

Sistem Pemberi Informasi dan Visualisasi Ketersediaan Parkir pada Lahan Parkir Bertingkat Berbasis IoT Menggunakan ESP32

IoT-Based System for Providing Information and Visualizing Parking Availability in Multi-Storey Parking Lots Using ESP32

Yohana Susanthi^{1*}, Agus Prijono², Patrick Felipe Army³

^{1,2,3}Teknik Elektro, Universitas Kristen Maranatha

Jln. Prof. drg. Surya Sumantri no.65 Bandung, telp. (022)-2012186
yohana.susanthi@eng.maranatha.edu^{1*}, agus.prijono@eng.maranatha.edu²,
1822022@eng.maranatha.edu³

Abstrak – Seiring dengan semakin meningkatnya jumlah kendaraan bermotor khususnya mobil akan berdampak pada keterbatasan jumlah slot parkir yang tersedia. Salah satu upaya untuk dapat menampung sebanyak mungkin kendaraan bermotor adalah menggunakan lahan parkir yang bertingkat. Namun seringkali pengunjung tidak mendapatkan slot parkir dan harus berkeliling beberapa kali di setiap lantai parkir untuk mencari slot parkir yang kosong sehingga sangat tidak efisien dan banyak membuang waktu. Dengan menerapkan teknologi Internet of Things (IoT) memungkinkan segala hal terhubung dengan internet, dipantau dan dilakukan dari smartphone. Oleh karenanya penelitian ini bertujuan untuk membantu mengatasi masalah parkir tersebut dengan menampilkan visualisasi ketersediaan slot parkir di setiap lantai parkir kepada pengunjung pada layar display, smartphone pengunjung dan lampu indikator. Sistem yang dibuat menggunakan sensor ultrasonik HC-SR04 dan pengendali mikro ESP32 untuk mendeteksi slot parkir yang kosong. Data dari sensor ultrasonik dikirimkan oleh pengendali mikro ESP32 secara nirkabel ke server yang berada di cloud. Sistem berhasil menampilkan visualisasi ketersediaan slot parkir di setiap lantai parkir dalam bentuk gambar denah parkir pada layar display dan smartphone pengunjung. Slot parkir yang sudah terisi mobil akan berwarna merah, sedangkan slot parkir yang masih kosong berwarna hijau. Warna slot parkir pada denah parkir dan warna lampu indikator pada slot parkir dapat berubah sesuai dengan kondisi slot parkir secara real time.

Kata Kunci: slot parkir, sensor ultrasonik, ESP32

Abstract – Along with the increasing number of motorized vehicles, especially cars, it will have an impact on the limited number of available parking slots. One of the efforts to be able to accommodate as many motorized vehicles as possible is to use multi-storey parking lots. However, visitors often do not get a parking slot and have to go around several times on each parking floor to find an empty parking slot, which is very inefficient and a waste of time. By implementing Internet of Things (IoT) technology, it is possible for everything to be connected to the internet, monitored and controlled from a smartphone. Therefore, this study aims to help overcome this problem by visualizing the availability of parking slots on each parking

TELKA, Vol.10, No.2, Juli 2024, pp. 144~157

ISSN (e): 2540-9123

ISSN (p): 2502-1982

■ 144

floor to visitors on the display screen, smartphone and indicator lights. The system designed uses the HC-SR04 ultrasonic sensor and the ESP32 microcontroller to detect empty parking slots. Data from the ultrasonic sensor is sent by the microcontroller wirelessly to a server in the cloud. The system successfully displays a visualization of available parking slots on each parking floor in the form of a parking plan image on the display screen and visitor's smartphone. Parking slots that are already filled with cars will be red, while empty parking slots will be green. The color of the parking slots on the parking plan and the color of the indicator lights on the parking slots can change according to the conditions of the parking slots in real time.

Keywords: *parking slots, ultrasonic sensor, ESP32*

1. Pendahuluan

Setiap tahun terjadi peningkatan jumlah kendaraan bermotor. Pertumbuhan ekonomi memberikan peluang yang lebih besar bagi masyarakat untuk memiliki kendaraan bermotor [1]. Dengan peningkatan jumlah kendaraan bermotor, muncul masalah mengenai jumlah slot parkir yang tersedia sehingga salah satu upaya untuk dapat menampung sebanyak mungkin kendaraan bermotor adalah menggunakan lahan parkir yang bertingkat [2]. Dengan adanya lahan parkir bertingkat pengunjung masih mengalami kesulitan saat menemukan slot parkir yang kosong karena pengunjung tidak dapat melihat semua lantai parkir secara bersamaan. Masalah yang ditimbulkan dalam sistem perparkiran adalah kurangnya informasi mengenai slot parkir yang kosong.

Pada penelitian [3] dirancang sebuah model sistem pemantauan perparkiran dengan fasilitas pemilihan slot parkir dengan menekan tombol. Model miniatur lahan parkir dengan jumlah slot parkir sebanyak 20 buah dibuat untuk simulasi pengujian sistem. RFID (*Radio Frequency Identification*) digunakan untuk identifikasi pengunjung dan proses pengambilan data secara kontinu menggunakan Atmega16. Sistem perparkiran dengan menggunakan RFID juga dikembangkan dalam penelitian [4]. Peneliti lainnya [5] mengembangkan sistem pemantauan perparkiran dengan fasilitas pemilihan slot parkir berbasis Raspberry Pi. Pengujian sistem dilakukan secara simulasi menggunakan model miniatur perparkiran mobil 2 lantai dengan ukuran panjang 50 cm dan lebar 30 cm dengan jumlah total slot parkir sebanyak 12 buah. Pengunjung harus memilih slot parkir kosong yang ditampilkan pada layar *display* dan menekan tombol yang sesuai. Setelah pengguna memarkirkan kendaraan di slot parkir yang telah dipilih, sensor LDR (*Light Dependent Resistor*) akan mendeteksi bahwa slot parkir tersebut terisi mobil dan pada layar *display* slot parkir tersebut akan berwarna merah. Peneliti [6] mengembangkan sistem perparkiran cerdas menggunakan kamera CCTV (*Closed-circuit television*) dan sistem pemrosesan citra untuk mendeteksi slot parkir yang kosong pada lahan parkir yang terbuka. Setelah citra atau gambar slot parkir kosong berhasil dideteksi maka pengunjung yang hendak masuk ke lahan parkir diberikan kode QR (*Quick-Response code*) yang berisi tautan untuk mengakses halaman *web* yang memuat tampilan citra slot parkir kosong itu dari *smartphone* pengunjung. Pengunjung yang telah mengetahui area slot parkir kosong dapat segera menuju ke slot parkir tersebut. Selanjutnya penggunaan sensor ultrasonik dan teknologi *Internet of Things* juga telah dikembangkan untuk memberikan informasi posisi slot parkir yang kosong kepada pengguna parkir [7].

Untuk mengatasi masalah perparkiran maka penelitian ini bertujuan untuk menampilkan visualisasi ketersediaan slot parkir kosong di setiap lantai parkir kepada pengunjung secara real time pada layar *display* dan *smartphone* pengunjung. Sistem yang dibuat menggunakan sensor ultrasonik HC-SR04 dan pengendali mikro ESP32 untuk mendeteksi slot parkir yang kosong. Data dari sensor ultrasonik dikirimkan oleh pengendali mikro ESP32 secara nirkabel menggunakan protokol MQTT (*Message Queuing Telemetry Transport*) ke *server*. Protokol MQTT telah didesain agar menjadi protokol yang ringan dan membutuhkan *bandwidth* yang rendah [8]. Apabila dibandingkan dengan protokol HTTP (*Hypertext Transfer Protocol*), protokol MQTT memiliki *packet size* yang lebih kecil sehingga protokol MQTT lebih unggul dalam transmisi data [9]. Selanjutnya data yang diterima oleh *server* akan diolah menggunakan program Node-RED dan ditampilkan pada halaman *web* berupa gambar denah setiap lantai lahan parkir.

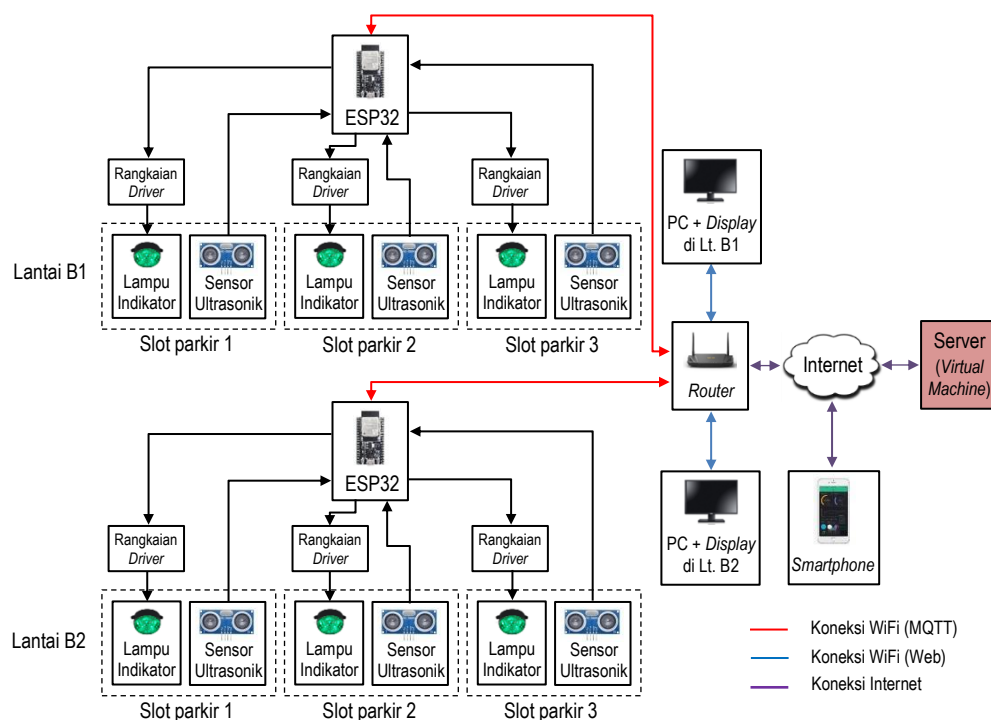
Slot parkir yang sudah terisi mobil akan berwarna merah, sedangkan slot parkir yang masih kosong berwarna hijau. Demikian pula lampu indikator yang terpasang pada bagian atas setiap slot parkir akan menyala merah apabila slot parkir itu terisi mobil dan menyala hijau jika kosong.

2. Metode Penelitian

Metode penelitian dalam penelitian ini terdiri dari beberapa tahapan yaitu perancangan diagram blok sistem, merealisasi perangkat keras dan perangkat lunak, menguji sistem dan mengevaluasi kinerja sistem. Sistem hanya diimplementasikan di 2 lantai parkir saja dan masing-masing lantai hanya 3 slot parkir tetapi sistem dapat dikembangkan untuk jumlah lantai parkir dan jumlah slot parkir yang lebih banyak.

2.1. Diagram Blok Sistem

Diagram blok sistem dapat dilihat pada Gambar 1.



Gambar 1. Diagram blok sistem.

Sensor ultrasonik HC-SR04 digunakan untuk mendeteksi slot parkir yang kosong. Setiap 1 slot parkir dipasang 1 buah sensor ultrasonik HC-SR04 dan 1 buah lampu indikator. Kemudian setiap 3 buah sensor ultrasonik HC-SR04 dan 3 buah lampu indikator dari 3 slot parkir yang bersebelahan dihubungkan dengan 1 pengendali mikro ESP32. Dalam uji coba, sistem direalisasi pada 3 slot parkir di lantai *basement* B1 dan B2 di sebuah gedung parkir. Data dari sensor ultrasonik kemudian dikirimkan oleh pengendali mikro ESP32 menggunakan protokol MQTT secara nirkabel melalui koneksi WiFi dan *router* ke *server* yang berada di internet. Sebagai MQTT *broker* dipilih Mosquitto yang merupakan salah satu MQTT *broker open source* yang cukup umum digunakan karena ringan dan tersedia pada berbagai *platform*. Kemudian *server* yang digunakan adalah Ubuntu Server 20.04 LTS berupa *virtual machine* (VM) di layanan *cloud computing* Microsoft Azure. Kinerja *webservice* pada *platform* Microsoft Azure telah teruji menggunakan metode Benchmarking dan lebih unggul dibandingkan *platform* lain [10]. Data yang diterima oleh *server* diproses menggunakan program Node-RED untuk menampilkan gambar denah lahan parkir lantai parkir B1 dan B2 pada halaman *web* dan ditampilkan pada sebuah layar *display* yang dipasang di jalur masuk lantai parkir B1 dan B2. Di dalam tampilan tersebut slot parkir yang sudah terisi mobil akan berwarna merah, sedangkan slot parkir yang

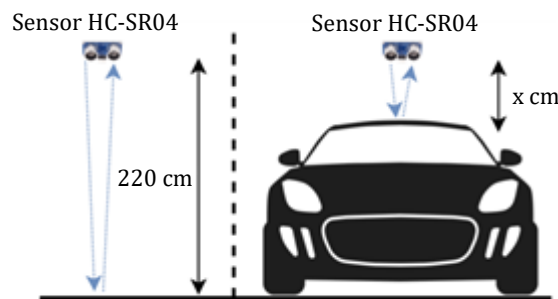
masih kosong berwarna hijau. Pengendali mikro ESP32 juga akan menyalakan lampu indikator yang terpasang pada bagian atas setiap slot parkir dengan warna merah jika slot parkir tersebut terisi mobil atau dengan warna hijau jika slot parkir kosong. Lampu indikator ini sebagai tambahan untuk lebih memudahkan pengunjung saat mencari slot parkir kosong pada lantai tersebut. Pengunjung juga dapat menggunakan *smartphone* mereka untuk mengakses tampilan denah parkir tersebut melalui internet dari aplikasi peramban *web*.

2.2. Perancangan Perangkat Keras

Perancangan perangkat keras meliputi pemasangan sensor ultrasonik HC-SR04 pada bagian atas slot parkir dan perancangan rangkaian elektronika.

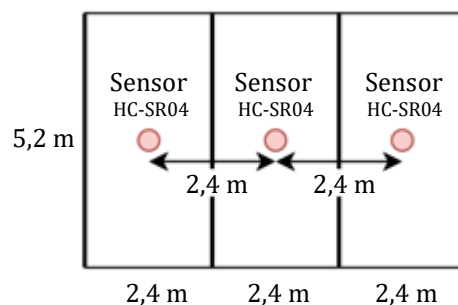
2.2.1. Letak Sensor Ultrasonik HC-SR04

Sensor ultrasonik HC-SR04 diletakkan pada bagian atas slot parkir dan mendeteksi jarak ke arah bawah. Sensor dipasang dengan ketinggian kurang lebih 220 cm dari permukaan lantai. Gambar 2 menunjukkan ilustrasi posisi sensor terhadap mobil pada slot parkir.



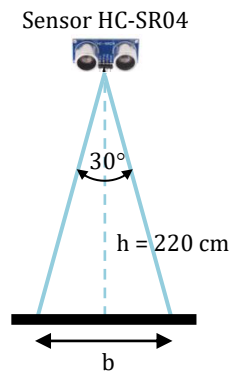
Gambar 2. Ilustrasi letak sensor HC-SR04.

Sensor akan mengukur nilai jarak x . Hasil pengukuran nilai x inilah yang akan menentukan apakah terdapat mobil pada slot parkir atau tidak. Sensor diletakkan tepat pada titik tengah slot parkir. Dimensi slot parkir diambil secara umum yaitu 2,4 m x 5,2 m. Dengan data tersebut, posisi sensor harus diletakkan dengan rentang 2,4 meter dari titik tengah slot parkir seperti yang ditunjukkan pada Gambar 3.

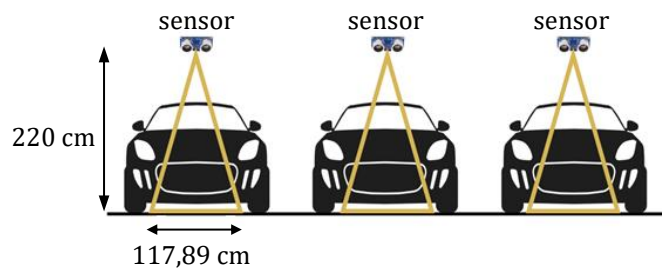


Gambar 3. Jarak antar sensor (tampak atas).

Sedangkan jika dilihat dari depan maka akan tampak seperti Gambar 4. Menurut *datasheet*, sensor ultrasonik HC-SR04 memiliki sudut efektif pengukuran jarak sekitar 30° . Dengan mengetahui ketinggian sensor dan sudut efektif dari sensor maka dapat dihitung nilai b yaitu 117,89 cm atau 1,17 m. Karena nilai b lebih kecil dibandingkan jarak antar sensor yaitu 2,4 m, maka dapat diambil kesimpulan tidak akan terjadi kesalahan deteksi dari sensor-sensor yang bersebelahan. Gambar 5 memperlihatkan rentang deteksi sensor dimana tidak ada *overlap* area deteksi dari ke 3 sensor yang bersebelahan.



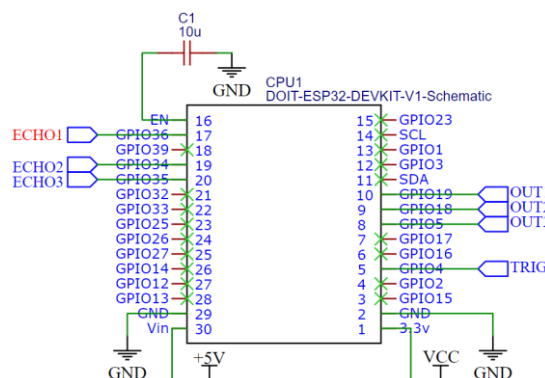
Gambar 4. Area deteksi sensor (tampak depan).



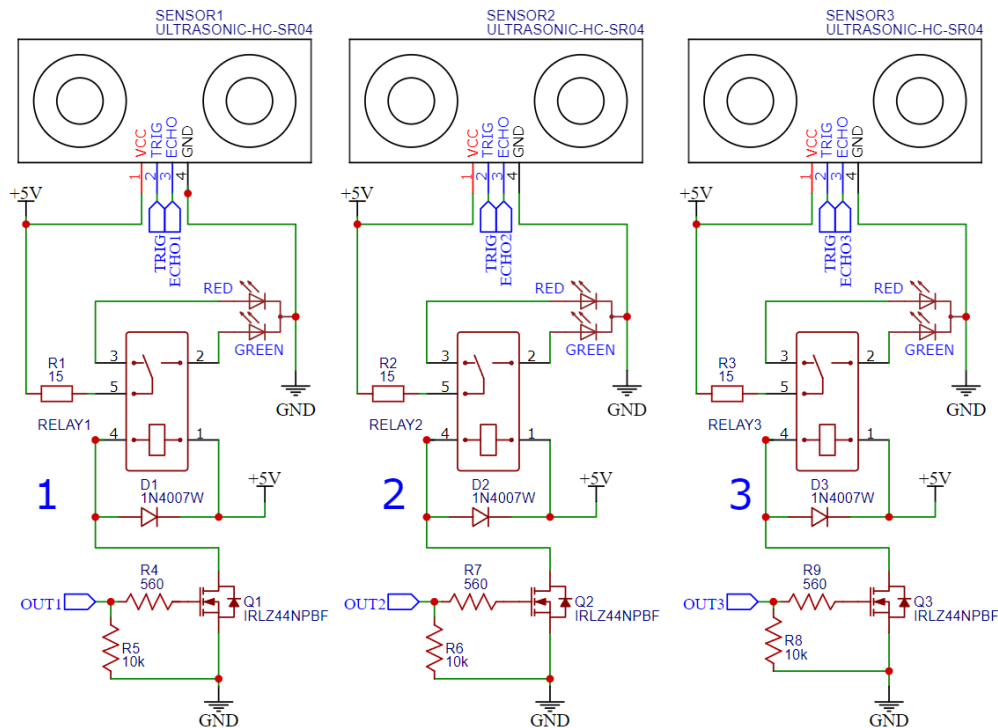
Gambar 5. Rentang deteksi sensor.

2.2.2. Rangkaian Elektronika

Rangkaian elektronika dibutuhkan untuk menghubungkan setiap 1 pengendali mikro ESP32 dengan 3 buah sensor ultrasonik HC-SR04 dan 3 buah lampu indikator. Skema rangkaian elektronika yang dibuat dapat dilihat pada Gambar 6 dan Gambar 7. *Pin echo* dari sensor ultrasonik HC-SR04 dihubungkan ke ESP32 melalui GPIO 36, GPIO 39 dan GPIO 35, dan *pin trigger* dihubungkan ke GPIO 4. ESP32 mengontrol lampu indikator yang berupa LED HPL (*High Power Led*) melalui GPIO 19, GPIO 18 dan GPIO 5. Untuk menyalakan lampu indikator digunakan *relay* 5V dan rangkaian penguat yang menggunakan transistor tipe IRLZ44NPBF. Dioda IN4007W pada rangkaian *relay* digunakan untuk melindungi transistor dari arus balik yang ditimbulkan oleh kumparan *relay*. Seluruh rangkaian elektronika ini membutuhkan catu daya 5V DC.



Gambar 6. Skema rangkaian untuk ESP32.

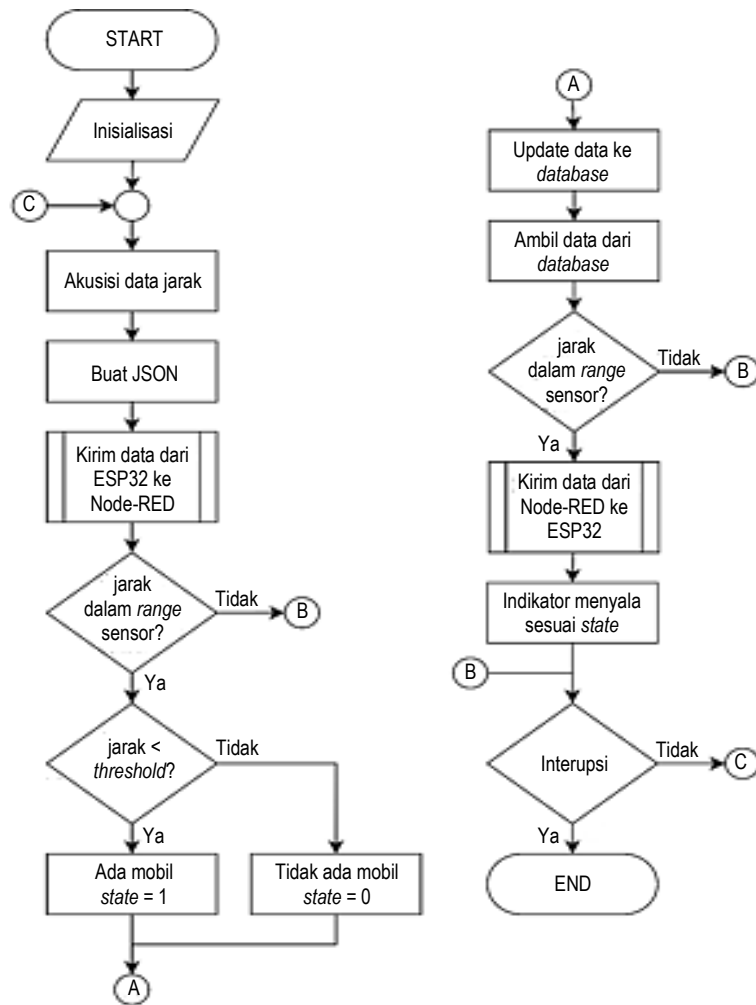


Gambar 7. Skema rangkaian sensor ultrasonik, relay dan lampu indikator.

2.3. Perancangan Perangkat Lunak

Pengendali mikro ESP32 digunakan untuk melakukan 4 tugas utama yaitu melakukan pembacaan jarak dengan sensor ultrasonik HC-SR04, melakukan pengiriman data jarak ke Node-RED yang berada di server dengan menggunakan protokol MQTT, menerima data *state* dari Node-RED dengan protokol MQTT, dan menyalakan lampu indikator sesuai dengan informasi data *state* yang diterima. Sedangkan program Node-RED di *server* melakukan tugas yaitu menerima data dari ESP32, mengolah data *state*, mengupdate *database* dan mengolah gambar untuk halaman *web* (denah lahan parkir). Diagram alir sistem dapat dilihat pada Gambar 8.

Proses awal adalah melakukan inisialisasi untuk koneksi WiFi & MQTT pada ESP32, MQTT *broker* dan Node-RED. Setelah koneksi MQTT sudah dibuat, proses selanjutnya ESP32 mengirimkan data jarak yang diambil dengan sensor ultrasonik HC-SR04. Data jarak dikemas menggunakan format JSON (*JavaScript Object Notation*) agar lebih ringkas, dan dikirimkan dengan protokol MQTT menggunakan koneksi WiFi. Data dengan format JSON adalah umum digunakan untuk format pertukaran data. Data diterima oleh Mosquitto dan diteruskan ke Node-RED dengan protokol MQTT. Data yang masuk ke Node-RED kemudian diolah dan divalidasi apakah jarak yang dikirimkan berada pada *range* atau jangkauan sensor atau tidak. Untuk mendapatkan nilai *state* digunakan sebuah kondisional untuk membandingkan data jarak dengan nilai *threshold* (batas deteksi jarak yang sudah ditentukan). Jika terdeteksi ada mobil maka nilai *state* = 1 dan jika tidak ada mobil maka nilai *state* = 0. Data *state* ini disimpan di dalam *database* sesuai dengan lokasi setiap sensor pada slot parkir. Sebuah fungsi pada Node-RED kemudian memanggil data *state* terakhir untuk dikirimkan kembali ke ESP32. Lalu ESP32 akan menyalakan lampu indikator pada lokasi slot parkir yang sesuai, dengan warna merah jika nilai *state* = 1 atau hijau jika nilai *state* = 0. Pada saat yang sama, slot parkir pada halaman *web* juga akan mengikuti nilai dari *state*. Halaman *web* berisikan denah dari lahan parkir dengan slot parkir berwarna merah jika terdapat mobil, atau slot parkir berwarna hijau jika tidak terdapat mobil. Semua proses ini akan terus diulang (dari titik C) sampai terjadi *interupsi* yaitu catu daya dimatikan dan program berhenti.



Gambar 8. Diagram alir dari sistem.

3. Hasil dan Pembahasan

3.1. Realisasi Perangkat Keras

Rangkaian elektronika yang berisi pengendali mikro ESP32 dan *relay* yang telah dirakit pada PCB dapat dilihat pada Gambar 9 (a). Rangkaian tersebut kemudian dimasukkan ke dalam boks seperti pada Gambar 9 (b). Sedangkan sensor ultrasonik dan lampu indikator dipasang secara terpisah dari rangkaian ESP32 dan diberi penutup transparan seperti Gambar 10. Setiap 1 slot parkir dipasang 1 sensor ultrasonik dan 1 lampu indikator yang terhubung dengan rangkaian ESP32. Di dalam sistem yang dirancang ini, setiap 1 rangkaian ESP32 untuk menangani 3 slot parkir sehingga realisasi perangkat keras untuk 3 slot parkir adalah seperti pada Gambar 11. Rangkaian ESP32, sensor ultrasonik dan lampu indikator dipasang di bagian atas slot parkir.



(a)



(b)

Gambar 9. (a) Rangkaian ESP32 dan *relay* yang sudah dirakit pada PCB
(b) Boks untuk rangkaian ESP32.



Gambar 10. Sensor ultrasonik dan lampu indikator.



Gambar 11. Realisasi perangkat keras untuk 3 slot parkir.

3.2. Pengujian Nilai Batas Deteksi Mobil

Untuk menentukan nilai batas (*threshold*) deteksi mobil, dilakukan uji coba secara empiris untuk menemukan nilai batas deteksi yang paling optimal. Sensor diletakkan dengan ketinggian 220 ± 1 cm. Nilai batas ditetapkan dari 100-200 cm dengan interval 25 cm dengan jarak dihitung dari sensor. Sebuah penghalang ditempatkan di bawah sensor untuk menirukan adanya mobil pada slot parkir. Penghalang diletakkan dengan ketinggian 100-200 cm dengan interval 25 cm (mengikuti tinggi mobil pada umumnya) dengan jarak dihitung dari lantai. Tabel 1 menunjukkan data yang diperoleh dari pengujian nilai batas deteksi. Tanda ✓ menunjukkan bahwa penghalang terdeteksi, sedangkan tanda ✕ menunjukkan bahwa penghalang tidak terdeteksi.

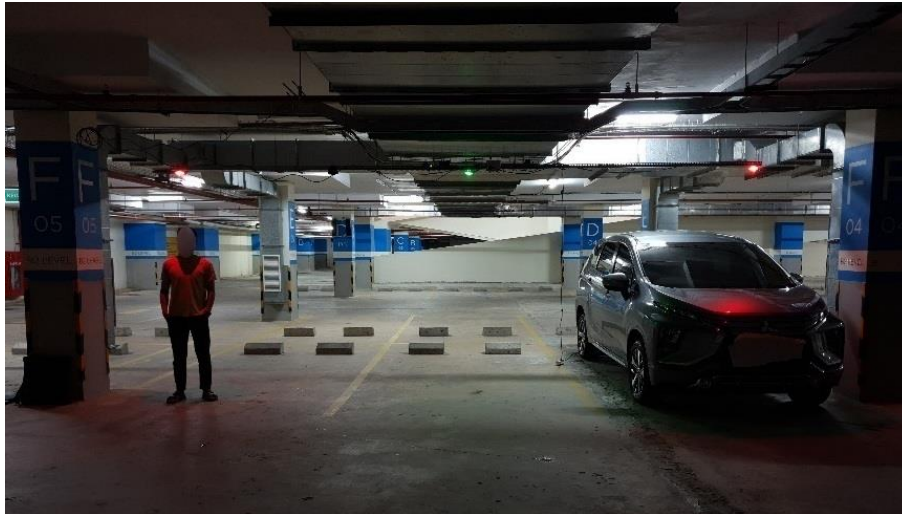
Tabel 1. Data pengujian nilai batas deteksi.

Nilai batas deteksi x (cm)	Tinggi penghalang (cm)				
	100	125	150	175	200
100	✕	✓	✓	✓	✓
125	✕	✓	✓	✓	✓
150	✓	✓	✓	✓	✓
175	✓	✓	✓	✓	✓
200	✓	✓	✓	✓	✓

Nilai batas deteksi yang dapat diterima adalah 150 cm, 175 cm, dan 200 cm. Dari tiga nilai tersebut, dipilih nilai batas 150 cm karena dianggap sudah cukup bagus untuk mendeteksi mobil pada slot parkir.

3.3. Pengujian dengan Mobil

Pengujian deteksi dengan mobil dilakukan untuk mengkonfirmasi dan memvalidasi sistem yang telah dibuat. Pengujian dilakukan dengan mobil secara riil di slot parkir yang sebenarnya. Obyek manusia terdeteksi juga oleh sistem namun karena keberadaan manusia di slot parkir dapat menghalangi mobil parkir maka bisa diasumsikan bahwa slot parkir tersebut tidak dapat dipakai atau tidak tersedia. Gambar 12 memperlihatkan saat dilakukan pengujian deteksi mobil dengan obyek mobil dan manusia pada 3 slot parkir yang bersebelahan. Terlihat lampu indikator yang berada di atas slot parkir menyala dengan warna sesuai dengan nilai *state*.



Gambar 12. Pengujian dengan obyek mobil dan manusia.

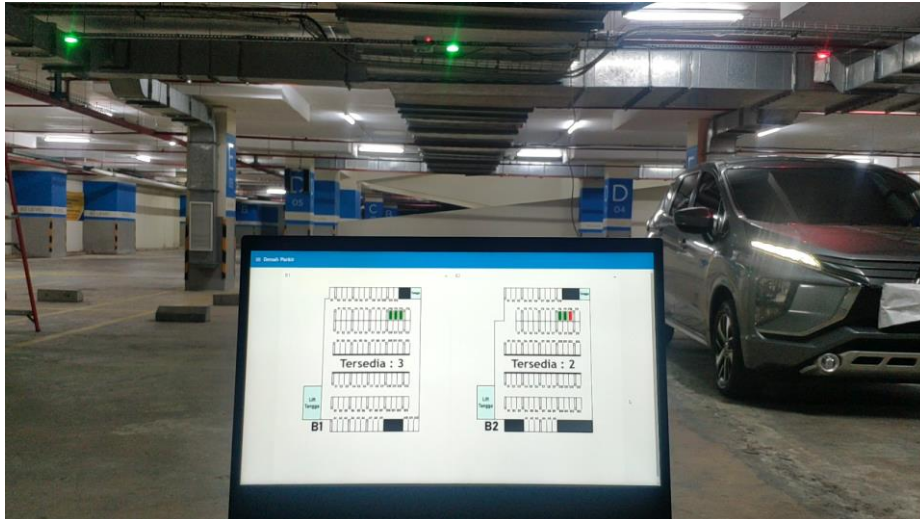
Tabel 2. Data Pengujian dengan mobil.

No	machineID	Ada Mobil?			Pembacaan Jarak (cm)			Kondisi Lampu Indikator		
		Slot 1	Slot 2	Slot 3	Slot 1	Slot 2	Slot 3	Slot 1	Slot 2	Slot 3
1	1	Tidak	Tidak	Tidak	219	221	219	H	H	H
2		Tidak	Tidak	Ya	218	223	99	H	H	M
3		Tidak	Ya	Tidak	216	60	218	H	M	H
4		Tidak	Ya	Ya	216	62	95	H	M	M
5		Ya	Tidak	Tidak	102	220	220	M	H	H
6		Ya	Tidak	Ya	52	221	58	M	H	M
7		Ya	Ya	Tidak	53	97	221	M	M	H
8		Ya	Ya	Ya	100	97	59	M	M	M
9	2	Tidak	Tidak	Tidak	217	221	218	H	H	H
10		Tidak	Tidak	Ya	218	221	57	H	H	M
11		Tidak	Ya	Tidak	218	101	218	H	M	H
12		Tidak	Ya	Ya	218	98	57	H	M	M
13		Ya	Tidak	Tidak	59	221	219	M	H	H
14		Ya	Tidak	Ya	57	220	95	M	H	M
15		Ya	Ya	Tidak	121	97	220	M	M	H
16		Ya	Ya	Ya	123	94	92	M	M	M

Keterangan:

H = Hijau M = Merah

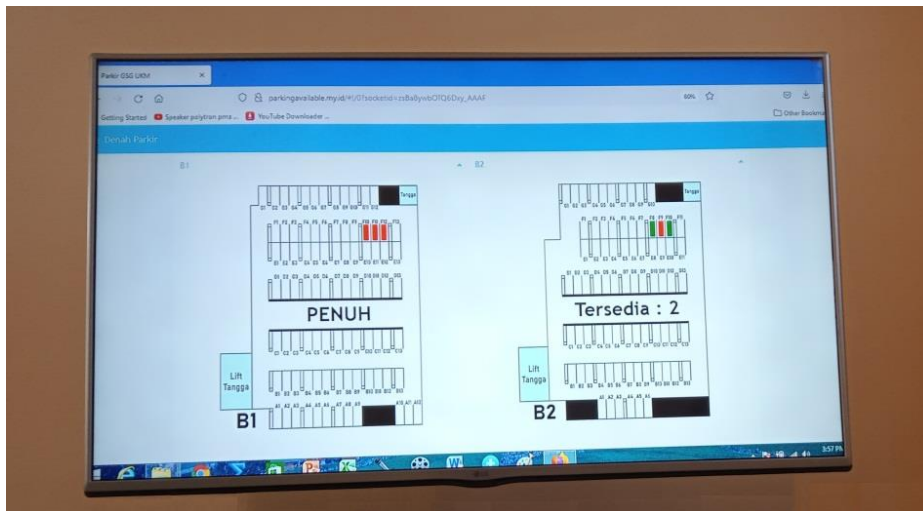
Tabel 2 menunjukkan data dari hasil pengujian menggunakan mobil penumpang. Pada tabel tersebut terdapat 2 buah rangkaian ESP32 yang dinyatakan dengan *machine ID* 1 dan *machine ID* 2. *Machine ID* 1 diletakkan pada lantai B1 pada slot parkir F10, F11, dan F12 sedangkan *machine ID* 2 diletakkan pada lantai B2 pada slot parkir F8, F9, dan F10. Sensor berhasil mendeteksi mobil pada slot parkir dan sistem berhasil menunjukkan kondisi tersebut pada halaman *web* dan lampu indikator. Informasi dan visualisasi ketersediaan parkir yang ditampilkan pada laptop saat pengujian sistem dapat dilihat pada Gambar 13. Slot parkir yang terisi mobil ditunjukkan dengan warna merah sedangkan slot parkir yang kosong berwarna hijau. Warna slot parkir pada denah parkir dan warna lampu indikator dapat berubah sesuai dengan kondisi slot parkir secara *real time*. Pada tampilan tersebut juga tertera jumlah slot parkir kosong yang tersedia di lantai B1 dan B2.



Gambar 13. Tampilan informasi dan visualisasi ketersediaan parkir pada laptop saat dilakukan pengujian.

3.4. Hasil Luaran Sistem

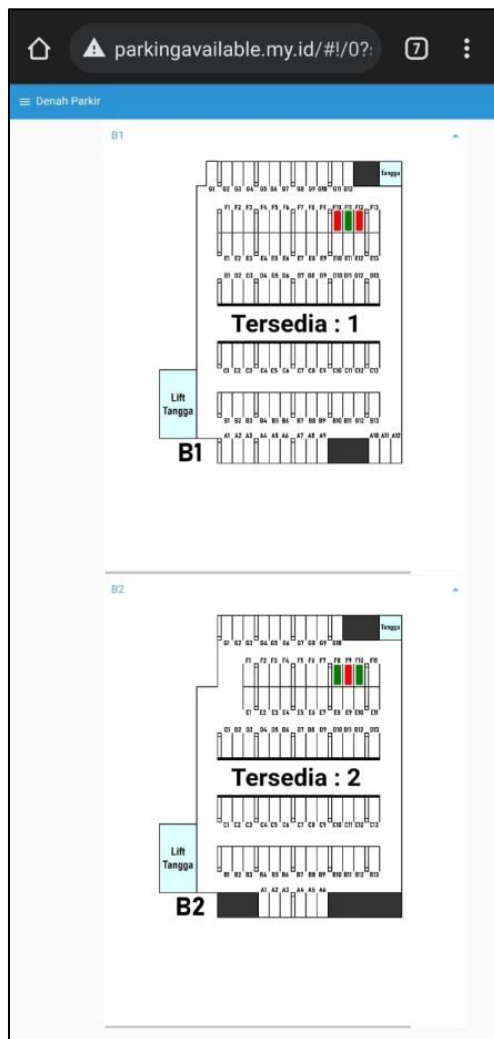
Hasil luaran sistem yaitu informasi dan visualisasi ketersediaan parkir bagi pengunjung yang ditampilkan pada layar *display* berupa TV LED berukuran cukup besar dapat dipasang pada jalur masuk setiap lantai parkir ditunjukkan oleh Gambar 14. Hal ini untuk memudahkan pengunjung saat mencari slot parkir kosong di semua lantai. Pengunjung juga dapat melihat lampu-lampu indikator yang terpasang di atas setiap slot parkir seperti pada Gambar 13 untuk menemukan slot parkir kosong di lantai tersebut terutama apabila posisi pengunjung sudah agak jauh dari layar *display*.



Gambar 14. Tampilan informasi dan visualisasi ketersediaan parkir pada sebuah TV LED.

Sistem juga dapat diakses dari internet melalui *smartphone* pengunjung menggunakan aplikasi peramban *web*. Untuk lebih memudahkan akses menuju halaman *web*, digunakan halaman *web* dengan domain tertentu dari penyedia domain. Domain yang digunakan adalah *parkingavailable.my.id*. Jadi alamat *web server* pada *virtual machine* Microsoft Azure yang digunakan dalam penelitian ini pada saat uji coba adalah *parkingavailable.my.id*. Dari domain tersebut kemudian diarahkan ke *public ip address* dari server menggunakan *DNS management*. Lalu agar pengunjung dapat mengakses halaman *web* tanpa harus memasukkan *port* yang ada di

server maka digunakan fitur *reverse proxy* dari aplikasi NGINX. Gambar 15 memperlihatkan tampilan pada *smartphone* saat mengakses alamat *parkingavailable.my.id*.



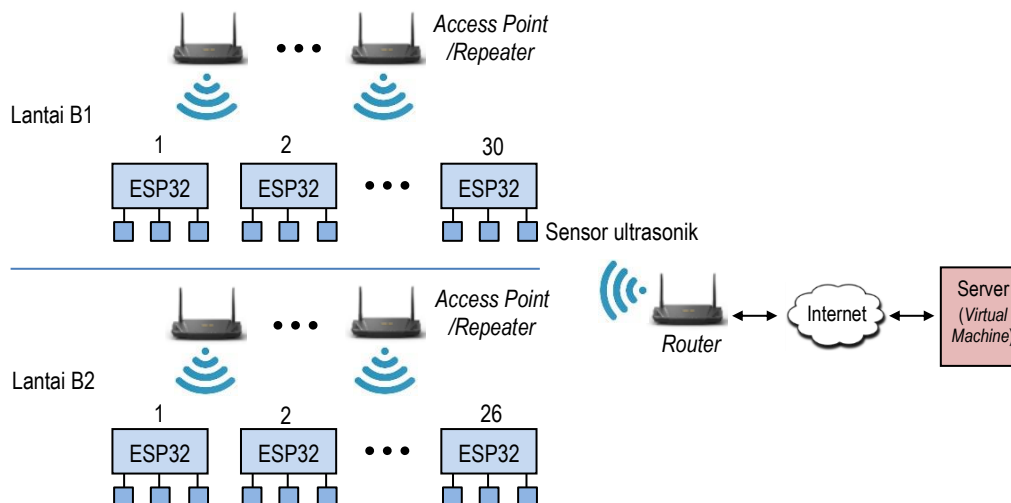
Gambar 15. Tampilan informasi dan visualisasi ketersediaan parkir pada *smartphone* pengunjung yang diakses dari internet.

Untuk mengubah warna indikator slot parkir menjadi merah atau hijau sesuai dengan kondisi slot parkir digunakan SVG (*Scalable Vector Graphics*) yang dijalankan di dalam program Node-RED. SVG adalah bentuk gambar vektor 2 dimensi yang didasarkan pada XML (*eXtensible Markup Language*).

3.5. Implementasi Sistem Secara Riil

Walaupun dalam penelitian ini hanya dilakukan uji coba pada 3 slot parkir pada lantai B1 dan 3 slot parkir pada lantai B2, tetapi tim peneliti meyakini bahwa sistem ini dapat direalisasikan pada seluruh slot parkir pada gedung parkir bertingkat. Gedung Administrasi Pusat Universitas Kristen Maranatha tempat uji coba penelitian ini memiliki 89 slot parkir pada lantai B1 dan 77 slot parkir pada lantai B2. Karena setiap 3 slot parkir diperlukan 1 ESP32 dengan 3 sensor ultrasonik maka di lantai B1 membutuhkan 30 ESP32 dan di lantai B2 membutuhkan 26 ESP32. Jadi seluruh slot parkir dapat diimplementasikan dengan total 56 ESP32. Jumlah ini dapat berkurang apabila jumlah sensor ultrasonik yang terpasang pada 1 buah ESP 32 lebih dari 3 sensor. Secara teoritis 1 buah ESP32 dapat dipasang hingga 10 sensor ultrasonik. Namun semakin banyak sensor ultrasonik yang dipasang pada ESP32 akan semakin merepotkan dalam

pemasangan kabel-kabel penghubungnya. Perangkat keras yang digunakan untuk jaringan WiFi seperti *router* dan *repeater* harus ditambah untuk mengakomodasi jaringan WiFi untuk semua ESP32 yang akan digunakan. Gambar 16 memperlihatkan ilustrasi jika sistem diimplementasi pada seluruh slot parkir.



Gambar 16. Ilustrasi implementasi seluruh slot parkir.

Pada implementasinya protokol MQTT digunakan untuk mengirimkan data jarak dari ESP32 (*machine*) ke Node-RED dan mengirimkan data *state* dari Node-RED ke ESP32. Setiap *machine* mengirimkan data yang dikemas menggunakan JSON ke Node-RED dengan topologi *Many-to-1* (*many machine to 1 Node-RED*). Setiap JSON berisikan identifier *machineID* yang menunjukkan identitas dari setiap *machine*. Gambar 17 memperlihatkan data dalam format JSON yang dibuat untuk transmisi dari *machine* (ESP32) ke Node-RED.

```
JSON =
{
  "machineID": 1,
  "jarak": [
    22,
    150,
    39
  ],
  "rssi": -56
}
```

Gambar 17. Data dalam format JSON yang dikirim dari ESP32 ke Node-Red.

Setelah data jarak diolah menjadi data *state* di dalam program Node-RED, data tersebut dikirimkan dalam format JSON dengan topologi *1-to-Many* (*1 Node-RED to many machine*). JSON yang dikirimkan dari Node-RED memuat seluruh data *state* untuk semua *machineID* dan bersifat *broadcast*. Di setiap *machine* (ESP32), jika *machineID* yang diterima dalam JSON cocok dengan *machineID* yang terprogram, maka *state* dari *machineID* tersebut diambil. Data dalam format JSON yang dibuat untuk transmisi dari Node-RED ke *machine* (ESP32) dapat dilihat pada Gambar 18.

```
JSON =
[
  {
    "machineID": 1,
    "state": [
      1,
      0,
      1
    ],
  },
  {
    "machineID": 2,
    "state": [
      0,
      1,
      1
    ],
  },
]
```

Gambar 18. Data dalam format JSON yang dikirim dari Node-Red ke ESP32.

Data yang dikirim dari ESP32 ke Node-RED dan dari Node-RED ke ESP32 mempunyai *packet size* yang kecil. Mosquitto sebagai MQTT *broker* dapat menerima sampai dengan 60 ribu pesan per detik untuk QoS (*Quality of Service*) 0 dan lebih dari cukup untuk implementasi sistem ini. QoS yang digunakan pada pengiriman pesan dari machine ke Node-RED adalah QoS 0. QoS 0 digunakan sebagai hasil dari implementasi *library PubSubClient* oleh O'Nick Knolleary yang hanya dapat melakukan *publish* pada QoS 0. *Library* ini dipilih karena merupakan *library* yang paling banyak digunakan untuk protokol MQTT. *Library* ini dapat melakukan *publish* pada QoS 0, QoS 1, dan QoS 2 tetapi QoS 1 dan QoS 2 hanya bersifat semu/tidak nyata. Jika QoS 1 atau QoS 2 dipilih, pesan tetap terkirim dengan QoS 0 dengan label QoS 1 atau QoS 2.

4. Kesimpulan

Sistem yang dirancang berhasil mendeteksi ketersediaan slot parkir menggunakan sensor ultrasonik HC-SR04. Sensor mampu mendeteksi dengan baik obyek mobil dengan ketinggian 100-200 cm dan tidak terjadi kesalahan deteksi dari sensor-sensor yang bersebelahan. Data dari sensor ultrasonik berhasil dikirimkan ke *server* menggunakan protokol MQTT agar pemakaian *bandwidth* lebih efisien. Penggunaan *server* berupa *virtual machine* yang berada di *cloud* memberikan kemudahan untuk diakses dari internet menggunakan *smartphone* pengunjung hanya dengan aplikasi peramban *web*. Dari hasil pengujian sistem secara riil di gedung parkir, sistem berhasil menampilkan visualisasi ketersediaan slot parkir kosong di setiap lantai parkir dalam bentuk gambar denah parkir pada layar *display* dan *smartphone* pengunjung. Slot parkir yang terisi mobil ditunjukkan dengan warna merah sedangkan slot parkir yang kosong berwarna hijau. Warna slot parkir pada denah parkir dan warna lampu indikator pada slot parkir dapat berubah sesuai dengan kondisi slot parkir secara *real time*.

Penelitian lebih lanjut untuk pengembangan sistem perlu dilakukan dengan memperbanyak jumlah sensor ultrasonik dan ESP32 secara bertahap hingga jumlah sensor ultrasonik sama dengan jumlah slot parkir yang sebenarnya di setiap lantai parkir. Pengujian deteksi mobil perlu dilakukan kembali untuk memastikan tidak terjadi interferensi antara sensor-sensor ultrasonik yang berdekatan.

Ucapan Terima Kasih

Ucapan terima kasih kami sampaikan kepada Institusi Universitas Kristen Maranatha yang telah mendukung dan mendanai penelitian ini.

Referensi

ISSN (e): 2540-9123

ISSN (p): 2502-1982

- [1] B. Oktaviastuti and H. S. Wijaya, "Urgensi Pengendalian Kendaraan Bermotor Di Indonesia," *J. Rekayasa Teknik Sipil Univ. Madura*, vol. 2, no. 1, pp. 5–8, 2017.
- [2] S. A. Hassan and M. Safaa Husian, "Sustainability consideration in multistory car parking design University multistory car parking as an example Sustainability consideration in multistory car parking design University multistory car parking as an example," *Proceeding ICCEET 2019*, no. August, p. 8, 2019.
- [3] F. A. Imbiri, N. Taryana, and D. Nataliana, "Implementasi Sistem Perparkiran Otomatis dengan Menentukan Posisi Parkir Berbasis RFID," *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 4, no. 1, p. 31, 2018, doi: 10.26760/elkomika.v4i1.31.
- [4] A. Kristanto, "Rancang Bangun Sistem Parkir Otomatis Pada Kampus II ITN Malang Menggunakan Minimum Sistem Arduino Dengan Website Sebagai Media Pelaporan," *J. Mhs. Tek. Inform.*, vol. 3, no. 1, pp. 46–52, 2019.
- [5] D. Nataliana, I. Syamsu, and G. Giantara, "Sistem Monitoring Parkir Mobil menggunakan Sensor Infrared berbasis Raspberry Pi," *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 2, no. 1, p. 68, 2014, doi: 10.26760/elkomika.v2i1.68.
- [6] I. Iszaidy *et al.*, "IOT Smart Guidance Parking Search System for Open Space Parking Area," *J. Phys. Conf. Ser.*, vol. 1962, no. 1, 2021, doi: 10.1088/1742-6596/1962/1/012060.
- [7] A. D. Limantara, Y. C. S. Purnomo, and S. W. Mudjanarko, "Pemodelan Sistem Pelacakan Lot Parkir Kosong Berbasis Sensor Ultrasonic Dan Internet of Things (Iot) Pada Lahan Parkir Diluar Jalan," *Semin. Nas. Sains dan Teknol.*, vol. 1, no. 2, pp. 1–10, 2017, [Online]. Available: jurnal.umj.ac.id/index.php/semnastek.
- [8] A. Banks and R. Gupta, "MQTT Version 3.1.1 OASIS Standard," 2014. <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html> (accessed May 21, 2023).
- [9] N. P. Windryani, N. Bogi, and R. Mayasari, "Analisa Perbandingan Protokol MQTT Dengan HTTP Pada IoT Platform Patriot," *e-Proceeding Eng.*, vol. 6, no. 2, pp. 3192–3199, 2019.
- [10] A. Amrullah, A. Nugroho, and Z. Ramadhan, "Perbandingan Kinerja Webserver Pada Penyedia Layanan Cloud Microsoft Azure Dan Amazon Web Services Menggunakan Metode Benchmarking," *Jinteks*, vol. 5, no. 1, pp. 92–97, 2023.