

Rancang Bangun Sistem IoT pada *Smart and Economical Traffic Light* (SMILE)

IoT System Design for Smart and Economical Traffic Light

Lalu Herdinan Eka Terryatmana^[1], Fitri Bimantoro^[1], Ahmad Zafrullah Mardiansyah^[2]

^[1,2]Program Studi Teknik Informatika, Universitas Mataram

Jl. Majapahit 62, Mataram, Lombok NTB, Indonesia

Email: dinaneka2309@gmail.com, [bimo, zaf]@unram.ac.id

Abstract Traffic congestion in urban areas often occurs due to the increasing number of vehicles and limited road infrastructure. The impacts of congestion include increased emissions and decreased productivity. To address this issue, there are volunteers who help regulate traffic, known as "Pak Ogah." A study was conducted to create a traffic light system integrated with IoT as a replacement for "Pak Ogah." The study utilized experimental methods to examine the influence of machine learning in a controlled IoT system. This system, called SMILE (Smart and Economical Traffic Light), utilizes the NodeMCU ESP-32 and internet connectivity for command center access. The implementation of this IoT system aims to reduce road congestion by improving traffic light management effectiveness. Additionally, it assists the government in enforcing regulations on "Pak Ogah" and monitoring traffic conditions in a specific area. Ultimately, this system can contribute to the development of Smart Cities in Indonesia.

Key words: IoT, SMILE, NodeMCU, Smart City, Traffic Light.

I. PENDAHULUAN

Indonesia adalah negara yang memiliki populasi terbesar kedua setelah Amerika Serikat, yaitu berada pada posisi ke-4 dengan jumlah penduduk sekitar 276 juta jiwa. Banyaknya penduduk di Indonesia memengaruhi tingkat penggunaan kendaraan pribadi, terutama di kota-kota besar Indonesia.

Kemacetan merupakan salah satu masalah utama yang terjadi di beberapa daerah di Indonesia. Banyak akibat yang ditimbulkan oleh kemacetan, yaitu terjadi peningkatan polusi udara, dimana kendaraan menyumbang hampir 80% polusi udara di Indonesia, dampak negatif lainnya adalah terbuangnya waktu setiap pengguna jalan yang mengakibatkan bahan bakar yang terpakai pada kendaraan pribadi terus meningkat, hal ini kemudian mengakibatkan gas emisi bertambah.

Selain itu, kemacetan juga menimbulkan kecelakaan lalu lintas. Menurut data dari Korlantas Polri, sepanjang tahun 2022 telah tercatat 6.707 kasus kecelakaan lalu lintas dengan korban meninggal dunia sebanyak 452 orang dan total kerugian mencapai Rp. 13 Miliar lebih. Rendahnya kesadaran masyarakat dalam berlalu lintas menyebabkan pelanggaran lalu lintas banyak

terjadi, hal tersebut menjadi salah satu penyebab utama kecelakaan. Beberapa kota di Indonesia sudah menerapkan kebijakan untuk menekan angka kecelakaan akibat pelanggaran lalu lintas.

Pemerintah sudah membuat beberapa kebijakan untuk mengatasi kemacetan di beberapa kota besar di Indonesia, salah satunya yaitu dengan pengadaan lampu lalu lintas di setiap persimpangan jalan dan tempat arus lalu lintas lainnya. Lampu lalu lintas merupakan salah satu pengatur lalu lintas yang bertujuan agar lalu lintas menjadi lebih teratur sehingga dapat memberikan rasa aman kepada pengguna jalan atau pengendara dan juga dapat mengurangi kemacetan di tiap ruas jalan.

Selain kebijakan yang ditetapkan oleh pemerintah, terdapat beberapa masyarakat yang secara sukarela bertindak sebagai pengatur lalu lintas pengganti polisi untuk mengurai kemacetan dan memperlancar arus lalu lintas yang biasa disebut dengan "Pak ogah" (*volunteer* pengatur lalu lintas yang berasal dari masyarakat). "Pak ogah" ini banyak bermunculan di kota-kota yang ada di Indonesia, salah satunya Kota Mataram. Keberadaan dari "Pak ogah" ini juga dirasa membantu masyarakat dalam berlalu lintas, sehingga tidak sedikit pengendara yang memberikan uang tip kepada "Pak ogah" tersebut. "Pak ogah" ini berasal dari masyarakat yang sukarela mengatur lalu lintas tanpa mengetahui aspek-aspek hukum dari lalu lintas yang ada. Akan tetapi, menurut Korlantas dan Dinas Perhubungan, keberadaan "Pak ogah" ini bertentangan dengan peraturan perundang-undangan yang ada. Dalam waktu dekat, pihak kepolisian beserta Dinas Perhubungan juga akan menertibkan keberadaan pak ogah ini dan akan dilakukan pembinaan, edukasi, dan pelatihan yang mengarah pada kesempatan kerja yang lebih layak.

Berdasarkan permasalahan di atas, penulis mengajukan untuk membuat sebuah sistem IoT pada lalu lintas sebagai pengganti dari keberadaan pak ogah, yaitu SMILE (*Smart and Economical Traffic Light*). SMILE adalah sebuah sistem lalu lintas yang menggunakan IoT dan terintegrasi dengan teknologi *computer vision* untuk melihat volume kendaraan di suatu arus lalu lintas. SMILE terintegrasi dengan server untuk dapat menerima dan mengirim data dengan menggunakan protokol HTTP.

Perancangan IoT ini akan dilakukan menggunakan sebuah *microcontroller*, yaitu NodeMCU ESP-32.

NodeMCU ESP-32 adalah salah satu *board microcontroller* yang mirip dengan Arduino Uno, namun memiliki lebih banyak fungsi dan *powerful*. NodeMCU dilengkapi dengan *Wi-Fi module* ESP8266 bawaan sehingga cocok untuk *project* IoT yang menggunakan koneksi internet. Selain NodeMCU ESP-32, dalam sistem ini juga menggunakan ESP32-CAM yang digunakan sebagai pendeteksi objek pada persimpangan. Kelebihan ESP32-CAM adalah memiliki kamera bawaan yang cocok untuk *project* yang dikombinasikan dengan *machine learning*. Selain itu, ESP32-CAM juga memiliki konektivitas *Wi-Fi* sehingga memungkinkan perangkat untuk terhubung ke jaringan internet dengan menggunakan protokol HTTP atau MQTT. Penggunaan NodeMCU ESP-32 sebagai *microcontroller* pada sistem IoT SMILE ini bertujuan untuk membuat program yang berisi perintah untuk menerima data yang dikirimkan oleh *machine learning* dalam bentuk JSON dari model yang sudah dibuat. NodeMCU ESP-32 juga akan dihubungkan dengan mesin penggerak, yaitu motor servo yang digunakan sebagai penggerak buka tutup palang otomatis. SMILE ini nantinya akan terintegrasi juga dengan *command center* yang akan melakukan *monitoring* pada setiap arus lalu lintas dan persimpangan-persimpangan kecil yang sering menjadi titik kemacetan. Penulis berharap dengan adanya SMILE dapat mengimplementasikan penggunaan *microcontroller* dengan mengkombinasikan *machine learning* sebagai sistem *smart traffic light*.

II. TINJAUAN PUSTAKA

A. Penelitian Terkait

Yohana Tri Utami dan Yuri Rahmanto melakukan penelitian berjudul "Rancang Bangun Sistem Pintu Parkir Otomatis Berbasis Arduino dan RFID". Dalam penelitian ini, mereka menggunakan berbagai perangkat termasuk RFID Card sebagai kartu parkir, Arduino sebagai mikrokontroler, motor servo sebagai penggerak pintu palang, sensor ultrasonik untuk mendeteksi kendaraan masuk, dan LCD untuk menampilkan waktu masuk dan keluar. Hasil penelitian menunjukkan bahwa saat RFID Card dibaca oleh RFID Reader sebagai kartu parkir, data dikirim ke mikrokontroler untuk diproses sesuai dengan perintah yang telah diatur sebelumnya. Jika akses diterima, palang pintu akan terbuka dan LCD akan menampilkan waktu masuk atau keluar. Setelah kendaraan melewati sensor ultrasonik yang dipasang di belakang pintu, palang pintu akan kembali tertutup [1].

Fakhrur Rahman M. dan M. Ferdianto S. melakukan penelitian berjudul "Prototipe Palang Pintu Parkir Otomatis dan Informasi Parkir Kendaraan Roda Empat di Pondok Pesantren Nurul Jadid dengan Sensor *Infra Red* Berbasis Mikrokontroler". Penelitian ini menggunakan sensor *infra red* sebagai sensor pada slot parkir, Arduino, motor servo, dan LCD. Prototipe palang pintu parkir ini bekerja dengan mendeteksi mobil yang akan parkir dan mendekati ke sensor *infra red* pada pintu masuk. Sensor tersebut

mengirimkan sinyal ke mikrokontroler untuk membuka servo 1 (pintu masuk) dan otomatis tertutup ketika mobil masuk dan menjauhi sensor. Di area parkir, mobil memilih untuk parkir di *slot* 1 atau *slot* 2, dimana setiap *slot* memiliki sensor *infra red* yang memberikan sinyal ke mikrokontroler apakah ada mobil yang parkir atau tidak di *slot* tersebut [2].

Reski M. dan Budayawan K. telah melakukan sebuah penelitian berjudul "*Smart Traffic Light* Berbasis Arduino" yang menggunakan sensor ultrasonik untuk mengukur volume kendaraan di persimpangan. Penelitian ini berhasil mengembangkan sebuah prototipe yang dapat mengatur durasi penyalakan lampu lalu lintas di persimpangan yang mengalami kemacetan. Sistem ini membagi kondisi lalu lintas menjadi tiga, yaitu sangat macet, macet, dan sedang. Pada setiap kondisi, terdapat sensor ultrasonik yang berfungsi untuk mengendalikan lamanya penyalakan lampu hijau. Misalnya, jika sensor pada kondisi sangat macet mendeteksi keberadaan mobil, maka lampu hijau pada persimpangan tersebut akan menyala selama 90 detik. Jika sensor pada kondisi macet mendeteksi kendaraan, durasi penyalakan lampu hijau akan menjadi 60 detik. Sedangkan jika pada kondisi tidak macet terdeteksi kendaraan, lampu hijaunya akan menyala selama 30 detik. Namun, jika ketiga sensor tidak mendeteksi adanya kendaraan, lampu hijau akan menyala hanya selama 10 detik. Dengan demikian, waktu tunggu di persimpangan lainnya akan semakin berkurang [3].

Sofwan Adha M., Yacobus Padang S., dan Patimang A. telah melakukan penelitian yang berjudul "Sistem Palang Pintu Parkiran Sepeda Motor Berbasis RFID". Penelitian ini menghasilkan sebuah prototipe palang pintu untuk parkir yang menggunakan RFID tag dan RFID reader. Cara kerja prototipe ini adalah ketika RFID tag didekatkan ke RFID reader, palang pintu akan terbuka sesuai dengan waktu yang telah ditentukan. Namun, jika kendaraan masih berada di area palang pintu, palang pintu akan tetap terbuka sampai kendaraan meninggalkan area tersebut [4].

Satriadi A., Wahyudi, dan Yuli Christiono telah melakukan penelitian yang berjudul "Perancangan *Home Automation* Berbasis NodeMCU". Penelitian ini menghasilkan sebuah sistem *home automation* yang menggunakan NodeMCU sebagai basisnya. Pengujian dilakukan dengan menggunakan *provider* XL dengan sinyal HSPA+ dan Telkomsel dengan sinyal 4G. Pengujian dimulai dengan mengakses *web* server yang telah dibuat, lalu menekan tombol-tombol secara berurutan untuk mengaktifkan dan mematikan lampu, kipas angin, AC, serta membuka dan menutup pintu pagar. Dari hasil pengujian tersebut, ditemukan bahwa waktu tanggap menggunakan sinyal 4G lebih cepat dibandingkan dengan menggunakan sinyal HSPA+ [5].

Moch. Bahrul Ulum, Moch. Lutfi, dan Arif Faizin telah melakukan penelitian dengan judul "Otomatisasi Pompa Air Menggunakan NodeMCU ESP8266 Berbasis *Internet of Things*". Penelitian ini menggunakan NodeMCU ESP8266, *relay*, sensor ultrasonik, dan *access point*.

Penelitian ini menghasilkan sebuah alat pompa air otomatis yang menggunakan NodeMCU ESP8266. Pompa air akan diaktifkan saat air mencapai level 0 dan akan berhenti saat air mencapai level 4. Perhitungan ketinggian air dilakukan menggunakan sensor ultrasonik. Meskipun akurasi sensor memiliki nilai kesalahan rata-rata sebesar 14,78%, namun tetap memberikan keakuratan yang cukup baik dengan selisih $\pm 0,5$ cm pada setiap levelnya. Untuk memantau ketinggian dan status pompa air, serta mengontrolnya secara manual, dapat dilakukan melalui aplikasi yang terinstal pada *smartphone*. [6].

Penelitian mengenai Prototype *Smart Home* Dengan NodeMCU ESP8266 Berbasis IoT yang dilakukan oleh Mariza Wijayanti menghasilkan sebuah alat *automation* yang dapat mematikan perangkat elektronik dengan menggunakan beberapa sensor sebagai pendeteksi. Penelitian ini juga menggunakan aplikasi Blynk yang terintegrasi dengan alat-alat *automation* sehingga dapat *me-monitoring* dan dapat menampilkan notifikasi [7].

Mochamad Fajar Wicaksono dengan penelitian yang berjudul Implementasi Modul Wi-Fi NodeMCU ESP8266 Untuk *Smart Home* dilakukan pengujian terhadap beberapa perangkat keras dan berjalan dengan baik. NodeMCU ESP8266 dapat membaca semua masukan dari semua sensor, dapat mengontrol *relay* untuk kondisi lampu dan *relay* untuk mengaktifkan dan mematikan kipas. Selain itu, NodeMCU dapat mengirim data ke dalam *database* yang kemudian dapat ditampilkan melalui *website*. Aplikasi juga dapat mengirimkan *e-mail* notifikasi ke pengguna jika mendeteksi pergerakan [8].

Penelitian yang dilakukan oleh Ilham Santoso, Miftah Farid Adiwisastro, Bambang Kelana Simpony, Deddy Supriadi, dan Dini Silvi Purnia dengan judul Implementasi NodeMCU Dalam *Home Automation* Dengan Sistem Kontrol Aplikasi Blynk menghasilkan sebuah *home automation* yang dapat mengontrol perangkat elektronik di rumah, dapat *me-monitoring* suhu dan kelembaban suatu ruangan, serta mendeteksi asap atau gas yang berpotensi menyebabkan kebakaran dan dengan sistem kontrol aplikasi Blynk yang dapat dioperasikan dimana saja selama terhubung dengan internet [9].

B. Teori Penunjang

B.1. Lampu Lalu Lintas

Lampu lalu lintas, menurut UU No. 22/2009 tentang lalu lintas dan angkutan jalan, adalah alat pemberi isyarat lalu lintas (APILL) yang dipasang di persimpangan jalan, zebra cross, dan tempat lain untuk mengatur arus lalu lintas. Merancang lampu lalu lintas di persimpangan memiliki beberapa keuntungan, antara lain mengurangi kecelakaan tipe sudut kanan, menciptakan pergerakan lalu lintas yang teratur, menyediakan arus yang kontinu bagi kendaraan, memungkinkan pejalan kaki dan kendaraan melintasi persimpangan yang ramai, dan lebih ekonomis dibandingkan metode manual. Pada dasarnya, merancang lampu lalu lintas di persimpangan harus memperhatikan keuntungan dalam mengatur lalu lintas dengan efisien dan aman, serta mengurangi kerugian dengan memastikan koordinasi dan pengaturan yang tepat [10].

B.2. Arduino IDE

Arduino IDE adalah perangkat lunak yang disediakan oleh arduino.cc sebagai alat pengembangan *sketch* (program) untuk papan Arduino. IDE (*Integrated Development Environment*) berarti itu adalah sebuah lingkungan pengembangan terpadu yang menyediakan berbagai fitur dan fungsi dalam bentuk antarmuka berbasis menu. Dengan Arduino IDE, pengguna dapat menulis *sketch*, memeriksa kesalahan dalam *sketch*, dan mengunggah (*upload*) *sketch* yang sudah dikompilasi ke papan Arduino. Arduino IDE menggunakan bahasa pemrograman berbasis *Wiring*, yang merupakan turunan dari bahasa pemrograman C/C++. Bahasa ini memungkinkan pengguna untuk mengakses dan mengontrol perangkat keras Arduino melalui perintah-perintah sederhana yang telah disediakan. Dengan Arduino IDE, pengguna dapat dengan mudah mengembangkan program untuk berinteraksi dengan sensor, aktuator, dan komponen elektronik lainnya yang terhubung ke papan Arduino [11].

B.3. Internet of Things

Internet of Things (IoT) adalah konsep di mana semua objek di dunia fisik dapat saling berkomunikasi sebagai bagian dari sistem terpadu melalui jaringan internet. Misalnya, CCTV yang terpasang di sepanjang jalan dapat terhubung ke internet dan diintegrasikan dalam satu pusat kontrol yang mungkin berjarak puluhan kilometer atau sebuah rumah pintar yang dapat dikelola melalui *smartphone* menggunakan koneksi internet. Pada dasarnya, perangkat IoT terdiri dari sensor untuk mengumpulkan data, koneksi internet sebagai media komunikasi, dan server sebagai pusat pengumpulan informasi dari sensor dan untuk analisis lebih lanjut [12].

B.4. NodeMCU

NodeMCU adalah sebuah *board* elektronik yang berbasis *chip* ESP8266 dengan kemampuan menjalankan fungsi mikrokontroler dan juga koneksi internet (WiFi). Terdapat beberapa pin I/O sehingga dapat dikembangkan menjadi sebuah aplikasi *monitoring* maupun *controlling* pada proyek IoT [13]. NodeMCU telah menggabungkan ESP8266 ke dalam sebuah *board* yang kompak dengan berbagai fungsi layaknya mikrokontroler ditambah juga dengan kemampuan akses terhadap WiFi juga *chip* komunikasi USB to *Serial* sehingga untuk memrogramnya hanya diperlukan ekstensi kabel data mikro USB [5]. Secara fungsi modul ini hampir menyerupai dengan *platform* modul Arduino, tetapi yang membedakan yaitu dikhususkan untuk "*Connected to Internet*".

B.5. HTTP

Protokol pengiriman data dengan HTTP (*Hypertext Transfer Protocol*) adalah protokol komunikasi yang digunakan untuk *men-transfer* data melalui jaringan, khususnya pada *World Wide Web*. HTTP memungkinkan klien (misalnya *web browser*) untuk meminta dan menerima dokumen atau data dari server *web*. HTTP menggunakan model klien-server, di mana klien (misalnya *web browser*) mengirimkan permintaan (*request*) ke server

web, dan server web merespons (*response*) dengan mengirimkan data yang diminta. Permintaan dan respons ini terdiri dari *header* yang berisi informasi tentang permintaan/respons serta *body* yang berisi data aktual HTTP.

B.6. ESP32-CAM

ESP32-CAM adalah mikrokontroler yang memiliki kamera video terintegrasi dan soket kartu microSD. Papan pada ESP32-CAM ini ditenagai oleh SoC ESP32-S dari Espressif, MCU yang kuat dan dapat diprogram dengan *Wi-Fi* dan *Bluetooth* siap pakai. ESP32-CAM memiliki mikroprosesor *dual-core* 32-bit LX6, beroperasi pada 240 MHz dan bekerja hingga 600 MHz. ESP32-CAM memiliki daya komputasi yang cukup untuk tugas pembelajaran mesin sederhana dan untuk pengambilan gambar kamera serta mempersiapkan tugas lainnya. ESP32-CAM dapat digunakan dengan berbagai modul kamera, dan tersedia papan pemrograman untuk membuatnya lebih mudah digunakan.

B.7. Motor Servo

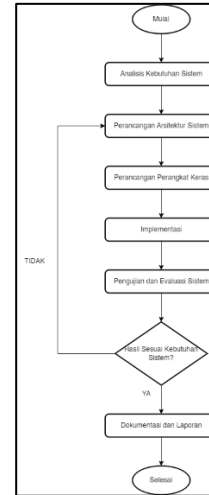
Motor servo adalah jenis motor yang dilengkapi dengan sistem umpan balik tertutup (*closed feedback*), di mana posisi motor dikembalikan ke rangkaian kontrol di dalam motor servo. Motor ini terdiri dari motor, rangkaian gigi (*gear*), potensiometer, dan rangkaian kontrol. Potensiometer digunakan untuk menentukan batas sudut putaran servo. Sudut pada sumbu motor servo dikendalikan oleh lebar pulsa yang dikirim melalui sinyal yang diterima melalui kabel motor. Motor servo memiliki kemampuan untuk berputar dengan kecepatan yang lambat, ditunjukkan oleh tingkat kecepatan putaran yang rendah. Namun, motor servo ini memiliki torsi yang kuat dikarenakan adanya gigi internal (*gear*) yang membantu meningkatkan daya torsi motor tersebut [14].

III. METODE PENELITIAN

Pada tahapan metode penelitian ini meliputi alur pelaksanaan sistem, analisis kebutuhan sistem, perancangan arsitektur sistem, konfigurasi perangkat, implementasi sistem, skenario pengujian, dan dokumentasi dan laporan.

A. Alur Pelaksanaan Sistem

Untuk melakukan penelitian Rancang Bangun Sistem IoT pada *Smart and Economical Traffic Light (SMILE)* dilakukan beberapa tahapan. Tahapan-tahapan yang dilakukan dalam penelitian dapat dilihat pada Gambar 1.



Gambar 1. Alur Pelaksanaan Sistem

Langkah-langkah penelitian yang akan dilakukan untuk membangun sistem IoT pada SMILE adalah sebagai berikut:

TABLE I. ALUR PENELITIAN

Tahap	Penjelasan
Analisis Kebutuhan Sistem	Menganalisis terhadap kebutuhan untuk merancang sistem yang meliputi seluruh perangkat apa saja yang dibutuhkan untuk perancangan dan pembangunan sistem.
Perancangan Arsitektur Sistem	Membuat konsep perancangan terhadap arsitektur dan alur kerja pada sistem.
Implementasi	Menyusun perangkat keras yang akan digunakan pada sistem.
Pengujian dan Evaluasi Sistem	Menguji sistem yang sudah dibangun dan mengevaluasi dari hasil pengujian yang dilakukan. Jika sistem yang dibangun sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya, maka akan dilanjutkan ke tahap dokumentasi dan laporan. Sebaliknya, jika sistem yang dibangun tidak sesuai dengan kebutuhan sistem yang sudah didefinisikan, maka akan dilakukan perancangan arsitektur sistem kembali sampai sesuai dengan kebutuhan sistem yang akan dibangun.
Dokumentasi dan Laporan	Mendokumentasikan selama sistem ini dibangun dan membuat pencatatan hasil dalam bentuk laporan.

B. Analisis Kebutuhan Sistem

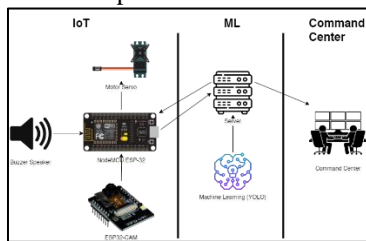
Pada tahap analisis kebutuhan sistem, akan dilakukan analisis terhadap kebutuhan sistem IoT yang diperlukan pada SMILE. Analisis ini akan mencakup analisis kebutuhan terhadap peralatan dan bahan yang akan digunakan.

TABLE II. ANALISIS KEBUTUHAN SISTEM

Perangkat Keras	Perangkat Lunak
NodeMCU ESP32	Arduino IDE
ESP32 CAM	Windows 11
Motor Servo	Web Server (Local)
Lampu LED	
Kabel <i>Junper</i>	
<i>Breadboard</i>	
<i>Buzzer Speaker</i>	

C. Perancangan Arsitektur Sistem

Dalam perancangan arsitektur, dilakukan upaya untuk merancang struktur dan alur kerja sistem IoT yang akan dibangun pada SMILE. Rancangan arsitektur sistem yang dihasilkan bisa dilihat pada Gambar 2.



Gambar 2. Rancangan Arsitektur Sistem

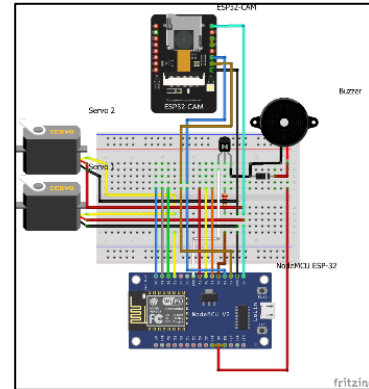
Sistem menggunakan beberapa perangkat keras, seperti NodeMCU ESP-32, motor servo, ESP32-CAM camera module, dan speaker. Kemudian menggunakan server untuk mengirimkan data ke bagian *command center* agar dapat mengetahui informasi terkait arus lalu lintas pada persimpangan yang sudah ditanamkan sistem IoT tersebut. Adapun proses yang dilakukan, yaitu:

TABLE III. PROSES PADA PERANGKAT KERAS

Perangkat Keras	Fungsi
NodeMCU ESP32	Memproses data yang diterima dalam bentuk data JSON dan mengirimkan data ke perangkat keras lainnya agar dapat memberikan respon selanjutnya.
ESP32 CAM	Menangkap dan mengirimkan gambar objek ke server.
Motor Servo	Membuka dan menutup palang sesuai dengan data JSON yang sudah diproses di NodeMCU.
Lampu LED	Menyalakan lampu LED sebagai representasi lampu lalu lintas berwarna merah, kuning, dan hijau.
<i>Buzzer Speaker</i>	Mengeluarkan suara sebagai tanda palang akan tertutup dan lampu akan berwarna merah.
<i>Breadboard</i>	Untuk membuat dan merancang seluruh komponen perangkat keras agar bisa terhubung satu sama lain.

D. Rancangan Perangkat Keras pada NodeMCU

Pada rangkaian ini, terdapat perangkat keras yang terhubung pada NodeMCU ESP-32 untuk membuat sistem *smart traffic light* yang dapat dilihat pada Gambar 3.



Gambar 3. Konfigurasi Perangkat

Pada Gambar 3 terdapat rangkaian perangkat keras untuk sistem yang terbuat, terdiri dari NodeMCU, motor servo, *buzzer speaker*, ESP32 CAM yang akan dirangkai menjadi satu rangkaian sehingga akan terhubung satu sama lain.

E. Implementasi Sistem

Setelah tahap perancangan selesai, tahap selanjutnya adalah tahap implementasi dari sistem yang akan dibangun. Tahap implementasi ini akan terbagi menjadi tiga bagian, yaitu penyusunan perangkat keras, pembangunan aplikasi kontrol, dan pembangunan sistem secara keseluruhan sebagai berikut:

E.1. Penyusunan Perangkat Keras

Pada tahap penyusunan perangkat keras, NodeMCU ESP-32, motor servo, *buzzer speaker*, dan ESP32-CAM camera module akan disusun menjadi satu rangkaian perangkat keras. Proses untuk merangkai alat ini disesuaikan dengan rancangan pada tahap rancangan perangkat keras sistem.

E.2. Pembangunan Kontrol Aplikasi

Pada tahap pembangunan kontrol aplikasi, rancangan kontrol aplikasi akan diimplementasikan ke dalam *microcontroller* NodeMCU ESP-32 menggunakan bahasa pemrograman C++, melalui Arduino IDE sebagai alat bantu.

E.3. Pembangunan Sistem

Pada tahap pembangunan sistem, sistem IoT pada SMILE akan terintegrasi dengan server internet. Protokol yang digunakan adalah protokol HTTP, guna melakukan *monitoring* pada setiap persimpangan yang ditanami sistem IoT pada SMILE.

F. Skenario Pengujian

Pada tahap skenario pengujian, dilakukan pengujian terhadap sistem yang sudah dibangun. Pada penelitian ini teknik pengujian yang digunakan, yaitu pengujian terhadap perangkat keras di lingkungan simulasi.

F.1. Skenario Pengujian Perangkat Keras

Pada pengujian perangkat keras, dilakukan pengecekan terhadap kinerja ESP-32 Camera Module dalam mengambil gambar, pengolahan data yang terhubung dengan NodeMCU ESP-32, pengecekan terhadap fungsionalitas dari motor servo yang terhubung dengan

NodeMCU ESP-32 untuk melakukan perputaran, pengecekan fungsionalitas *speaker* yang terhubung oleh NodeMCU ESP-32 untuk mengeluarkan suara, serta pengecekan lampu LED untuk memberikan peringatan. Pengujian perangkat keras dapat dilihat pada Tabel IV.

TABLE IV. SKENARIO PENGUJIAN PERANGKAT KERAS

Perangkat Keras	Fungsi
NodeMCU ESP-32	Pengujian dilakukan untuk melihat apakah sistem dapat menerima <i>input</i> yang sudah diolah dan memberikan <i>output</i> berupa instruksi pada perangkat yang terhubung, yaitu motor servo dan <i>buzzer speaker</i> .
Motor Servo	Pengujian pada motor servo dilakukan untuk melihat apakah motor servo dapat menerima input dari NodeMCU ESP-32 untuk dapat menggerakkan palang.
ESP-32 Camera Module	Pengujian pada ESP-32 Camera Module dilakukan untuk melihat apakah sistem dapat mendeteksi objek kendaraan dan mengirimkannya ke server.
Lampu LED	Pengujian pada lampu LED dilakukan untuk melihat apakah lampu dapat menyala sesuai dengan instruksi yang sudah diterima oleh NodeMCU ESP-32.
Buzzer Speaker	Pengujian pada <i>buzzer speaker</i> dilakukan untuk melihat apakah sistem dapat mengeluarkan suara setelah menerima <i>input</i> dari NodeMCU ESP-32 atau tidak.

F.2. Skenario Pengujian di Lingkungan Simulasi

Pada pengujian di lingkungan simulasi akan dilakukan beberapa fungsionalitas perangkat keras yang dikombinasikan dengan *machine learning* dan server internet. Pengujian ini bertujuan untuk melakukan pengecekan terhadap perangkat keras apakah dapat menerima instruksi dari kondisi yang telah diberikan oleh *machine learning* dan perangkat keras dapat mengirim data ke server. Skenario pengujian di lingkungan simulasi dibagi menjadi beberapa bagian, antara lain:

1. Membuat simulasi rancangan perangkat keras dengan alat-alat yang sudah ditentukan.
2. Pengujian pengiriman hasil *capture image* pada ESP32-CAM ke server.
3. Implementasi pengiriman data JSON dari model *machine learning* ke server.
4. Pengujian pengiriman data JSON dari server ke NodeMCU ESP-32 dengan protokol HTTP.
5. Pengujian respon perangkat keras dari data JSON yang sudah dikirimkan.

G. Pencatatan Status

Pencatatan status dilakukan untuk mengetahui respon dari perangkat keras yang telah diberikan *input* dari hasil *machine learning*, dimana data yang diambil adalah nilai kondisi dalam bentuk JSON. Pencatatan status dapat dilihat pada Tabel V.

TABLE V. PENCATATAN STATUS

Skenario	Kondisi	Durasi	Dokumentasi
Simpang 1 Terbuka	5	25 detik	-
Simpang 1 Tertutup	-	-	-
Simpang 2 Terbuka	4	20 detik	-
Simpang 2 Tertutup	-	-	-
Simpang 3 Terbuka	3	15 detik	-
Simpang 3 Tertutup	-	-	-
Simpang 4 Terbuka	2	10 detik	-
Simpang 4 Tertutup	-	-	-

H. Dokumentasi dan Laporan

Tahap dokumentasi dan laporan melibatkan penyusunan dokumentasi berdasarkan hasil pengujian sistem serta menyimpulkan temuan-temuan yang terdapat dalam dokumentasi tersebut. Kesimpulan yang diperoleh akan menjadi acuan penting untuk pengembangan sistem selanjutnya.

IV. HASIL DAN PEMBAHASAN

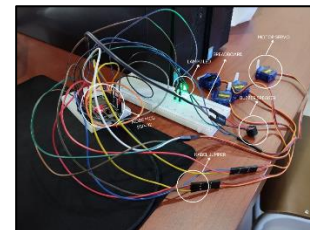
Pada hasil dan pembahasan dari penelitian yang berjudul “Rancang Bangun Sistem IoT pada *Smart and Economical Traffic Light (SMILE)*” menghasilkan beberapa seperti penerapan arsitektur sistem dan hasil pengujian dari sistem yang sudah dibuat.

A. Implementasi Sistem

Pada tahapan ini, akan dibahas mengenai bagaimana implementasi perancangan perangkat keras dan pembuatan sistem berdasarkan perancangan yang telah dijelaskan sebelumnya.

A.1. Implementasi Rancangan Perangkat Keras

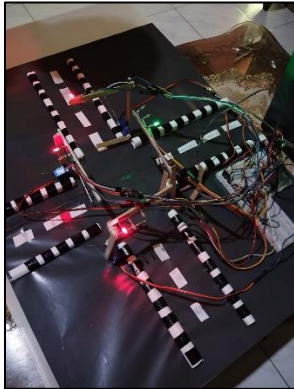
Pada tahap ini menggunakan ESP-32 Camera yang digunakan untuk mendeteksi objek kendaraan, NodeMCU ESP-32 yang terhubung dengan motor servo, lampu LED, dan *buzzer speaker* sebagai perangkat keras (*hardware*). Rangkaian perangkat keras yang digunakan dapat dilihat pada Gambar 4.



Gambar 4. Rancangan Perangkat Keras

Pada tahap implementasi ini, ada beberapa komponen utama yang digunakan dan dipasangkan pada NodeMCU ESP-32, yaitu motor servo, lampu LED, dan *buzzer*

speaker. Fungsi dari NodeMCU ESP-32 adalah untuk menerima *output* dari hasil model *machine learning* yang sudah di-*parsing* dalam format JSON dan NodeMCU ESP-32 akan memberikan instruksi sesuai *output* yang ada untuk bisa menggerakkan motor servo yang telah ditentukan, menyalakan lampu LED yang berfungsi sebagai lampu lalu lintas, dan membunyikan *buzzer speaker* sebagai tanda bahwa palang akan tertutup. Realisasi perangkat keras yang telah dirancang dapat dilihat pada Gambar 5.

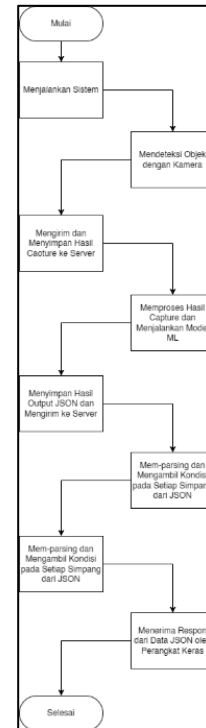


Gambar 5. Realisasi Perangkat Keras

Dapat dilihat pada Gambar 5 terdapat realisasi perangkat keras dengan menggunakan NodeMCU ESP-32, motor servo, kabel *jumper*, *breadboard*, *buzzer speaker*, dan lampu LED sudah dirangkai menjadi satu rangkaian. Penggunaan ESP32 CAM tidak dimasukkan ke dalam realisasi perangkat keras, melainkan hanya diuji untuk pengiriman gambar ke server dikarenakan ESP32 CAM tidak memiliki objek kendaraan secara *real/nyata* yang dideteksi karena kurangnya *resource* dari *machine learning* yang diberikan.

B. Implementasi Pembuatan Control Application

Pengimplementasian aplikasi kontrol menggunakan bahasa pemrograman C dengan menggunakan berbagai *library* dan modul yang dapat digunakan untuk memudahkan pengembangan sistem ini. Selain itu, sebuah *flowchart* program juga dibuat untuk memberikan gambaran tentang alur program seperti yang terlihat pada Gambar 6.



Gambar 6. Flowchart Sistem

Berdasarkan *flowchart* pada Gambar 6 pembangunan *control application* diawali dengan menjalankan sistem. Selanjutnya, ESP-32 *Camera* akan melakukan *capture* dan menyimpan hasil *capture* di server. Kemudian, dilakukan pendeteksian dari hasil *capture* yang sudah disimpan di server dan akan dilihat kendaraan yang telah terdeteksi mengisi garis, maka sistem akan mengirimkan *output* yang berisi banyak garis yang terisi kendaraan tersebut ke NodeMCU ESP-32. Selanjutnya, NodeMCU ESP-32 akan memberikan instruksi untuk menggerakkan motor servo untuk membuka palang atau menutup palang, menyalakan lampu LED sebagai lampu lalu lintas, dan membunyikan *buzzer speaker* untuk menandakan palang akan tertutup.

C. Pengujian Analisis Sistem

Pada tahap ini, evaluasi dilakukan terhadap sistem secara menyeluruh untuk memastikan kepatuhan sistem terhadap desain yang telah disusun sebelumnya. Terdapat beberapa langkah yang dilakukan dalam proses pengujian, yakni meliputi evaluasi perangkat keras dan pengujian pengiriman *output* dari model *machine learning* ke NodeMCU ESP-32.

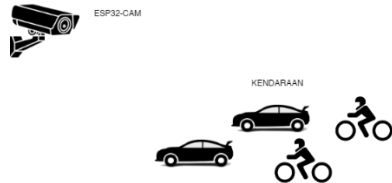
C.1. Pengujian Perangkat Keras

Pada tahap pengujian perangkat keras, dilakukan evaluasi terhadap kemampuan perangkat keras ESP-32 *Camera* dalam mendeteksi objek. Selanjutnya, dilakukan evaluasi terhadap kemampuan NodeMCU ESP-32 untuk menerima data *output* dari model *machine learning*. Selain itu, dilakukan evaluasi terhadap motor servo dalam menerima instruksi untuk bergerak membuka dan menutup, evaluasi terhadap lampu LED untuk menyala secara

otomatis sesuai instruksi yang telah diberikan, dan evaluasi terhadap *buzzer speaker* untuk mengeluarkan bunyi.

C.2. Pengujian ESP32-CAM

Pada pengujian ESP32-CAM dilakukan dengan meng-*capture* atau menangkap gambar dalam interval waktu tertentu untuk memantau kondisi kendaraan pada persimpangan. Selanjutnya, hasil *capture* gambar akan dikirimkan dan disimpan ke server dengan menggunakan protokol HTTP. Ilustrasi dapat dilihat pada Gambar 7.



Gambar 7. Pengujian ESP32-CAM

C.3. Pengujian Garis Batas Objek Kendaraan

Terdapat *frame* atau kotak jika kendaraan yang terdeteksi di dalam kotak tersebut, maka sistem akan mengirimkan kondisi berisi banyak kotak yang terisi oleh kendaraan. Tujuannya dibuatkan kotak adalah untuk mengetahui berapa banyak kendaraan yang terhitung mengisi kotak tersebut. Tampilan *frame* atau kotak dapat dilihat pada Gambar 8.

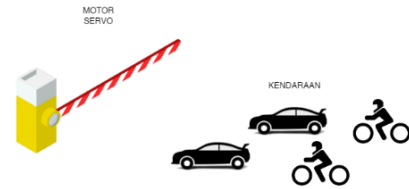


Gambar 8. Pengujian Garis Batas

Gambar 8 merupakan tampilan *frame* yang terdapat kotak yang digunakan untuk menghitung banyak kendaraan yang dideteksi mengisi kotak tersebut sehingga hanya fokus pada objek yang berada di dalam kotak tersebut untuk menghindari kesalahan pengiriman data.

C.4. Pengujian Motor Servo

Pada pengujian motor servo akan dilakukan pengujian untuk melihat apakah motor servo dapat bergerak sesuai dengan kondisi atau instruksi yang ada pada suatu persimpangan. Misal, simpang 1 akan terbuka terlebih dahulu dalam waktu beberapa detik tergantung kondisi dari data JSON yang ada. Selanjutnya, simpang 2 akan terbuka setelah simpang 1 tertutup dan waktunya tergantung kondisi dari data JSON yang ada. Ilustrasi dapat dilihat pada Gambar 9.

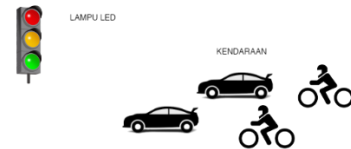


Gambar 9. Pengujian Motor Servo

Gambar 9 merupakan ilustrasi pengujian untuk motor servo apakah dapat bergerak sesuai kondisi yang ada. Pengujian akan dilakukan dengan 4 motor servo, karena akan diasumsikan dengan simpang 4 lalu lintas. Pergerakan motor servo akan sesuai dengan aturan berlawanan arah jarum jam (*counter clock wise*).

C.5. Pengujian Lampu LED

Pada pengujian lampu LED dengan mengeluarkan lampu berwarna merah, kuning, dan hijau secara bergantian yang disimulasikan sebagai lampu lalu lintas untuk memberikan tanda ke pengendara. Ketika lampu berwarna kuning, servo akan perlahan menutup sampai lampu berwarna merah yang artinya pengendara diharuskan berhenti. Sedangkan, lampu berwarna hijau, maka servo akan terbuka yang artinya pengendara dapat melaju. Ilustrasi dapat dilihat pada Gambar 10.

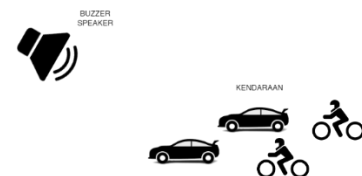


Gambar 10. Pengujian Lampu LED

Pengujian lampu LED akan dilakukan untuk melihat apakah lampu akan menyala warna hijau jika servo terbuka, warna kuning jika servo perlahan menutup, dan warna merah jika servo sudah tertutup.

C.6. Pengujian Buzzer Speaker

Pada pengujian sistem *buzzer speaker* dengan mengeluarkan suara yang berfungsi sebagai penanda bahwa lampu akan berwarna merah dan servo akan menutup yang artinya pengendara diharuskan untuk memperlambat laju kendaraannya. Ilustrasi dapat dilihat pada Gambar 11.



Gambar 11. Pengujian Buzzer Speaker

Pengujian *buzzer speaker* akan dilakukan untuk melihat apakah *buzzer speaker* akan mengeluarkan suara saat lampu berwarna kuning dan servo menutup perlahan.

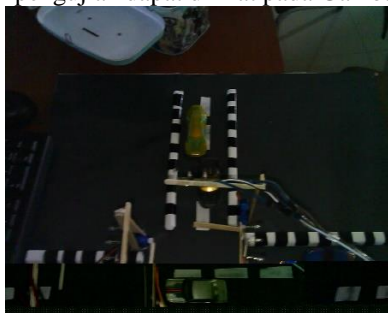
Tujuannya adalah sebagai pemberitahuan ke pengendara bahwa lampu akan berwarna merah dan servo tertutup.

D. Hasil Pengujian Alat

Pengujian dilakukan untuk mengetahui apakah alat dapat mendeteksi kendaraan, dapat menerima *output* berupa JSON dari model *machine learning*, dan dapat melakukan respon setelah menerima data dari model *machine learning*. Pengujian dilakukan di dalam ruangan yang disimulasikan sebagai persimpangan dan terdapat beberapa kendaraan dalam tiap persimpangan.

D.1. Hasil Pengujian ESP32-CAM

Pengujian dilakukan untuk menguji jangkauan kamera dalam mendeteksi objek yang ada. Pengujian ini dilakukan dengan membuat simulasi maket representasi dari persimpangan yang sudah terpasang palang dan lampu LED. Hasil pengujian dapat dilihat pada Gambar 12.

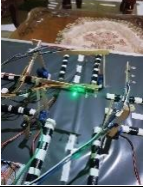
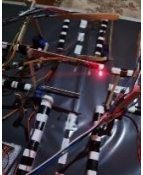



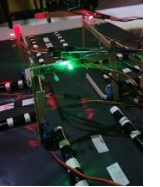
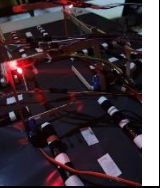
Gambar 12. Hasil Pengujian ESP32-CAM

D.2. Hasil Pengujian Respon Perangkat Keras

Pengujian dilakukan untuk menguji respon perangkat keras dalam menerima *output* berupa data JSON dari model *machine learning*. Pengujian ini melibatkan beberapa skenario dengan tujuan untuk menentukan kondisi (jumlah garis batas yang terisi kendaraan) dengan menggunakan ESP32-CAM. Pada pengujian ini dilakukan pengukuran kemampuan sistem dalam menerima respon dari kondisi berupa JSON yang didapat dari model *machine learning*. Pengujian respon perangkat keras dapat dilihat pada Tabel VI.

TABLE VI. HASIL PENGUJIAN

Skenario	Kondisi	Durasi	Dokumentasi
Simpang 1 Terbuka	5	25 detik	
Simpang 1 Tertutup	-	-	

Simpang 2 Terbuka	4	20 detik	
Simpang 2 Tertutup	-	-	
Simpang 3 Terbuka	3	15 detik	
Simpang 3 Tertutup	-	-	
Simpang 4 Terbuka	2	10 detik	
Simpang 4 Tertutup	-	-	

Pada dilakukan beberapa skenario pengujian. Pada pengujian pertama, yaitu simpang 1 dengan nilai kondisi 5 akan terbuka selama 25 detik dan akan menyalakan lampu hijau selama 25 detik. Selanjutnya, simpang 1 akan tertutup setelah durasi habis dan akan menyalakan lampu merah serta membunyikan *buzzer speaker*. Pada pengujian kedua juga dilakukan hal yang sama pada simpang 2 dengan kondisi 4, pengujian ketiga pada simpang 3 dengan kondisi 3, dan pengujian keempat pada simpang 4 dengan kondisi 2.

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Dari hasil penelitian dan pengujian yang telah dilakukan, dengan menggunakan modul ESP32 Camera dapat dilakukan pendeteksian objek kendaraan dalam suatu persimpangan. Dalam penelitian yang dilakukan ESP32 Camera dapat menangkap objek dan mengirimkannya ke server yang sudah dibuat dengan menggunakan HTTP. Pengiriman gambar dilakukan selama interval 20 detik

untuk tiap persimpangan. Gambar yang ditangkap menggunakan resolusi maksimal atau sebesar 2 Megapixel. Setelah gambar berhasil di *capture*, server akan memberikan *response* berupa nama file gambar yang berisi *timestamp* dari diambilnya gambar tersebut. Penggunaan NodeMCU ESP32 dapat diimplementasikan dengan kombinasi beberapa perangkat keras, seperti motor servo, lampu LED, dan *buzzer speaker* dengan menghubungkannya ke pin yang ada di NodeMCU ESP32. NodeMCU ESP32 juga dapat menerima instruksi dari data JSON dari server dengan cara melakukan *parsing* terlebih dahulu agar dapat membaca data JSON. Setelah itu, perangkat keras lainnya melakukan respon dari hasil *parsing* data JSON. Pengiriman dapat dilakukan dengan menggunakan protokol HTTP dengan beberapa konfigurasi.

B. Saran

Jika dilakukan penelitian lebih lanjut terkait dengan penelitian ini ke depannya, disarankan untuk dapat menggunakan *microcontroller* lebih baik dari NodeMCU ESP32 untuk mendapatkan hasil kecepatan pemrosesan data yang lebih bagus. Selain itu, diharapkan dapat menggunakan kamera yang lebih bagus seperti IP CCTV karena memiliki deteksi objek yang lebih akurat, *real-time*, dan memiliki kualitas gambar yang lebih baik. Dalam sistem *smart traffic light* ini juga disarankan dapat mengimplementasikan sensor suara atau sensor lainnya, agar kedepannya bisa mendeteksi suara dari *ambulance* atau mobil yang memiliki kepentingan darurat sehingga bisa memberikan respon secara *real-time* ke sistem. Dalam pengiriman data secara *real-time* diharapkan sistem *smart traffic light* ini dapat mengimplementasikan teknologi *cloud computing* atau dengan menggunakan protokol MQTT yang lebih baik karena dapat terhubung dengan sistem *cloud*. Selain itu, penting juga dilakukan pengujian secara nyata yang dapat dilakukan di persimpangan yang nyata atau diterapkan di lingkungan kampus agar dapat merepresentasikan kondisi yang nyata dan sebenarnya sehingga dapat memberikan informasi yang akurat untuk menunjang kinerja sistem tersebut. Kemudian, diharapkan kedepannya pengiriman gambar ke bagian *machine learning* bisa dapat dilakukan secara *real-time* dengan *delay* yang seminimal mungkin untuk dapat memberikan hasil yang maksimal.

DAFTAR PUSTAKA

- [1] Y. T. Utami and Y. Rahmanto, "Rancang Bangun Sistem Pintu Parkir Otomatis Berbasis Arduino dan RFID," *Jurnal Teknologi dan Sistem Tertanam*, vol. 02, no. 02, pp. 25–35, 2021.
- [2] M. F. Rahman, Sulistiyanto, and S. Ferdianto, "Prototipe Palang Pintu Parkir Otomatis dan Informasi Parkir Kendaraan Roda Empat di Pondok Pesantren Nurul Jadid dengan Sensor Infra Red Berbasis Mikrokontroler," *18 JEECOM*, vol. 1, no. 1, 2019.
- [3] M. Reski and K. Budayawan, "Smart Traffic Light Berbasis Arduino," *Jurnal Vocational Teknik Elektronika dan Informatika*, vol. 9, no. 3, Sep. 2021, [Online]. Available: <http://ejournal.unp.ac.id/index.php/voteknika/>
- [4] M. S. Adha, S. Y. Padang, and A. A. Patimang, "Sistem Palang Pintu Parkiran Sepeda Motor Berbasis RFID," *Journal Dynamic saint*, vol. 6, no. 1, 2021, doi: 10.47178/dynamicsaint.v5xx.xxxx.
- [5] A. Satriadi and Y. Christiyono, "Perancangan Home Automation Berbasis NodeMCU," *Transient: Jurnal Ilmiah Teknik Elektro*, vol. 8, no. 1, pp. 64–71, Jul. 2019, [Online]. Available: <https://ejournal3.undip.ac.id/index.php/transient>
- [6] M. B. Ulum, M. Lutfi, and A. Faizin, "Otomatisasi Pompa Air Menggunakan NodeMCU ESP8266 Berbasis Internet of Things (IoT)," *Jurnal Mahasiswa Teknik Informatika*, vol. 6, no. 1, 2022.
- [7] M. Wijayanti, "Prototype Smart Home Dengan NodeMCU ESP8266 Berbasis IOT," *Jurnal Ilmiah Teknik*, vol. 1, no. 2, pp. 101–107, May 2022.
- [8] M. F. Wicaksono, "Implementasi Modul WiFi NodeMCU ESP8266 Untuk Smart Home," *Jurnal Teknik Komputer Unikom-Komputika*, vol. 6, no. 1, 2017.
- [9] I. Santoso, M. F. Adiwisastro, B. K. Simpony, D. Supriadi, and D. S. Purnia, "Implementasi NodeMCU Dalam Home Automation Dengan Sistem Kontrol Aplikasi Blynk," *JURNAL SWABUMI*, vol. 9, no. 2, pp. 32–40, Mar. 2021.
- [10] T. Kurniati, A. Latif, and E. E. Putri, "Evaluasi dan Perencanaan Lampu Lalu Lintas Pada Simpang Jalan Syekh Umar Khalil-Bypass Kota Padang," *Jurnal Rekayasa Sipil (JRS-Unand)*, vol. 16, no. 1, p. 49, Apr. 2020, doi: 10.25077/jrs.16.1.49-64.2020.
- [11] P. Widya Kumara, "Robot LINE Follower Berbasis Mikrokontroler Arduino UNO ATMEGA328," *Jurnal Informanika*, vol. 5, no. 1, 2019.
- [12] Y. Efendi, "Internet of Things (IoT) Sistem Pengendalian Lampu Menggunakan Raspberry PI Berbasis Mobile," *Jurnal Ilmiah Ilmu Komputer*, vol. 4, no. 1, 2018, [Online]. Available: <http://ejournal.fikom-unasman.ac.id>
- [13] N. H. L. Dewi, M. F. Rohmah, and S. Zahara, "Prototype Smart Home Dengan Modul NodeMCU ESP8266 Berbasis Internet of Things (IoT)," Thesis, Universitas Islam Majapahit, Mojokerto, 2018.
- [14] A. Hilal and S. Manan, "Pemanfaatan Motor Servo Sebagai Penggerak CCTV Untuk Melihat Alat-Alat Monitor dan Kondisi Pasien di Ruang ICU," *Gema Teknologi*, vol. 17, no. 2, 2012.