

Prototipe Sistem Parkir Berbasis Internet of Things dengan Live Dashboard MQTT Server

*Prototype of Parking System Based on Internet of Things
with MQTT Server Live Dashboard*

Ida Bagus Irawan Purnama¹, I Wayan Raka Ardana², I Gede Made Putra Suardana³, I Dewa Gede Dodi Pranata⁴, I Gusti Alit Wiraguna Jaya⁵

^{1,2,3,4,5}Program Studi Teknik Otomasi, Jurusan Teknik Elektro, Politeknik Negeri Bali

E-mail: ¹ida.purnama@pnb.ac.id, ²rakawyn@pnb.ac.id, ³putrasuardana6@gmail.com,

⁴dodikpranata64@gmail.com, ⁵alitwiraguna01@gmail.com

Abstrak

Pertumbuhan penggunaan kendaraan bermotor umumnya tidak dibarengi dengan ketersediaan lahan parkir yang memadai, terutama di daerah perkotaan. Hal ini memicu masalah sulitnya mencari slot parkir pada jam-jam sibuk. Oleh karena itu pengaturan slot parkir terutama di gedung parkir perlu dilakukan. Langkah ini bisa dijalankan dengan penerapan teknologi *Internet of Things* (IoT). Penelitian ini mengusulkan penerapan IoT pada sistem parkir berbasis *live dashboard* dengan NodeMCU ESP8266 dan MQTT Server. Sistem parkir ini ditujukan untuk mengetahui jumlah kapasitas parkir yang *available* dari seluruh kapasitas parkir yang ada pada sebuah gedung parkir dan waktu pendudukan setiap slot parkir secara *real-time*. Kemudian, pengaturan kendaraan yang bisa masuk adalah berdasarkan kapasitas parkir yang masih tersedia tersebut. Sistem parkir ini terdiri dari dua komponen yang saling terhubung satu sama lain yaitu komponen pengendali sistem dan komponen manajemen parkir. Dalam hal ini MQTT Server Adafruit IO Platform digunakan untuk pengiriman data kontrol dan monitoring slot parkir yang tersedia. Dashboard MQTT Server ini juga dapat digunakan sebagai *display* untuk menampilkan status kapasitas parkir yang tersedia pada bagian depan gedung parkir dengan layar besar sehingga dapat dilihat oleh pengguna. Teknologi parkir ini diharapkan dapat membantu baik pengguna dalam mencari slot parkir maupun pengelola dalam mengatur usaha parkir mereka.

Kata kunci: IoT; sistem parkir; live dashboard; MQTT Server; NodeMCU ESP8266

Abstract

The growth of vehicles usage is generally not accompanied by the availability of adequate parking spaces, especially in urban areas. This triggers the problem of finding parking slots during peak hours. Therefore, the arrangement of parking slots, particularly in the parking building, is needed. This can be carried out with the implementation of Internet of Things (IoT) technology. Hence, this study proposes a prototype of live dashboard-based parking system with NodeMCU ESP8266 and MQTT Server. This system aims to determine the availability of parking capacity in a parking building and the occupation times of each parking slot in real-time. Then, the arrangement of vehicles that can enter is based on the available parking capacity. This parking system consists of two components namely the control component and the management component. Here, the Adafruit IO Platform MQTT Server is used for data transmission and monitoring. The MQTT Server dashboard can also be used as a display to inform the available parking capacity at the front of the parking building with a large screen so that it can be seen by the user. It is expected that it can help both users in finding parking slots and managers in managing their parking business.

Keywords: IoT; parking system; live dashboard; MQTT Server; NodeMCU ESP8266

1.PENDAHULUAN

Alat transportasi pada saat ini sudah menjadi suatu kebutuhan pokok bagi masyarakat dari berbagai kalangan mulai dari kalangan atas sampai kalangan bawah. Oleh karena itu alat transportasi bertumbuh sangat cepat. Kondisi ini sering menimbulkan ketidakseimbangan antara kendaraan yang ada dengan tempat parkir yang tersedia. Ketersediaan lahan parkir yang luas tentu menjadi kebutuhan untuk menangani bertambahnya penggunaan alat transportasi tersebut. Kenyataan yang terjadi di lapangan saat ini masih banyak masalah dalam sistem parkir. Sering kali terlihat bahwa kapasitas parkir cenderung kurang dibanding kuantitas kendaraan khususnya di tempat keramaian seperti pasar, pertokoan, rumah sakit, perkantoran, dan juga tempat-tempat pendidikan. Kekurangan slot parkir ini terjadi terutama untuk kendaraan roda empat yang membutuhkan tempat yang lebih luas. Untuk itu diperlukan manajemen parkir guna memaksimalkan *usability* tempat yang terbatas.

Secara lebih spesifik, masalah parkir yang sering ditemui saat ini adalah kesulitan menemukan informasi ketersediaan zona kosong pada tempat parkir umum. Sulitnya menemukan sistem informasi parkir tersebut menyebabkan pengguna kendaraan kesulitan untuk mencari atau mengetahui slot parkir yang tersedia. Apalagi jika lahan parkir dalam keadaan padat. Berdasarkan masalah tersebut, maka sistem parkir yang dapat menampilkan informasi ketersediaan slot parkir secara *real-time* sangat diperlukan [1]. Sehingga jika setiap kendaraan yang akan memasuki lahan parkir, pemilik kendaraan akan mengetahui informasi zona parkir yang tersedia. Hal ini tentu sangat membantu pengguna parkir jika nantinya lahan parkir tersebut sedang penuh [2]. Selain itu, penelitian ini juga ditujukan untuk menjadi bagian dari salah satu teknologi *smart city* yang dapat membantu pemerintah kota dalam memberi pelayanan terbaik kepada masyarakat.

Penelitian terkait perancangan dan penerapan teknologi sistem parkir ini sudah dilakukan oleh beberapa peneliti sebelumnya. Namun, umumnya alat yang digunakan lebih banyak berbasis Arduino Uno. Penelitian [3] merancang *smart parking* dengan pengendali sistem Raspberry Pi dan Arduino Uno R3. Penelitian ini mengusulkan sistem parkir dimana pemilik kendaraan akan diarahkan ke tempat parkir yang kosong. Sedangkan penelitian [4] mengajukan sistem parkir berbasis web dengan aplikasi yang dapat membedakan kendaraan keluarga dan pengunjung biasa dengan tarif yang berbeda. Ini diklaim akan mempermudah petugas ketika menentukan pembayaran tarif parkir. Penelitian [5] merancang sistem parkir dengan Arduino menggunakan sensor ultrasonik. Cara kerja sistem parkir ini adalah dengan mendeteksi adanya gerakan yang dirasakan oleh sensor, selanjutnya sensor mengirimkan sinyal ke modul kontrol Arduino. Tetapi sistem ini hanya dapat diterapkan pada kendaraan roda empat. Lebih lanjut, penelitian tentang *smart parking* menggunakan Arduino Uno dengan menampilkan indikator lahan parkir yang kosong antara lain dilakukan pada penelitian [6] dan [7].

Di lain pihak, penelitian [8] mengusulkan *smart parking* berbasis aplikasi Android yang bertujuan membantu pengemudi untuk menemukan lahan parkir kosong. Dengan demikian maka pengemudi tidak harus mendatangi tempat parkir untuk mencari slot. Sama dengan penelitian sebelumnya, penelitian [9] juga menawarkan *smart parking* untuk membantu pengemudi yang kesulitan menemukan lahan parkir yang masih kosong menggunakan sensor cahaya. Sedangkan penelitian [10] menggunakan sensor *fingerprint* sebagai pengganti dari RFID (*Radio Frequency Identification*) yang rentan hilang atau rusak.

Perancangan sistem parkir dengan Arduino juga dilakukan oleh [11] yang mengimplementasikan pengukuran jarak pada mobil dengan gelombang ultrasonik. Dengan sistem ini maka akan didapatkan output *value* dalam satuan sentimeter. Selain itu, terdapat juga perancangan parkir otomatis dengan konsep *Internet of Things* (IoT) bertujuan untuk mengembangkan sistem reservasi dan pemantauan lokasi lahan parkir. Kemudian hasilnya ditampilkan pada sebuah *smartphone* dengan koneksi internet menggunakan mikrokontroler Arduino [12]. Sama seperti penelitian [12], penelitian [13] juga menggunakan sensor ultrasonik untuk memonitoring lahan parkir dengan Arduino Uno sebagai mikrokontroler. Pada penelitian lain, [14] menggunakan mikrokontroler AT89S51 sebagai sistem kendali utama pada

pemrosesan data jarak parkir yang dihasilkan dari sensor ultrasonik. Sementara itu, penelitian [15] menggunakan sistem parkir berbasis QR Code, sedangkan penelitian [16] mengusulkan sistem parkir berbasis web server yang dilengkapi dengan tampilan *seven segment*. Namun, *seven segment* memiliki keterbatasan karena hanya mampu menampilkan 16 karakter heksadesimal.

Pada penelitian yang telah dilakukan sebelumnya, beberapa diantaranya sudah memanfaatkan sistem aplikasi *smart parking* dengan menggunakan mikrokontroler Arduino dan sensor ultrasonik. Berbeda dalam basis mikrokontroler dan jenis sensor, penelitian ini bertujuan untuk merancang dan membuat aplikasi sistem parkir terkontrol berbasis NodeMCU ESP8266 dan sensor *infrared* (IR) dengan penggunaan yang lebih praktis dimana penelitian ini menampilkan data jumlah kendaraan yang ada di tempat parkir serta lama waktu parkir pada MQTT Server Adafruit IO Platform. Dashboard MQTT Server ini lebih lanjut bisa ditampilkan pada sebuah layar lebar yang dapat diletakkan di depan gedung parkir sehingga mudah dilihat oleh pengguna.

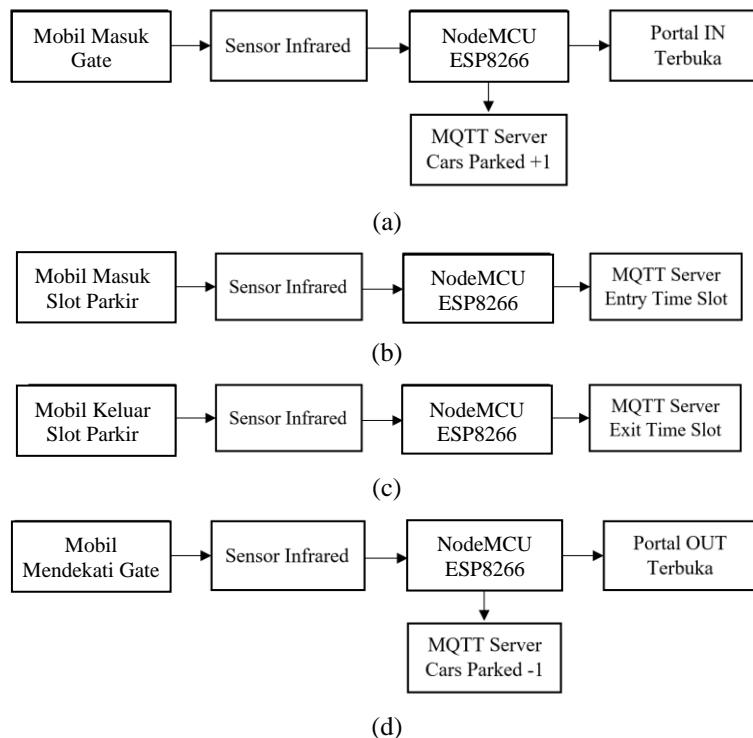
2.METODE PENELITIAN

2.1 Tahapan Penelitian

Tahapan penelitian dilakukan dengan perancangan blok diagram, pembuatan skematik diagram, pembuatan *flowchart*, realisasi prototipe sistem parkir, dan pengujian sistem.

2.2 Blok Diagram

Sistem parkir ini dirancang dengan proses pengontrolan dimulai dengan proses identifikasi, yaitu input berupa masukan dari sensor *infrared* yang akan dimasukkan pada mikrokontroler, kemudian akan menggerakkan motor servo pada portal IN dan portal OUT. Dalam sistem parkir ini terdapat empat proses yang masing-masing digambarkan dalam diagram blok Gambar 1 (a-d) berikut:



Gambar 1. Diagram blok sistem: (a) mobil masuk lahan parkir, (b) mobil masuk slot parkir, (c) mobil keluar slot parkir, (d) mobil keluar lahan parkir

Gambar 1 menampilkan diagram blok sistem yang dimulai saat mobil masuk lahan parkir, lalu menempati slot parkir, kemudian meninggalkan slot parkir tersebut, dan terakhir

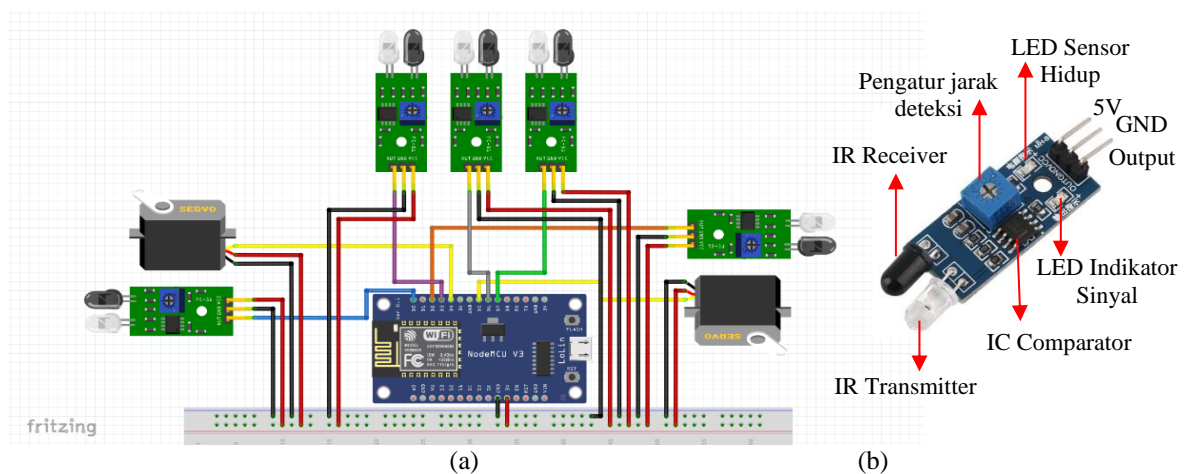
keluar dari lahan parkir. Adapun komponen utama yang bekerja dari keempat blok diatas adalah:

- Sensor *infrared* berfungsi memberikan *value* kepada mikrokontroler NodeMCU ESP8266 berupa *value high* dan *low*, dimana *high* yang berarti buka dan *low* yang berarti menutup gerbang, serta berfungsi untuk memberikan *value* jika terdapat kendaraan di slot parkir.
- NodeMCU ESP8266 berfungsi sebagai mikrokontroler yaitu sebagai pemroses dan pengendali utama pada sistem parkir. NodeMCU ini akan memerintahkan untuk membuka portal IN dan OUT serta mengirimkan data ke MQTT Server.
- Motor Servo berfungsi sebagai penggerak portal. Dengan masuknya input dari sensor *infrared* melalui NodeMCU ESP8266 maka motor servo disini melanjutkan perintah yang diberikan dari pengendali utama untuk membuka dan menutup portal.
- MQTT Server berfungsi untuk memonitoring slot parkir yang tersisa yang akan dimunculkan pada *dashboard* sekaligus juga mampu memberi penanda waktu saat masuk dan keluar.

2.3 Skematik Diagram

Pada perancangan perangkat keras, skematik diagram sistem merupakan salah satu bagian yang sangat penting. Rangkaian skematik terdiri dari beberapa jenis komponen yang dirangkai menjadi satu kesatuan untuk memudahkan saat diaplikasikan. Disini, NodeMCU ESP8266 sebagai pengendali dan pemroses data input sensor yang memiliki banyak pin dan output yang mampu memproses keluaran dari sensor untuk menggerakkan motor servo dengan sudut tertentu. Skematik diagram sistem dapat dilihat pada Gambar 2 (a).

Sedangkan sensor *infrared* mampu mendeteksi suatu benda yang menutupinya dengan *IR transmitter* dan *receiver*, Gambar 2 (b). Output dari sensor ini masuk ke mikrokontroler lalu dieksekusi dengan memberikan suatu nilai pada pin-pin ESP8266. Nilai inilah yang akan menggerakkan motor servo sehingga portal IN dan portal OUT mampu membuka dan menutup. Kemudian, nilai ini akan langsung terhubung pada MQTT Server sehingga mampu memonitoring lahan parkir dari baik dari laptop, smartphone maupun layar SmartTV.

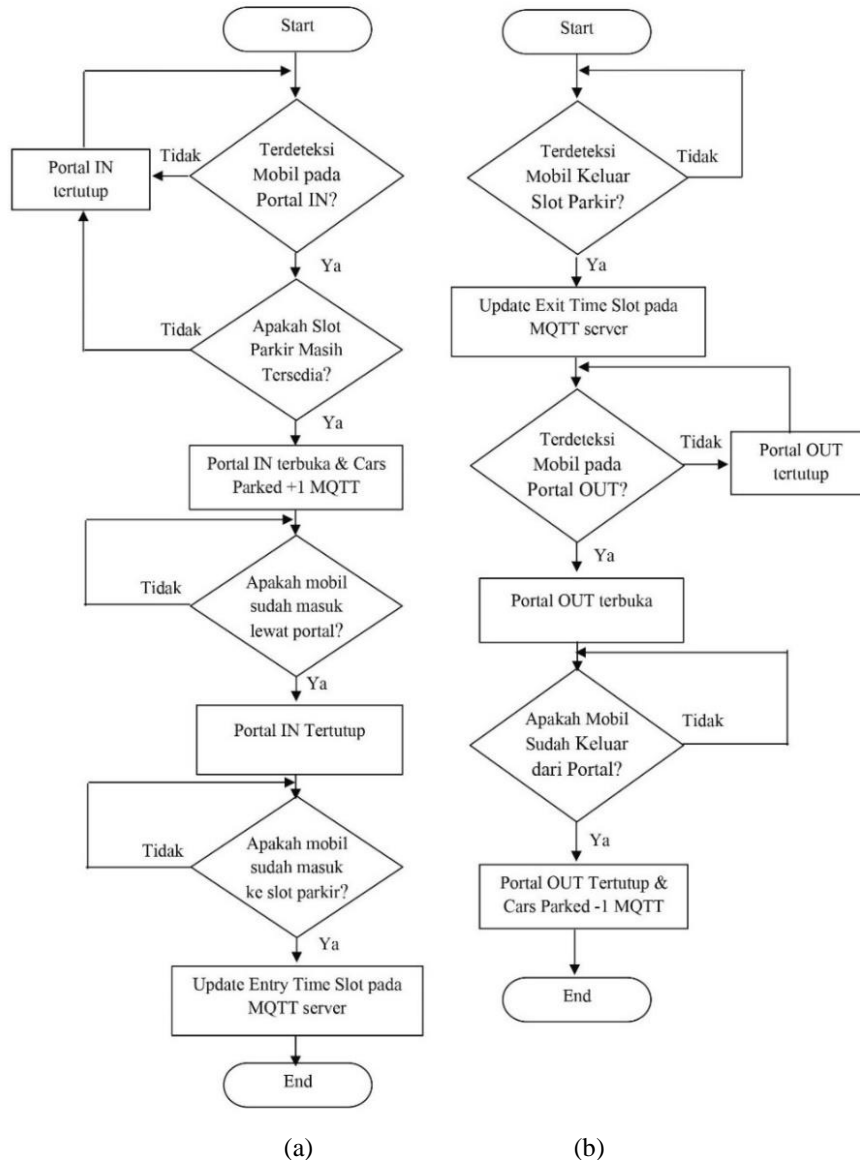


Gambar 2. (a) Skematik diagram, dan (b) sensor infrared sebagai komponen sensor utama

2.4 Flowchart

Pada perancangan ini ESP8266 adalah pengendali utama dan pemberi nilai pada pin-pinnya. Pembacaan data dimulai dari monitor MQTT Server yang menunjukkan slot parkir yang belum terisi. Untuk proses selanjutnya sensor *infrared* mendapat masukan dari pembacaan sensor yang membuat motor servo (portal IN) terbuka dan langsung menutup sendirinya. Sehingga selanjutnya pada MQTT Server langsung menyatakan bahwa pada lahan parkir itu terdapat satu kendaraan yang sudah terparkir. Begitu juga saat sensor *infrared* mendapat nilai

pada portal OUT. Portal akan terbuka dan langsung menutup dengan sendirinya setelah sekian detik. MQTT Server langsung mengupdate status kendaraan pada lahan parkir dengan mengurangi status satu kendaraan yang terparkir sesaat setelah portal OUT terbuka lalu menutup. Flowchart untuk portal IN dan portal OUT terlihat pada Gambar 3.



Gambar 3. (a) Flowchart pada portal IN, dan (b) Flowchart pada Portal OUT

3.HASIL DAN PEMBAHASAN

3.1 Realisasi Alat

Realisasi prototipe dari sistem parkir ini terlihat pada Gambar 4. Prototipe ini dirancang untuk mensimulasikan sistem parkir kendaraan roda empat dengan kapasitas 3 kendaraan. Alat ini dirancang dengan memiliki alur proses pengontrolan yang berpusat pada NodeMCU ESP8266. Pada pintu masuk, Gambar 4 (a), terdapat sensor *infrared* disebelahnya untuk mengidentifikasi apabila ada kendaraan mendekati sensor kira-kira 2-5 cm. Apabila sensor telah ON, maka akan menggerakkan motor servo untuk membuka portal dengan sudut 90° dan

diberikan *delay* beberapa detik untuk memastikan kendaraan telah lewat lalu portal menutup kembali secara otomatis. Pada setiap slot parkir, terdapat juga sensor *infrared* untuk memastikan apakah di slot parkir tersebut ada kendaraan yang terparkir atau tidak, Gambar 4 (b) dan (c). Sensor-sensor *infrared* ini akan memberikan data ke MQTT Server untuk mengetahui berapa slot parkir yang terisi dan yang kosong. Dari sini, waktu pendudukan dari masing-masing slot parkir akan didapat. Ketersediaan slot parkir akan menentukan apakah kendaraan yang mau masuk lahan parkir akan diijinkan atau tidak. Untuk pusat kontrol yang menggunakan NodeMCU ESP8266 adalah seperti yang ditunjukkan oleh Gambar 4 (d). Lebih lanjut, NodeMCU ESP8266 akan memakai WiFi untuk melakukan transfer data ke MQTT Server yang dalam hal ini akan ditampilkan di layar monitor. Untuk pintu keluar, sistemnya sama seperti pintu masuk dimana terdapat sebuah portal OUT yang dilengkapi oleh sensor *infrared* dan akan menggerakkan motor servo jika terdeteksi kendaraan yang akan keluar. Dengan rancangan prototipe seperti ini diharapkan dapat dilakukan simulasi untuk mewakili sistem parkir gedung yang sebenarnya.



Gambar 4. Prototipe sistem parkir otomatis: (a) Bagian portal masuk, (b) Slot parkir, (c) Sensor infrared di masing-masing slot parkir, (d) Bagian pusat kontrol.

3.2 Koding Program dan MQTT Server

Untuk dapat mengoperasikan sistem parkir ini dibuat sebuah program melalui Arduino IDE dengan menggunakan bahasa C/C++ yang dinamakan *sketch* berdasarkan alur logika *flowchart* pada Gambar 3 diatas. Arduino IDE ini bersifat *open source* yang berguna untuk membuat, mengedit, dan kemudian mengupload program ke *board* NodeMCU setelah dilakukan *compile* menggunakan perintah *verify* atau *compile*. Contoh potongan *coding* dari sistem parkir ini ditunjukkan pada Gambar 5 yang menampilkan bagian inisialisasi dan *void setup()*. Pada bagian inisialisasi terdapat pemanggilan beberapa *libraries* yang diperlukan seperti untuk

identifikasi ESP8266, servo, NTPClient, serta Wifi. Sedangkan library Adafruit_MQTT.h dan juga Adafruit_MQTT_Client.h digunakan agar sistem ini bisa terkoneksi dengan MQTT Server. Parameter MQTT_SERV diarahkan ke io.adafruit.com. MQTT (*Message Queuing Telemetry Transport*) sendiri adalah salah satu protokol IoT yang digunakan untuk komunikasi yang berjalan diatas *stack* TCP/IP [17]. Di lain pihak, *void setup()* merupakan fungsi yang dipanggil ketika program mulai dijalankan dan hanya akan berjalan sekali saja ketika *board* NodeMCU *reset* atau *PowerUp*. Dalam hal ini, *void setup()* digunakan untuk mendeklarasikan variabel, mode pin yang digunakan baik INPUT maupun OUTPUT serta memulai penggunaan *libraries*.

<pre> project_smartparking \$ #include <ESP8266WiFi.h> #include <Servo.h> #include <NTPClient.h> #include <WiFiUdp.h> #include <NTPClient.h>; #include <WiFiUdp.h> #include "Adafruit_MQTT.h" #include "Adafruit_MQTT_Client.h" const char *ssid = "OMING"; const char *pass = "GENTONGwater"; #define MQTT_SERV "io.adafruit.com" #define MQTT_PORT 1883 #define MQTT_NAME "putrasuardana6" #define MQTT_PASS "aio_tmiH49YEtFWIO14NH6CW5bwckSh1" WiFiUDP ntpUDP; NTPClient timeClient(ntpUDP, "pool.ntp.org", 19800,60000); Servo servoIn; Servo servoOut; int carEnter = D0; int carExited = D2; int slot3 = D7; int slot2 = D6; int slot1 = D3; int count =0; int CLOSE_ANGLE = 80; int OPEN_ANGLE = 0; </pre>	<pre> project_smartparking \$ void setup() { delay(1000); Serial.begin (9600); mqtt.subscribe(&EntryGate); mqtt.subscribe(&ExitGate); timeClient.begin(); timeClient.setTimeOffset(28800); servoIn.attach(D4); servoOut.attach(D5); pinMode(carExited, INPUT); pinMode(carEnter, INPUT); pinMode(slot1, INPUT); pinMode(slot2, INPUT); pinMode(slot3, INPUT); WiFi.begin(ssid, pass); Serial.print("Connecting to "); Serial.print(ssid); while (WiFi.status() != WL_CONNECTED) { Serial.print("."); delay(500);} Serial.println(); Serial.print("Connected to "); Serial.println(ssid); Serial.print("IP Address is : "); Serial.println(WiFi.localIP()); } </pre>
(a)	(b)

Gambar 5. Potongan koding bagian: (a) Inisialisasi dan (b) Void Setup()

3.3 Komponen Sistem

Sistem parkir terkontrol ini terdiri dari dua komponen yang saling terhubung satu sama lain yaitu komponen pengendali sistem dan komponen manajemen parkir. Komponen pengendali secara keseluruhan yang meliputi NodeMCU ESP8266, motor servo, serta sensor *infrared*. Prototipe yang dibuat menggunakan 5 sensor *infrared*, 2 motor servo, dan 1 NodeMCU ESP8266. Dua sensor *infrared* diletakkan di masing-masing portal IN dan OUT dimana sensor *infrared* ini akan dijadikan *trigger* untuk membuka atau menutup portal jika terdeteksi kendaraan yang akan masuk maupun yang keluar area parkir. Disini sensor slot parkir yang diuji adalah 3 buah. Adapun pengolahan datanya digunakan MQTT Server Adafruit IO Platform untuk pengiriman data kontrol dan monitoring slot parkir yang tersedia pada prototipe dibuat. Komponen manajemen parkir merupakan hasil dari komponen pengendali yang telah bekerja sesuai dengan rancangan. Manajemen parkir ditujukan untuk mengetahui berapa slot parkir yang terisi dan juga tersisa. Sedangkan informasi parkir yang tampil pada aplikasi Adafruit IO Platform ialah hasil dari manajemen tersebut.

3.4 Pengujian Sistem

Pengujian sistem meliputi pengujian portal, keterisian slot, dan update data waktu pendudukan slot parkir. Tujuan utama dari pengujian sistem ini adalah untuk memastikan komponen-komponen yang terpasang pada alat mampu berjalan dan berfungsi dengan baik demikian juga dengan tampilan *dashboard*nya. Pengujian sensor ini juga bertujuan untuk

mengetahui baik atau buruknya pengaturan sensitifitas pada sensor *infrared*. Sensor ini terdapat di beberapa titik, diantaranya yaitu masing-masing pada portal IN, portal OUT, dan pada slot parkir nomor 1, slot parkir nomor 2, slot parkir nomor 3 dan seterusnya. Mekanisme pengujian sistem adalah sebagai berikut:

- a. *Sensor Infrared*
Dari pengujian sensor *infrared* didapat tegangan output sebesar 2,2 volt ketika sensor tidak terhalang oleh kendaraan dan jika sensor terhalang oleh kendaraan yang melewati sensor tersebut maka memiliki tegangan output sebesar 3,8 volt.
- b. Portal IN
Pengujian pada portal in dimaksudkan untuk mengetahui ada tidaknya pergerakan mobil pada portal ini, jika sensor membaca terdapat kendaraan yang menghalangi sensor ini, maka otomatis portal IN ini akan membuka dan menutup sendirinya.
- c. Portal OUT
Pengujian pada portal OUT dimaksudkan untuk mengetahui ada tidaknya pergerakan mobil pada portal ini, jika sensor membaca terdapat kendaraan yang menghalangi sensor ini, maka otomatis portal OUT ini akan membuka dan menutup sendirinya.
- d. Slot Parkir Satu
Pengujian pada slot parkir satu bertujuan untuk mengetahui pada slot parkir ini apakah sudah terisi atau belum. Untuk mengetahui pengujian ini berhasil atau belum dapat dilihat dari 2 lampu indikator yang menyala, jika hanya terdapat 1 lampu yang menyala, berarti slot parkir ini belum digunakan.
- e. Slot Parkir Dua
Pengujian pada slot parkir dua bertujuan untuk mengetahui pada slot parkir ini apakah sudah terisi atau belum. Untuk mengetahui pengujian ini berhasil atau belum dapat dilihat dari 2 lampu indikator yang menyala, jika hanya terdapat 1 lampu yang menyala, berarti slot parkir ini belum digunakan.
- f. Slot Parkir Tiga
Pengujian pada slot parkir tiga bertujuan untuk mengetahui pada slot parkir ini apakah sudah terisi atau belum. Untuk mengetahui pengujian ini berhasil atau belum dapat dilihat dari 2 lampu indikator yang menyala, jika hanya terdapat 1 lampu yang menyala, berarti slot parkir ini belum digunakan.
- g. Update Data
Pengujian ini untuk mengetahui apakah MQTT Server mampu memonitor saat terdapat kendaraan yang masuk atau keluar dari portal sehingga dapat menyimpan data pada server.
- h. Interaksi Web
Pengujian ini untuk mengetahui berapa jumlah kendaraan yang terparkir pada lahan parkir. Pada pengujian ini juga harus memperlihatkan jam masuk dan keluar kendaraan.

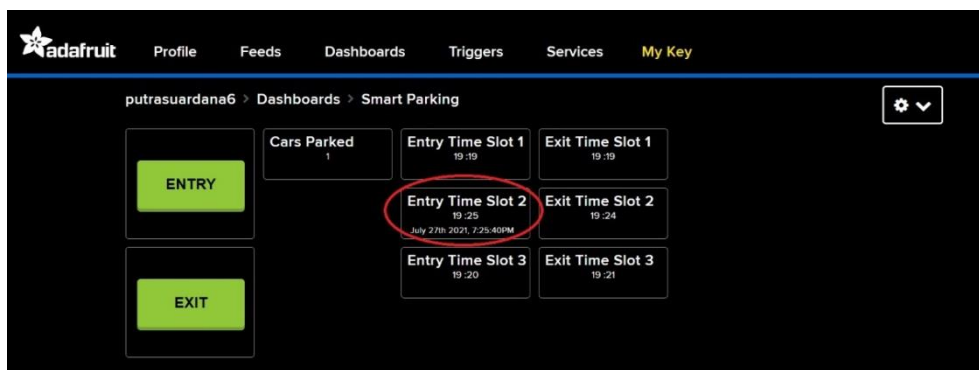
Hasil pengujian kelima sensor *infrared* dapat dilihat pada Tabel 1. Terlihat bahwa jika sensor mendapatkan *value* maka kedua lampu indikatornya akan menyala. Pada portal hal tersebut menandakan ada kendaraan yang melintas, sedangkan pada slot parkir menandakan ada kendaraan yang terparkir. Apabila semua hal yang menyangkut penerapan sistem telah berjalan dengan baik dan benar maka sistem tersebut dapat dioperasikan. Untuk lama parkir dihitung dari waktu kendaraan masuk dikurangi waktu kendaraan keluar, Tabel 2. Selanjutnya ditampilkan data kondisi pada saat terdapat 1, 2, dan 3 mobil di gedung parkir.

Tabel 1. Hasil Pengujian Sensor

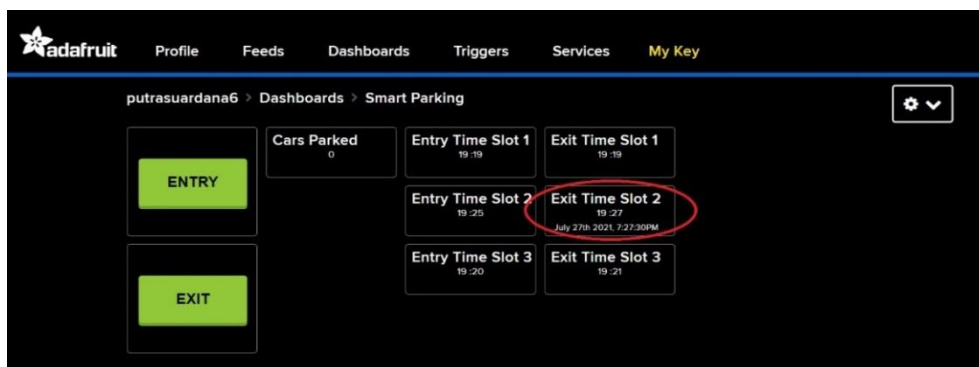
No.	Sensor	Indikator sensor	Keterangan
1	Sensor Infrared pada portal IN	2 lampu indikator menyala yang menandakan sensor mendapat value	Kondisi pada portal IN terdapat kendaraan melintas
2	Sensor Infrared pada portal OUT	2 lampu indikator menyala yang menandakan sensor mendapat value	Kondisi pada portal OUT terdapat kendaraan melintas
3	Sensor Infrared pada Slot 1	2 lampu indikator menyala yang menandakan sensor mendapat value	Kondisi pada slot 1 terdapat kendaraan yang terparkir
4	Sensor Infrared pada Slot 2	2 lampu indikator menyala yang menandakan sensor mendapat value	Kondisi pada slot 2 terdapat kendaraan yang terparkir
5	Sensor Infrared pada Slot 3	2 lampu indikator menyala yang menandakan sensor mendapat value	Kondisi pada slot 3 terdapat kendaraan yang terparkir

Tabel 2. Kondisi Pada Saat 1 Mobil didalam Gedung

No.	Slot Parkir	Waktu Mobil Masuk	Waktu Mobil Keluar	Lama parkir
1.	1	-	-	-
2.	2	11.25	11.27	2 menit
3.	3	-	-	-



Gambar 6. Entry time pada MQTT Server saat kondisi 1 mobil yang menggunakan slot

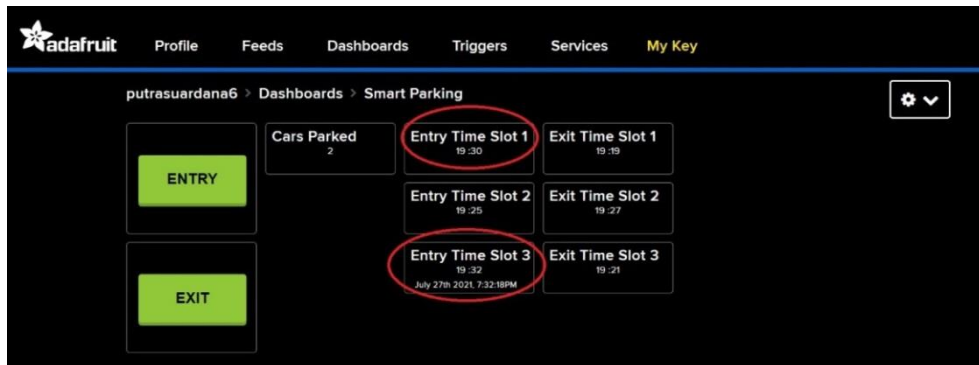


Gambar 7. Exit time pada MQTT Server saat kondisi mobil yang menggunakan slot keluar

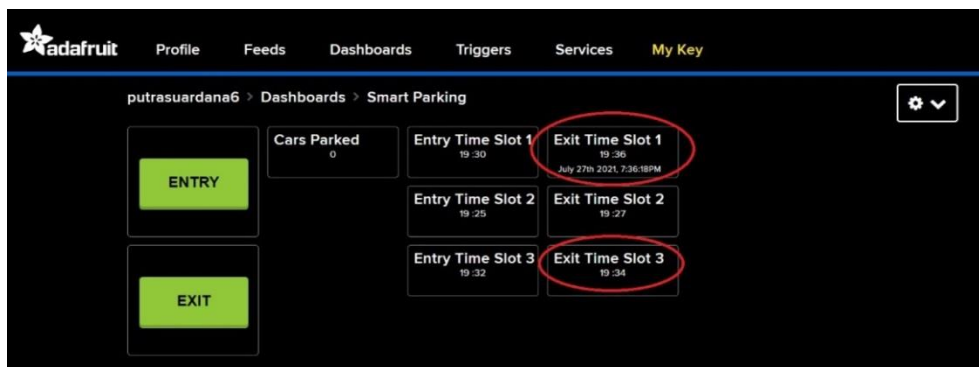
Dari Gambar 6 dan 7 dapat dilihat bahwa waktu yang pengguna pakai untuk parkir akan terupdate secara otomatis saat pengguna sudah keluar dari lahan parkir. Seperti pada slot 2, pada Gambar 6 diketahui pengguna masuk pada jam 19.25 namun pada kolom exit tertulis jam 19.24. Hal ini disebabkan pengguna belum keluar dari lahan parkir, dimana belum dapat dikalkulasi berapa waktu yang digunakan pengguna untuk parkir. Sehingga waktu yang diperlihatkan yaitu waktu pengendara sebelumnya keluar dari lahan parkir. Sedangkan pada Gambar 7 ketika mobil sudah keluar dari lahan parkir, sensor akan otomatis mencatat berapa lama waktu yang digunakan oleh pengguna. Data tersebut langsung diperlihatkan melalui server.

Tabel 3. Kondisi Pada Saat 2 Mobil didalam Gedung

No.	Slot Parkir	Waktu Mobil Masuk	Waktu Mobil Keluar	Lama Parkir
1.	1	19.30	19.36	6 Menit
2.	2	-	-	-
3.	3	19.32	19.34	2 Menit



Gambar 8. Entry time pada MQTT Server saat kondisi 2 mobil yang menggunakan slot

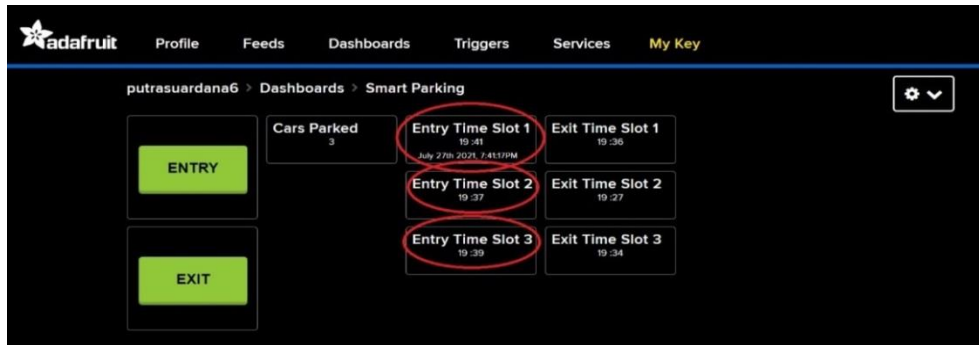


Gambar 9. Exit time pada MQTT Server saat kondisi 2 mobil yang menggunakan slot

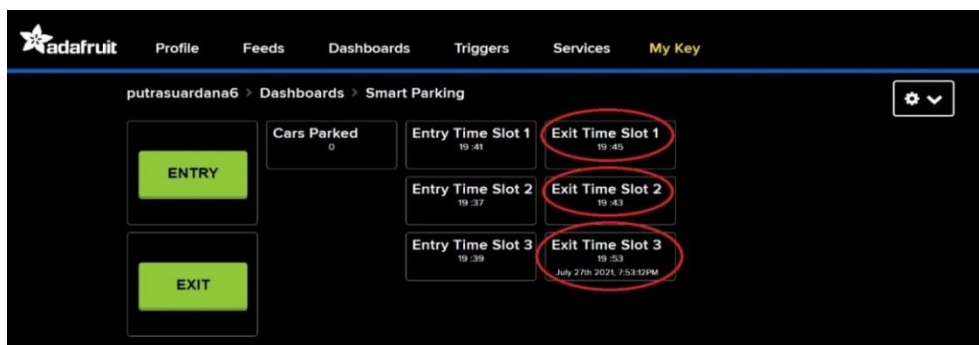
Selanjutnya dari Gambar 8 dan 9 dapat dilihat bahwa waktu yang pengguna gunakan untuk parkir akan terupdate secara otomatis saat pengguna sudah keluar dari area parkir. Seperti pada slot 1 dan 3, pada Gambar 8 diketahui pengguna masuk pada jam 19.30 dan 19.32 namun pada kolom exit tertulis jam 19.19 dan 19.21. Hal ini disebabkan pengguna belum keluar dari area parkir, dimana belum dapat dikalkulasi berapa waktu yang digunakan pengguna untuk parkir. Sehingga waktu yang diperlihatkan yaitu waktu pengendara sebelumnya keluar dari area parkir. Sedangkan pada Gambar 9 ketika mobil sudah keluar dari area parkir, sensor akan otomatis mencatat berapa lama waktu yang digunakan oleh pengguna. Data tersebut langsung diperlihatkan melalui server. Kemudian, lama waktu ini dapat digunakan lebih lanjut untuk penghitungan tarif parkir.

Tabel 4. Kondisi Pada Saat 3 Mobil didalam Gedung

No.	Slot Parkir	Waktu Mobil Masuk	Waktu Mobil Keluar	Lama Parkir
1.	1	19.41	19.45	4 Menit
2.	2	19.37	19.43	6 Menit
3.	3	19.39	19.53	14 Menit



Gambar 10. *Entry time* pada MQTT Server saat kondisi 3 mobil yang menggunakan slot



Gambar 11. *Exit time* pada MQTT Server saat kondisi 3 mobil yang menggunakan slot

Dari Gambar 10 dan 11 dapat dilihat bahwa waktu yang pengguna gunakan untuk parkir akan terupdate secara otomatis saat pengguna sudah keluar dari area parkir. Seperti pada slot 1, 2 dan 3, pada Gambar 10 diketahui pengguna masuk pada jam 19.41, 19.37, dan 19.39 namun pada kolom *exit* tertulis jam 19.36, 19.27, dan 19.34. Hal ini disebabkan pengguna belum keluar dari area parkir, dimana belum dapat dikalkulasi berapa waktu yang digunakan pengguna untuk parkir. Sehingga waktu yang diperlihatkan yaitu waktu pengendara sebelumnya keluar dari lahan parkir. Sedangkan pada Gambar 11 ketika mobil sudah keluar dari lahan parkir, sensor akan otomatis mencatat berapa lama waktu yang digunakan oleh pengguna. Data tersebut langsung diperlihatkan melalui server. Pengujian server MQTT disini untuk mengetahui apakah server sudah terhubung pada sistem parkir terkontrol ini, sehingga lahan parkir yang tersisa dapat dimonitor dan juga diketahui lama waktu yang digunakan pengguna untuk parkir. Dalam pengujian secara keseluruhan yang sudah dilakukan, prototipe alat ini mampu bekerja dengan baik dengan sistem *live dashboard*. Web MQTT mampu memberikan *value* jika slot parkir terisi kendaraan serta waktu masuk dan keluar kendaraan tersebut dari slot parkir.

4.KESIMPULAN DAN SARAN

Prototipe sistem parkir berbasis *live dashboard* ini terdiri dari dua komponen yang saling terhubung satu sama lain yaitu komponen pengendali sistem dan komponen manajemen parkir. Komponen pengendali sistem terdiri dari NodeMCU ESP8266 sebagai pusat kontrol, sensor *infrared* sebagai piranti akuisisi data, dan motor servo sebagai aktuator portal masuk maupun keluar. Sedangkan komponen manajemen parkir berupa MQTT Server Adafruit IO Platform sebagai pengolah data dan *display* dalam bentuk *live dashboard*. Hasil pengujian menunjukkan bahwa prototipe sistem parkir ini dapat memberikan informasi keterisian slot parkir dan waktu kendaraan yang masuk serta keluar gedung parkir selalu terupdate secara real-time menggunakan waktu GMT+8. Sehingga nantinya untuk perhitungan biaya parkir menjadi lebih mudah, karena sudah ada jangka waktu penggunaan parkir dari masing masing slot parkir yang terpakai. Selain itu, kapasitas parkir yang terpakai di dalam gedung juga terlihat pada

MQTT Server. Hal ini ditujukan untuk mempermudah melihat apakah masih ada slot parkir yang tersedia. Lebih lanjut, rencana pengembangan dari penelitian ini adalah penambahan sistem *reserve* and *pay* dimana pengguna dapat memesan dan membayar slot parkir untuk jangka waktu tertentu sebelum pengguna tiba di lokasi parkir. Dengan sistem ini pengguna akan mendapat jaminan mendapatkan slot parkir.

DAFTAR PUSTAKA

- [1] Fariyono, F. T. Wulandari, and A. R. Bahtiar, "Penerapan Internet of Thing pada Aplikasi Mobile Smart Lot Parking," *J. Media Inform. Budidarma*, vol. 6, no. 2, pp. 1051–1055, 2022, doi: 10.30865/mib.v6i2.3779.
- [2] S. Ariyani, H. Setyawan, and D. A. Dimas, "Prototipe Sistem Parkir Bergerak Berbasis IoT Menggunakan Rasperry Pi," *J. Tek. Elektro dan Komputasi*, vol. 2, no. 2, pp. 96–111, 2020, doi: 10.32528/elkom.v2i2.3438.
- [3] D. Susandi, W. Nugraha, and S. F. Rodiyansyah, "Perancangan Smart Parking System Pada Prototipe Smart Office Berbasis Internet Of Things," in *Prosiding Semnastek*, 2017, no. November, pp. 1–7.
- [4] Y. A. Ristanti and L. Sudarmana, "Sistem Informasi Pengelolaan Tempat Parkir RSUD Kertosono Nganjuk Berbasis Web," *J. I. Sunan Kalijaga*, vol. 1, no. 1, pp. 22–28, 2016.
- [5] R. Kurniawan and A. Zulius, "Sistem Smart Parking Menggunakan Ultrasonik Sensor," *J. Sist. Komput. Musirawas*, vol. 3, no. 1, p. 22, 2018, doi: 10.32767/jusikom.v3i1.309.
- [6] T. U. Anastasia, A. Mufti, and A. Rahman, "Rancang Bangun Sistem Parkir Otomatis Dan Informatif Berbasis Mikrokontroler Atmega2560," *Kitektro*, vol. 2, no. 1, pp. 29–34, 2017.
- [7] D. S. Nugroho *et al.*, "Pengembangan Smart Parking System Menggunakan Intensitas Cahaya," in *Seminar Nasional Teknik Industri UGM*, 2016, pp. 228–233.
- [8] A. Adwindea, A. Sofwan, and M. Arfan, "Perancangan Aplikasi Antarmuka Smart Open Parking Berbasis Internet Of Things (IoT) Pada Perangkat Android," *Transient*, vol. 7, no. 3, pp. 803–810, 2019, doi: 10.14710/transient.7.3.803-810.
- [9] M. Akbar and S. Jura, "Sistem Informasi Realtime Web Untuk Slot Parkir Berbasis Embedded System," *J. IKRA-ITH Inform.*, vol. 3, no. 2, pp. 33–38, 2019.
- [10] Agustiawan, B. Hendrawan, and M. A. H. Sirad, "Rancang Bangun Sistem Parkir Dengan Menggunakan Fingerprint," *Patria Artha Technol. J.*, vol. 4, no. 2, pp. 113–116, 2020, doi: 10.33857/patj.v4i2.358.
- [11] P. S. Frima Yudha and R. A. Sani, "Implementasi Sensor Ultrasonik HC-SR04 Sebagai Sensor Parkir Mobil Berbasis Arduino," *J. Einstein*, vol. 5, no. 3, pp. 19–26, 2017, doi: 10.24114/einstein.v5i3.12002.
- [12] D. Ichwana, R. Aisuwarya, S. Ardopa, and I. Purnama, "Sistem Cerdas Reservasi dan Pemantauan Parkir pada Lokasi Kampus Berbasis Konsep Internet of Things," *J. Teknol. dan Sist. Komput.*, vol. 6, no. 2, pp. 57–63, 2018, doi: 10.14710/jtsiskom.6.2.2018.57-63.
- [13] R. Aminuddin, M. Rais, and M. A. H. Sirad, "Implementasi Sistem Pengontrolan Smart Parking Menggunakan Mikrokontroler Arduino Uno Di Universitas Patria Artha," *Patria Artha Technol. J.*, vol. 2, no. 2, pp. 123–132, 2018, doi: 10.33857/patj.v2i2.179.
- [14] M. Yamin, "Prototipe Sensor Parkir Mobil Berbasis Mikro AT89S51," *UNS*, 2009.
- [15] A. Fernando Sianipar and N. Lutfiani, "Sistem Parkir Berbasis QR Code Pada Perguruan Tinggi Rahaarja," *S. Nas. Multi Disiplin Ilmu*, vol. 1, no. Nov 2017, pp. 43–54, 2017.
- [16] A. G. Sanjaya, "Prototipe Sistem Informasi Kendali Parkir Berbasis Web dengan Informasi Ruang Parkir Tersedia Menggunakan Tampilan 7 Segment," *Amikom*, 2012.
- [17] G. Y. Saputra, A. D. Afrizal, F. K. R. Mahfud, F. A. Pribadi, and F. J. Pamungkas, "Penerapan Protokol MQTT Pada Teknologi Wan (Studi Kasus Sistem Parkir Univeristas Brawijaya)," *J. Inform. Mulawarman*, vol. 12, no. 2, pp. 69–75, 2017.