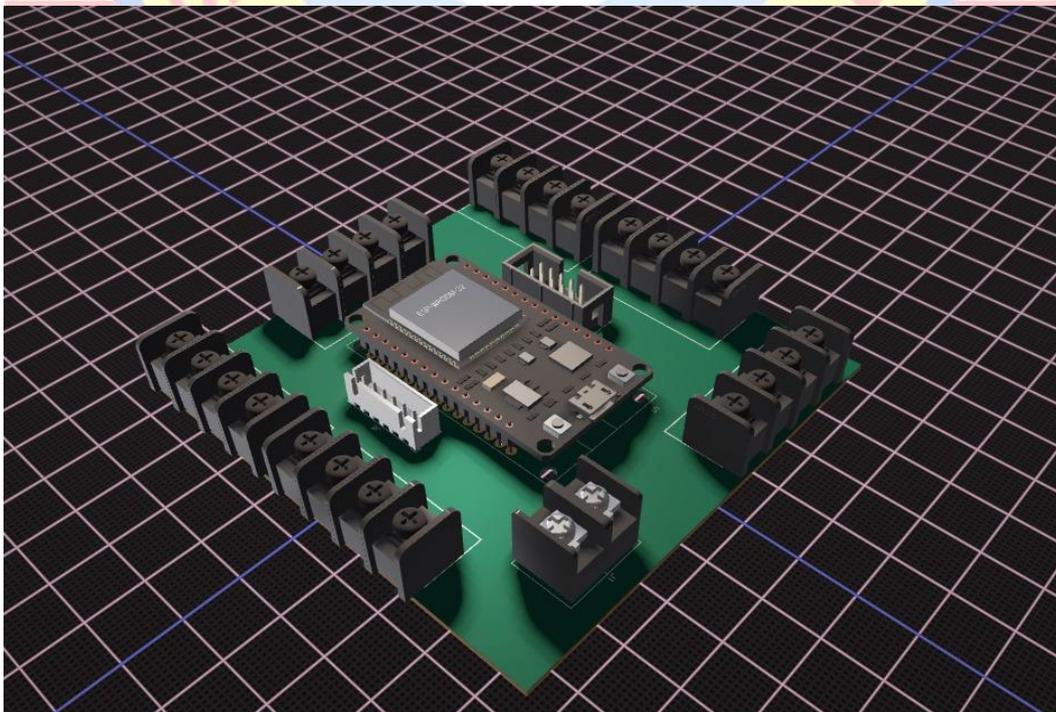
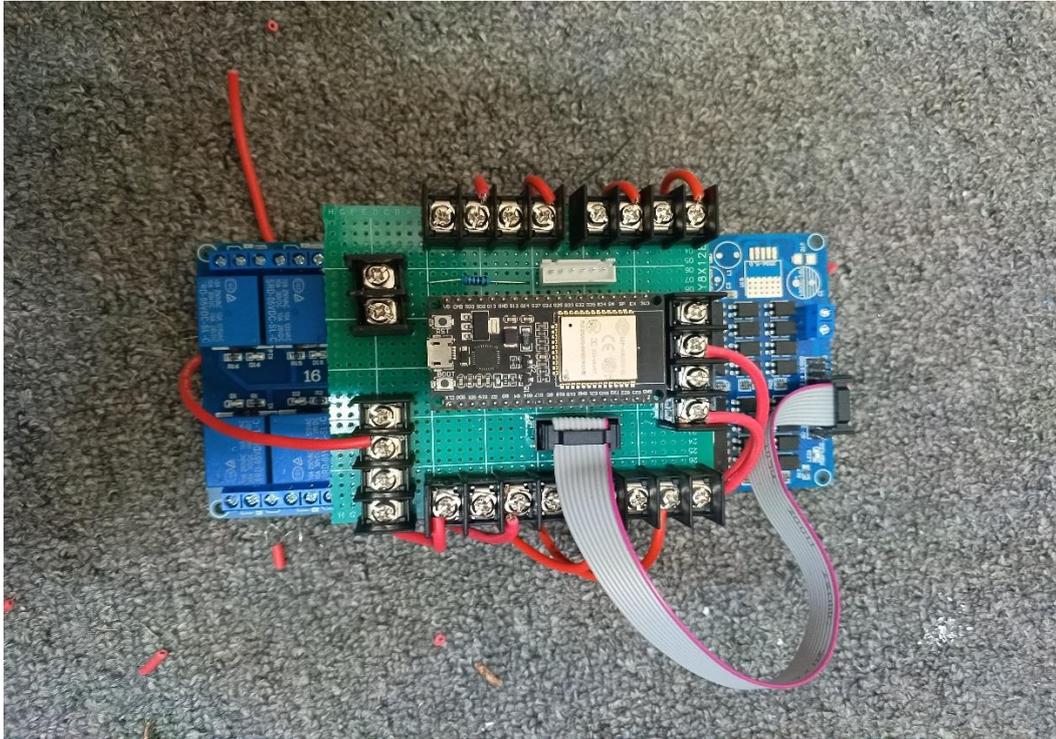


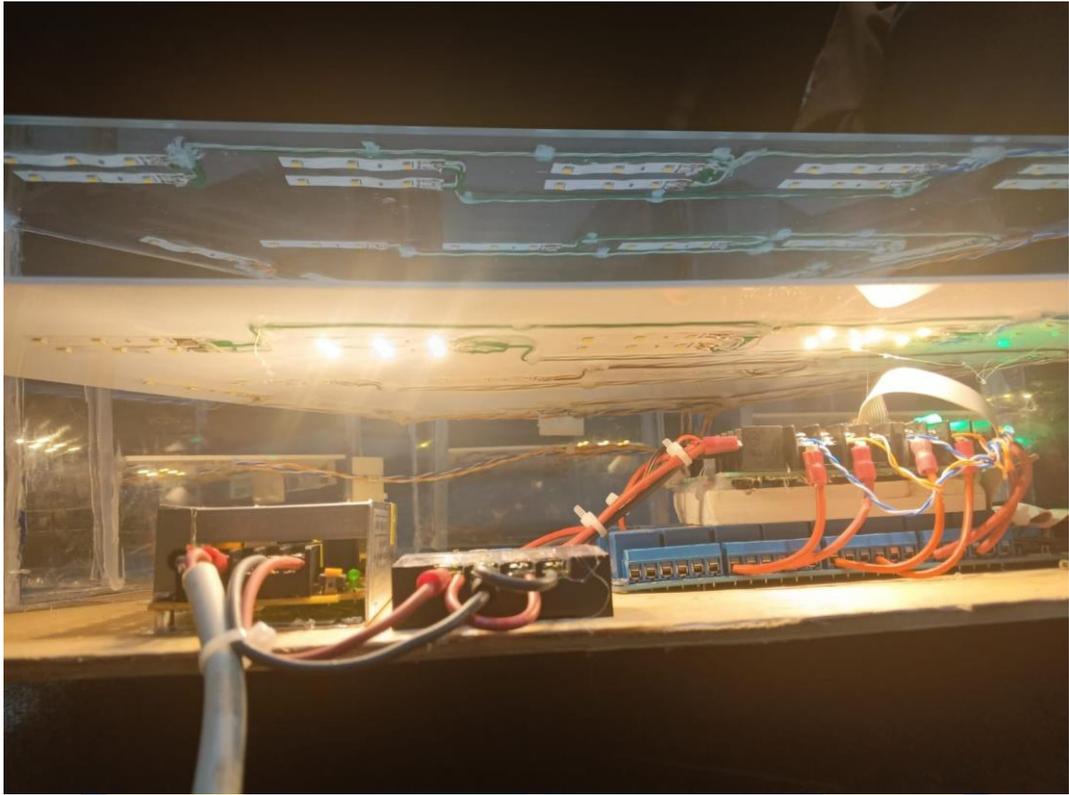
DAFTAR PUSTAKA

- [1] F. Farid and N. S. Salahuddin, "Sistem Pantau dan Kendali Peralatan Listrik Rumah Berbasis Internet of Things (IoT)," *G-Tech: Jurnal Teknologi Terapan*, vol. 8, no. 2, pp. 1079–1090, Apr. 2024, doi: 10.33379/gtech.v8i2.4089.
- [2] A. Prafanto, E. Budiman, P. P. Widagdo, G. Mahendra Putra, R. Wardhana, and U. Mulawarman, "PENDETEKSI KEHADIRAN MENGGUNAKAN ESP32 UNTUK SISTEM PENGUNCI PINTU OTOMATIS," *Jurnal Teknologi Terapan* |, vol. 7, no. 1, 2021.
- [3] D. Susilo, C. Sari, and G. W. Krisna, "Sistem Kendali Lampu pada Smart Home Berbasis IoT (Internet of Things)," *Jurnal ELECTRA : Electrical Engineering Articles*, vol. 2, no. 1, pp. 23–30, 2021.
- [4] M. Saleh and M. Haryanti, "RANCANG BANGUN SISTEM KEAMANAN RUMAH MENGGUNAKAN RELAY," 2017.
- [5] R. Muzawi, W. Joni Kurniawan, J. K. Purwodadi Indah, S. Barat, T.-P. Jln Jend Ahmad Yani No, and K. Baru, "Penerapan Internet of Things (IoT) Pada Sistem Kendali Lampu Berbasis Mobile," 2018. [Online]. Available: <http://tunasbangsa.ac.id/ejurnal/index.php/jsakti>
- [6] R. Apriza Dini, "Internet of Thing Sistem Pengendalian Lampu Jarak Jauh Menggunakan Nodemcu Amica CP2102 berbasis Mobile," vol. 10, no. 2, 2021, doi:10.21063/JTE.2021.31331014.
- [7] A. Kurniawan, "SEJARAH, CARA KERJA DAN MANFAAT INTERNET OF THINGS."
- [8] A. Wardhana, "ARSITEKTUR DAN STANDARISASI INTERNET OF THINGS (IOT)." [Online]. Available: <https://www.researchgate.net/publication/370605476>
- [9] U. Mahanin Tyas, A. Apri Buckhari, P. Studi Pendidikan Teknologi Informasi, and P. Studi Pendidikan Teknologi dan Kejuruan, " APLIKASI ARDUINO IDE PADA MATA KULIAH SISTEM DIGITAL," 2023.
- [10] N. Imamah and D. Sagara Andika, "PERANCANGAN SISTEM MONITORING DAN PENGENDALIAN LAMPU MENGGUNAKAN SENSOR GERAK DAN SENSOR CAHAYA DILENGKAPI INTERNET OF THINGS (IOT) (Studi Kasus Fakultas Teknologi Informasi Universitas Bale Bandung)."

LAMPIRAN

DOKUMENTASI PENELITIAN





PROGRAM YANG DIGUNAKAN

```
#define BLYNK_TEMPLATE_ID "TMPL6FU8Tfdy"
#define BLYNK_TEMPLATE_NAME "Kontrol Lampu"
#define BLYNK_AUTH_TOKEN "GwujQ9iE87trqT7bTLpSFy4VTSTCZ969"

#include <WiFi.h>
#include <BlynkSimpleEsp32.h>

// Kredensial WiFi
char ssid[] = "TEKIDO";
char pass[] = "1234567890";

// // Definisi pin relay untuk ESP32
// #define RELAY1 23 // GPIO23
// #define RELAY2 15 // GPIO22
// #define RELAY3 22 // GPIO21
// #define RELAY4 4 // GPIO19
// #define RELAY5 21 // GPIO18
// #define RELAY6 16 // GPIO5
// #define RELAY7 19 // GPIO4
// #define RELAY8 17 // GPIO2
// #define RELAY9 18 // GPIO15
// #define RELAY10 5 // GPIO13
// Definisi pin relay untuk ESP32
#define RELAY1 5
#define RELAY2 18
#define RELAY3 17
#define RELAY4 19
#define RELAY5 16
#define RELAY6 21
#define RELAY7 4
```

```

#define RELAY8 22
#define RELAY9 15
#define RELAY10 23

// LED Indikator
#define LED_INDIKATOR 2
#define LED_KUNING 33 // LED Kuning di GPIO33
#define LED_HIJAU 32 // LED Hijau di GPIO32

// Inisialisasi Blynk
BlynkTimer timer;

// Fungsi untuk mengontrol relay via Blynk
BLYNK_WRITE(V0) {
  int pinValue = param.asInt();
  digitalWrite(RELAY1, pinValue ? LOW : HIGH);
}

BLYNK_WRITE(V1) {
  int pinValue = param.asInt();
  digitalWrite(RELAY2, pinValue ? LOW : HIGH);
}

BLYNK_WRITE(V2) {
  int pinValue = param.asInt();
  digitalWrite(RELAY3, pinValue ? LOW : HIGH);
}

BLYNK_WRITE(V3) {
  int pinValue = param.asInt();
  digitalWrite(RELAY4, pinValue ? LOW : HIGH);
}

```

```
BLYNK_WRITE(V4) {  
  int pinValue = param.asInt();  
  digitalWrite(RELAY5, pinValue ? LOW : HIGH);  
}
```

```
BLYNK_WRITE(V5) {  
  int pinValue = param.asInt();  
  digitalWrite(RELAY6, pinValue ? LOW : HIGH);  
}
```

```
BLYNK_WRITE(V6) {  
  int pinValue = param.asInt();  
  digitalWrite(RELAY7, pinValue ? LOW : HIGH);  
}
```

```
BLYNK_WRITE(V7) {  
  int pinValue = param.asInt();  
  digitalWrite(RELAY8, pinValue ? LOW : HIGH);  
}
```

```
BLYNK_WRITE(V8) {  
  int pinValue = param.asInt();  
  digitalWrite(RELAY9, pinValue ? LOW : HIGH);  
}
```

```
BLYNK_WRITE(V9) {  
  int pinValue = param.asInt();  
  digitalWrite(RELAY10, pinValue ? LOW : HIGH);  
}
```

```
BLYNK_WRITE(V10) {  
  int pinValue = param.asInt(); // Membaca nilai dari Virtual Pin V10
```

```

if (pinValue == 1) {
digitalWrite(RELAY1, LOW);
digitalWrite(RELAY2, LOW);
digitalWrite(RELAY3, LOW);
digitalWrite(RELAY4, LOW);
digitalWrite(RELAY5, LOW);
digitalWrite(RELAY6, LOW);
digitalWrite(RELAY7, LOW);
digitalWrite(RELAY8, LOW);
digitalWrite(RELAY9, LOW);
digitalWrite(RELAY10, LOW);
} else {
    digitalWrite(RELAY1, HIGH);
digitalWrite(RELAY2, HIGH);
digitalWrite(RELAY3, HIGH);
digitalWrite(RELAY4, HIGH);
digitalWrite(RELAY5, HIGH);
digitalWrite(RELAY6, HIGH);
digitalWrite(RELAY7, HIGH);
digitalWrite(RELAY8, HIGH);
digitalWrite(RELAY9, HIGH);
digitalWrite(RELAY10, HIGH);
}
}
}
BLYNK_WRITE(V11) {
    int pinValue = param.asInt(); // Membaca nilai dari Virtual Pin V11
    if (pinValue == 1) {
        digitalWrite(RELAY1, LOW);
        digitalWrite(RELAY3, LOW);
        digitalWrite(RELAY5, LOW);
        digitalWrite(RELAY7, LOW);
        digitalWrite(RELAY9, LOW);
    }
}

```

```

digitalWrite(RELAY2, HIGH);
digitalWrite(RELAY4, HIGH);
digitalWrite(RELAY6, HIGH);
digitalWrite(RELAY8, HIGH);
digitalWrite(RELAY10, HIGH);
} else {
digitalWrite(RELAY1, HIGH);
digitalWrite(RELAY2, HIGH);
digitalWrite(RELAY3, HIGH);
digitalWrite(RELAY4, HIGH);
digitalWrite(RELAY5, HIGH);
digitalWrite(RELAY6, HIGH);
digitalWrite(RELAY7, HIGH);
digitalWrite(RELAY8, HIGH);
digitalWrite(RELAY9, HIGH);
digitalWrite(RELAY10, HIGH);
}
}
BLYNK_WRITE(V12) {
int pinValue = param.asInt(); // Membaca nilai dari Virtual Pin V12
if (pinValue == 1) {
digitalWrite(RELAY2, LOW);
digitalWrite(RELAY4, LOW);
digitalWrite(RELAY6, LOW);
digitalWrite(RELAY8, LOW);
digitalWrite(RELAY10, LOW);
digitalWrite(RELAY1, HIGH);
digitalWrite(RELAY3, HIGH);
digitalWrite(RELAY5, HIGH);
digitalWrite(RELAY7, HIGH);
digitalWrite(RELAY9, HIGH);
} else {

```

```

digitalWrite(RELAY1, HIGH);
digitalWrite(RELAY2, HIGH);
digitalWrite(RELAY3, HIGH);
digitalWrite(RELAY4, HIGH);
digitalWrite(RELAY5, HIGH);
digitalWrite(RELAY6, HIGH);
digitalWrite(RELAY7, HIGH);
digitalWrite(RELAY8, HIGH);
digitalWrite(RELAY9, HIGH);
digitalWrite(RELAY10, HIGH);
}
}
void setup() {
  // Memulai serial
  Serial.begin(115200);

  // Mengatur pin relay sebagai output
  pinMode(RELAY1, OUTPUT);
  pinMode(RELAY2, OUTPUT);
  pinMode(RELAY3, OUTPUT);
  pinMode(RELAY4, OUTPUT);
  pinMode(RELAY5, OUTPUT);
  pinMode(RELAY6, OUTPUT);
  pinMode(RELAY7, OUTPUT);
  pinMode(RELAY8, OUTPUT);
  pinMode(RELAY9, OUTPUT);
  pinMode(RELAY10, OUTPUT);

  // Set semua relay ke kondisi OFF
  digitalWrite(RELAY1, HIGH);
  digitalWrite(RELAY2, HIGH);
  digitalWrite(RELAY3, HIGH);

```

```

digitalWrite(RELAY4, HIGH);
digitalWrite(RELAY5, HIGH);
digitalWrite(RELAY6, HIGH);
digitalWrite(RELAY7, HIGH);
digitalWrite(RELAY8, HIGH);
digitalWrite(RELAY9, HIGH);
digitalWrite(RELAY10, HIGH);

// Setup LED Indikator
pinMode(LED_INDIKATOR, OUTPUT);
pinMode(LED_KUNING, OUTPUT);
pinMode(LED_HIJAU, OUTPUT);

// Memulai koneksi WiFi dan Blynk
Serial.print("Menghubungkan ke WiFi: ");
Serial.println(ssid);

// Set LED Kuning menyala dan LED Hijau mati saat koneksi WiFi belum berhasil
digitalWrite(LED_KUNING, HIGH);
digitalWrite(LED_HIJAU, LOW);

Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);

while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
  digitalWrite(LED_INDIKATOR, !digitalRead(LED_INDIKATOR)); // Kedip
selama mencoba koneksi
}

Serial.println("\nWiFi tersambung!");
Serial.print("IP Address: ");

```

```

Serial.println(WiFi.localIP());

// Koneksi WiFi berhasil, sekarang cek Blynk
while (!Blynk.connected()) {
  delay(500);
  Serial.print(".");
}

// Koneksi Blynk berhasil
Serial.println("\nKoneksi ke Blynk berhasil!");

// Set LED Hijau menyala dan LED Kuning mati setelah koneksi WiFi dan Blynk
berhasil
digitalWrite(LED_KUNING, LOW);
digitalWrite(LED_HIJAU, HIGH);

digitalWrite(LED_INDIKATOR, HIGH); // LED utama menyala ketika
terkoneksi
}

void checkConnection() {
  if (!Blynk.connected()) {
    Serial.println("Koneksi Blynk terputus!");
    digitalWrite(LED_INDIKATOR, LOW);
    digitalWrite(LED_KUNING, HIGH); // LED Kuning menyala saat Blynk
terputus
    digitalWrite(LED_HIJAU, LOW); // LED Hijau mati saat Blynk terputus
    Blynk.connect();
  }
}

void loop() {

```

```
Blynk.run();
timer.run();
checkConnection(); // Cek koneksi Blynk

// Kedip LED sebagai indikator program berjalan
static unsigned long previousMillis = 0;
const long interval = 1000; // Interval kedip 1 detik

unsigned long currentMillis = millis();
if (currentMillis - previousMillis >= interval) {
  previousMillis = currentMillis;
  digitalWrite(LED_INDIKATOR, !digitalRead(LED_INDIKATOR));
}
}
```

