

**APLIKASI SISTEM KONTROL PORTAL PARKIR
MENGUNAKAN METODE *LOCK* GPS
BERBASIS *INTERNET OF THINGS*
(Studi Kasus: Lahan Parkir Masjid Raya Mujahidin Pontianak)**

[¹]Weldi, [²]Dedi Triyanto, [³]Uray Ristian

[¹][²][³]Jurusan Rekayasa Sistem Komputer, Fakultas MIPA Universitas Tanjungpura
Jalan Prof Dr. H. Hadari Nawawi Pontianak
Telp./Fax. : (0561) 577963
e-mail: [¹]weldi96@student.untan.ac.id, [²]dedi.triyanto@siskom.untan.ac.id,
[³]eristian@siskom.untan.ac.id

ABSTRAK

Saat ini teknologi sistem parkir terutama di gedung-gedung bertingkat seperti swalayan, rumah sakit dan perkantoran sudah menggunakan sistem kendali dalam pengoperasiannya. Akan tetapi, sistem yang ada masih menggunakan operator yang bertugas menjalankan sistem parkir. Salah satu sistem parkir yang masih menggunakan operator/petugas parkir untuk menjalankannya adalah sistem parkir di Masjid Raya Mujahidin Pontianak, dimana petugas parkir harus memindahkan *cone* untuk membuka atau menutup akses lahan parkir khusus di lingkungan Masjid. Oleh karena itu dibutuhkan suatu sistem dan alat yang dapat membuka atau menutup akses lahan parkir menggunakan kendali jarak jauh berbasis *Android* dan internet. Pada sistem kontrol portal parkir, motor DC digunakan sebagai penggerak portal parkir dan ESP32 sebagai pengendali sistem serta komunikasi data menggunakan jaringan internet. Metode *lock* GPS digunakan untuk membatasi wilayah akses penggunaan sistem dalam melakukan kendali sistem portal parkir. Hasil pengujian menunjukkan bahwa rata-rata waktu respon aplikasi dalam mengendalikan portal parkir yang dilakukan sebanyak 5 kali percobaan dengan 5 lokasi yang berbeda adalah 0,27 detik untuk membuka portal parkir dan 0,28 detik untuk menutup portal parkir. Rata-rata waktu yang diperlukan alat dalam mengendalikan portal parkir dalam 5 kali percobaan adalah 19,35 detik untuk membuka portal parkir dan 18,88 detik untuk menutup portal parkir.

Kata kunci: Motor DC, Sistem Kontrol Portal Parkir, *Internet of Things*, *Lock* GPS.

1. PENDAHULUAN

Parkir adalah memberhentikan dan menyimpan kendaraan (mobil, sepeda motor, sepeda, dan sebagainya) untuk sementara waktu pada suatu ruang tertentu. Ruang tersebut dapat berupa tepi jalan, garasi atau pelataran yang disediakan untuk menampung suatu kendaraan [1]. Sistem parkir yang ada saat ini masih menggunakan beberapa operator yang bertugas untuk menjalankan sistem parkir. Salah satu sistem parkir yang masih menggunakan operator/petugas parkir untuk menjalankan sistem parkir adalah sistem parkir yang ada di Masjid Raya Mujahidin Pontianak. Sistem parkir pada Masjid Raya Mujahidin Pontianak masih menggunakan sistem parkir konvensional, dimana petugas parkir harus memindahkan *cone* untuk membuka atau menutup akses lahan parkir khusus di lingkungan Masjid. Oleh karena itu dibutuhkan suatu sistem dan

alat yang dapat membuka atau menutup akses lahan parkir tersebut menggunakan kendali jarak jauh berbasis *Android* dan internet. Akan tetapi, kendala yang dihadapi adalah bagaimana caranya sistem ini dapat bekerja hanya dilingkungan sekitar lahan parkir Masjid saja. Jika pengguna berada diluar radius lahan parkir, maka pengguna tidak dapat mengendalikan akses lahan parkir menggunakan internet. Pengguna hanya dapat mengontrol portal parkir pada radius ≤ 50 meter dari Masjid Raya Mujahidin Pontianak sehingga pengguna yang berada pada radius lebih dari 50 meter dari Masjid Raya Mujahidin tidak dapat melakukan kontrol portal parkir dikarenakan berada diluar radius yang ditentukan.

Penelitian yang terkait dengan sistem parkir berbasis *Internet of Things* pernah dilakukan sebelumnya dengan judul "Pemodelan Sistem Pelacakan Lot Parkir

Kosong Berbasis Sensor Ultrasonik dan *Internet of Things* (IoT) pada Lahan Parkir diluar Jalan” [2]. Penelitian yang terkait dengan parkir pintar berbasis *Smartphone* pernah dilakukan dengan judul “Rancang Bangun *Prototype* Sistem *Smart Parking* Berbasis *Arduino* dan Pemantauan melalui *Smartphone*” [3]. Selanjutnya penelitian yang terkait dengan sistem parkir juga pernah dilakukan sebelumnya dengan judul “Pengembangan Sistem Parkir Otomatis Menggunakan *Arduino Mega 2560* Berbasis *Website*” [1]. Penelitian yang terkait dengan metode *lock* GPS pernah dilakukan sebelumnya dengan judul “Aplikasi Absensi Menggunakan Metode *Lock* GPS dengan *Android* di PT. PLN (Persero) App Malang Basecamp Mojokerto” [4].

Berdasarkan dari masalah yang ada, maka dilakukan penelitian dengan judul “Aplikasi Sistem Kontrol Portal Parkir Menggunakan Metode *Lock* GPS Berbasis *Internet Of Things*”. Penelitian ini mengembangkan sebuah aplikasi sistem kontrol portal parkir di Masjid Raya Mujahidin Pontianak. Sistem ini dapat mengendalikan portal pada lahan parkir khusus serta mengetahui *log* aktivitas pengguna yang menggunakan sistem kendali lahan parkir khusus.

2. LANDASAN TEORI

2.1 Sistem Portal Parkir

Portal parkir merupakan palang parkir yang digunakan untuk menghalang pintu masuk dan keluar kendaraan dengan tujuan untuk membantu pengaturan kendaraan yang diperbolehkan masuk atau keluar area parkir [5]. Bentuk fisik dari portal parkir dapat dilihat pada Gambar 1.



Gambar 1 Portal Parkir

2.2 Lock GPS

Lock GPS adalah kondisi dimana GPS *receiver* sudah terhubung dengan sinyal satelit

dan mendapatkan sinyal informasi dari satelit. Sebuah GPS *receiver* harus mengunci sinyal minimal tiga satelit untuk menghitung posisi 2D (*latitude* dan *longitude*) dan jalur pergerakan [4]. Pada penelitian ini *lock* GPS digunakan untuk menentukan radius suatu wilayah dengan cara menentukan titik koordinat. Titik koordinat yang ditentukan yaitu pada lokasi Masjid Raya Mujahidin Pontianak. Penentuan titik koordinat bertujuan untuk membatasi wilayah akses penggunaan aplikasi *Android* untuk membuka dan menutup portal pada parkir khusus di Masjid Raya Mujahidin.

Dalam penggunaan metode *lock* GPS diperlukan perhitungan jarak antara dua lokasi yang berbeda dengan menggunakan teori *Haversin Formula*. *Haversine Formula* merupakan suatu metode yang digunakan untuk mengetahui jarak antar dua titik lokasi pada permukaan bumi berdasarkan bujur dan lintang [6]. Berikut rumus *haversine formula* dapat dilihat pada Persamaan 1.

$$a = \sin^2\left(\frac{\Delta\text{lat}}{2}\right) + \cos(\text{lat}1) \cdot \cos(\text{lat}2) \cdot \sin^2\left(\frac{\Delta\text{long}}{2}\right)$$
$$c = 2 \cdot \text{atan2}\left(\sqrt{a}, \sqrt{1-a}\right)$$
$$d = R \cdot c$$

Keterangan:

R	=	jari-jari bumi sebesar 6371 (km)
Δlat	=	besaran perubahan <i>latitude</i>
Δlong	=	besaran perubahan <i>longitude</i>
c	=	kalkulasi perpotongan sumbu
d	=	jarak (km)

2.3 Android Studio

Android Studio merupakan sebuah *Integrated Development Environment* (IDE) resmi yang digunakan untuk membangun aplikasi yang berjalan pada platform *android* [7]. Bahasa pemrograman utama yang digunakan adalah Java, sedangkan untuk membuat tampilan atau *layout*, digunakan bahasa XML. *Android studio* juga terintegrasi dengan *Android Software Development Kit* (SDK) untuk *deploy* ke perangkat *android*. Pada penelitian ini *android studio* digunakan untuk membangun aplikasi sistem kontrol portal parkir.

2.4 *Firestore*

Firestore Realtime Database adalah *database* yang di-host di *cloud* [8]. Data disimpan sebagai *JSON* dan disinkronkan secara *realtime* ke setiap klien yang terhubung. Ketika *developer* membuat aplikasi lintas-*platform* dengan *SDK Android*, *iOS*, dan *JavaScript*, semua klien akan berbagi sebuah *instance Realtime Database* dan menerima *update* data terbaru secara otomatis. Ketika koneksi perangkat pulih kembali, *Realtime Database* akan menyinkronkan perubahan data lokal dengan *update* jarak jauh yang terjadi selama klien *offline*, sehingga setiap perbedaan akan otomatis digabungkan. Pada penelitian ini *firebase* digunakan sebagai *server* untuk menyimpan *database* jamaah.

2.5 *Internet of Things (IoT)*

Internet of Things (IoT) merupakan suatu konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terus menerus [9]. Konsep prinsip kerja *Internet of Things* mengacu pada 3 elemen utama dalam arsitektur IoT diantaranya, barang fisik yang dilengkapi modul IoT, perangkat koneksi ke internet seperti Modem dan *Router Wireless Speedy* yang ada di rumah, dan *Cloud Data Center* tempat untuk menyimpan aplikasi beserta *database* [10]. Adapun konsep prinsip kerja dari *Internet of Things* dapat dilihat pada Gambar 2.



Gambar 2 Konsep *Internet of Things*

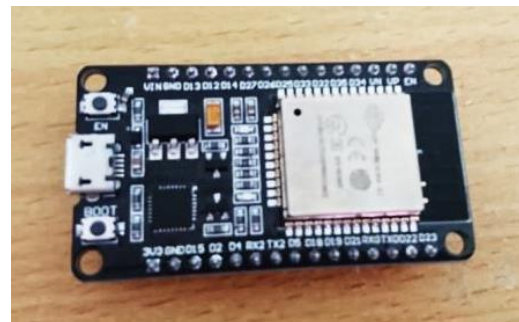
2.6 *Arduino IDE*

Arduino IDE (Integrated Development Environment) adalah sebuah perangkat lunak yang digunakan untuk mengembangkan aplikasi seperti menulis program, meng-*compile* menjadi kode biner dan meng-*upload* ke dalam *memory microcontroller*. *Software Arduino* dapat di install di beberapa *Operating system* diantaranya: *Windows*, *MacOS*, dan *Linux*. Pada penelitian ini *arduino IDE* digunakan untuk menulis kode program yang akan ditanamkan ke *NodeMCU ESP32*.

2.7 *NodeMCU ESP32*

NodeMCU ESP32 memiliki keunggulan yaitu sistem berbiaya rendah, dan juga berdaya rendah dengan modul *WiFi* yang terintegrasi dengan chip mikrokontroler serta memiliki *bluetooth* dengan mode ganda dan fitur hemat daya menjadikannya lebih fleksibel. *NodeMCU ESP32* kompatibel

dengan perangkat seluler dan aplikasi IoT (*Internet of Things*). Mikrokontroler ini dapat digunakan sebagai sistem mandiri yang lengkap atau dapat dioperasikan sebagai perangkat pendukung mikrokontroler host [11]. Pada penelitian ini *NodeMCU ESP32* digunakan sebagai pemogram dan komunikasi data antara alat dan aplikasi. Dapat dilihat bentuk fisik dari *NodeMCU ESP32* pada Gambar 3.



Gambar 3 *NodeMCU ESP32*

2.8 Motor DC

Motor DC memerlukan suplai tegangan yang searah pada kumparan medan untuk diubah menjadi energi mekanik [12]. Jika arus dalam kumparan jangkar berinteraksi dengan medan magnet, maka akan timbul torsi (T) yang akan memutar motor [13]. Pada penelitian ini motor DC berfungsi sebagai penggerak yang dapat membuka dan menutup portal parkir berdasarkan perintah yang telah ditentukan. Bentuk fisik dari motor DC yang dapat dilihat pada Gambar 4.



Gambar 4 Motor DC

3. METODE PENELITIAN

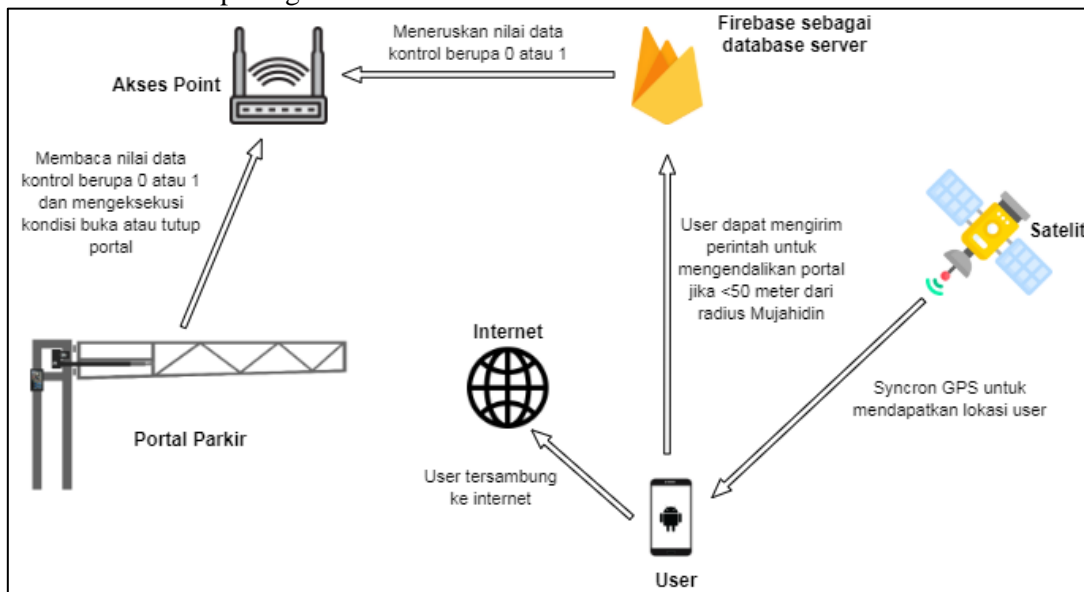
Metode penelitian dimulai dengan studi literatur yang berfungsi untuk mengumpulkan informasi berupa teori-teori pendukung yang akan digunakan dalam penelitian. Selanjutnya dilakukan tahap analisa kebutuhan sistem, yang terdiri dari analisa kebutuhan perangkat keras dan

perangkat lunak. Berdasarkan hasil analisa kebutuhan sistem, dilakukan proses perancangan sistem yang melibatkan komponen perangkat keras dan perangkat lunak. Setelah selesai proses perancangan, maka akan dilanjutkan pada proses integrasi dan implementasi sistem. Tahap terakhir yang dilakukan adalah proses pengujian sistem. Jika sistem tidak berhasil, maka dilakukan ulang pada tahap implementasi. Jika sistem berhasil maka dilanjutkan dengan analisa dan kesimpulan.

4. PERANCANGAN

4.1 Perancangan Sistem Kendali Portal Parkir

Perancangan sistem merupakan rancangan keseluruhan baik dari rangkaian, rancangan tampilan fisik dari portal parkir berbasis *internet of things*, kemudian menempatkan komponen-komponen alat disekitar rancang bangun yang telah ditetapkan dan merancang aplikasi berbasis *android* menggunakan GPS sebagai kendali dari sistem portal parkir. Pengendali sistem dan komunikasi data yang digunakan adalah NodeMCU ESP32. Perancangan sistem kontrol portal parkir menggunakan metode *lock GPS* berbasis *internet of things* dapat dilihat pada Gambar 5.



Gambar 5 Perancangan Sistem

Gambar 5 menunjukkan rancangan alur kerja sistem portal parkir berbasis internet of things. Alur kerja sistem portal parkir dapat diuraikan sebagai berikut:

1. Melakukan sinkronisasi GPS perangkat mobile pengguna yang terhubung ke satelit dan internet untuk mendapatkan lokasi pengguna berupa lokasi *latitude*

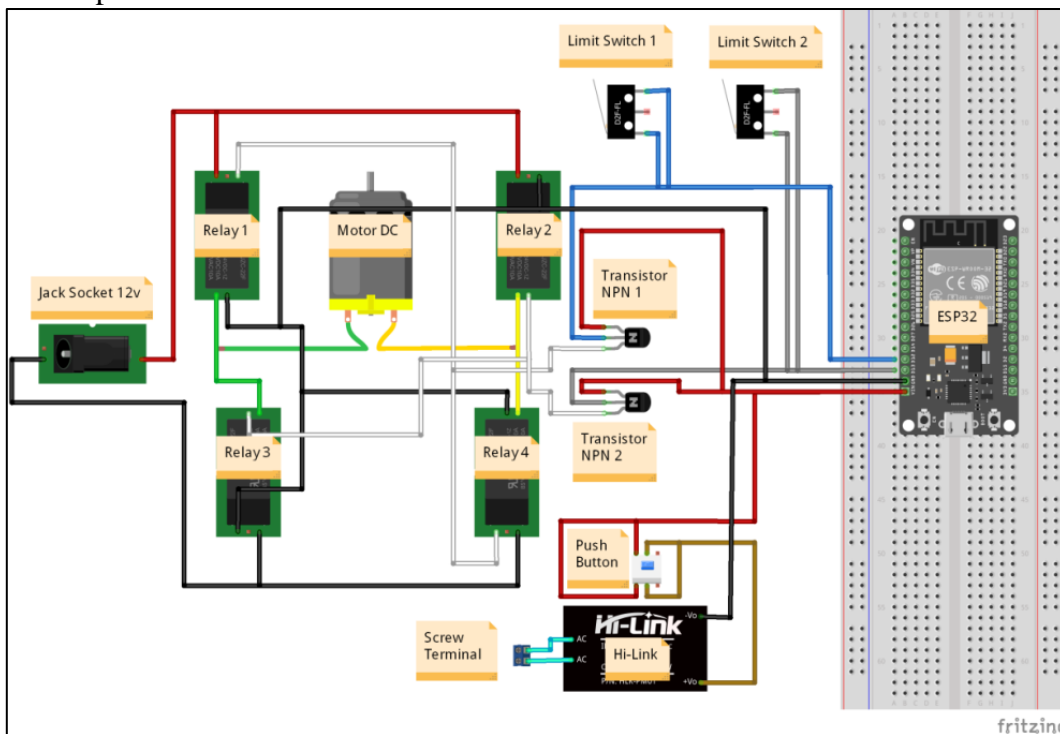
dan *longitude*, kemudian dilakukan konversi berupa jarak pada aplikasi bagian *user*.

2. Setelah mendapatkan jarak berupa angka bulat, kemudian pengguna hanya dapat melakukan kontrol portal parkir apabila jarak pengguna berada ≤ 50 meter dari radius wilayah Masjid Raya Mujahidin.
3. Pada saat pengguna menekan tombol buka portal parkir, maka aplikasi akan mengirim nilai data kontrol pada *firebase* berupa nilai 1.
4. Pada saat pengguna menekan tombol tutup portal parkir, maka aplikasi akan mengirim nilai data kontrol pada *firebase* berupa nilai 0.
5. Portal parkir yang terhubung ke akses point akan membaca nilai data

kontrol pada *firebase*, apabila nilai data kontrol bernilai 1 maka portal parkir akan terbuka sedangkan jika nilai data kontrol bernilai 0 maka portal parkir akan tertutup.

4.2 Perancangan Perangkat Keras

Rancangan perangkat keras terdiri dari NodeMCU ESP32, *limit switch*, motor DC dan rangkaian H-*bridge*. NodeMCU ESP32 sebagai otak pemrosesan data dari semua masukan dan keluaran. Sistem kontrol portal parkir ini menggunakan motor DC sebagai penggerak untuk membuka dan menutup portal parkir. Dalam hal ini, rangkaian H-*bridge* berfungsi menentukan arah gerakan motor DC. Rancangan perangkat keras secara keseluruhan dapat dilihat pada Gambar 6.

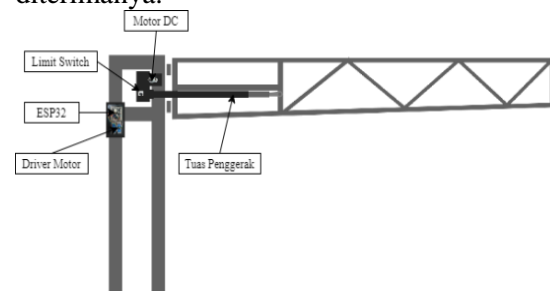


Gambar 6 Perancangan Keseluruhan Perangkat Keras

4.3 Perancangan Portal Parkir

Rancangan Portal Parkir pada Gambar 7 terdiri dari ESP32, motor DC, driver motor, *limit switch* dan tuas penggerak. Motor DC sebagai penggerak untuk membuka dan menutup portal parkir. Driver motor berfungsi untuk mengatur arah putaran motor DC dalam menggerakkan portal parkir untuk membuka dan menutup portal. *Limit switch* sebagai pemberi batas pergerakan motor DC dalam menggerakkan portal parkir. NodeMCU ESP32 bertugas memberikan perintah berupa

respon keluaran sesuai dengan perintah yang diterimanya.

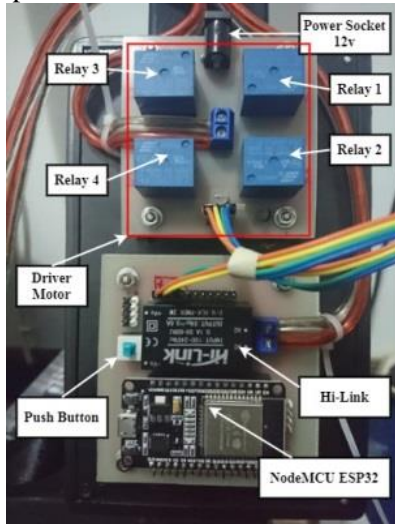


Gambar 7 Perancangan Portal Parkir

5. IMPLEMENTASI, PENGUJIAN DAN PEMBAHASAN

5.1 Implementasi Kendali Portal Parkir

Implementasi kendali portal parkir meliputi perakitan komponen-komponen perangkat keras yang dibutuhkan oleh sistem seperti sistem kendali portal dan portal parkir. Implementasi kendali portal parkir dapat dilihat pada Gambar 8.



Gambar 8 Implementasi Perangkat Keras Kendali Portal Parkir

Kendali portal parkir dikemas dengan kotak berwarna hitam dan diletakkan pada tiang portal. Motor DC yang terhubung dengan kendali portal diletakkan disisi samping atas portal dengan tujuan untuk memudahkan motor DC yang terhubung dengan portal dalam melakukan pergerakan mendorong dan menarik portal parkir. Bentuk fisik dari implementasi portal parkir dapat dilihat pada Gambar 9.

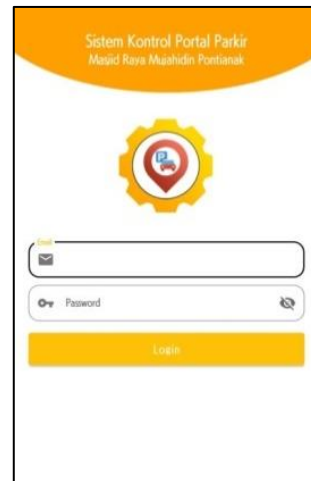


Gambar 9 Implementasi Portal Parkir

5.2 Implementasi Antarmuka Aplikasi Android

5.2.1 Halaman Login

Halaman ini merupakan halaman yang digunakan untuk melakukan proses login oleh pengguna aplikasi. Jika pengguna sudah pernah melakukan proses login pada aplikasi ini, maka halaman tidak akan muncul melainkan langsung menuju halaman utama aplikasi. Tampilan halaman login dapat dilihat pada Gambar 10.



Gambar 10 Halaman Login

5.2.2 Halaman Monitoring

Halaman ini merupakan halaman yang digunakan untuk memantau aktivitas tamu yang menggunakan aplikasi untuk menggunakan fasilitas lahan parkir khusus di Masjid Raya Muahidin Pontianak. Tampilan halaman monitoring dapat dilihat pada Gambar 11.



Gambar 11 Halaman Monitoring

5.2.3 Halaman Kontrol Parkir

Pada halaman kontrol parkir, terdapat dua tombol yaitu tombol buka portal dan tutup portal. Tombol buka portal berfungsi memberikan perintah untuk membuka portal parkir, sedangkan tombol tutup portal berfungsi memberikan perintah untuk menutup portal. Apabila jarak lokasi *user* kurang dari 50 meter dari wilayah radius Masjid Raya Mujahidin, maka portal akan terbuka saat *user* menekan tombol buka portal dan begitu juga dengan menekan tombol tutup portal maka portal akan tertutup. Namun, apabila jarak lokasi *user* lebih dari 50 meter maka *user* tidak dapat membuka maupun menutup portal parkir dan aplikasi akan menampilkan pesan “Anda berada diluar area Mujahidin”. Tampilan dari halaman kontrol parkir dapat dilihat pada Gambar 12.



Gambar 12 Halaman Kontrol Parkir

Pada Gambar 12 dapat dilihat bahwa halaman kontrol parkir menampilkan jarak antara lokasi pengguna dengan lokasi Masjid Raya Mujahidin. Jarak pengguna yang ditampilkan pada aplikasi adalah 29,74 meter dari Masjid Raya Mujahidin. Untuk menguji hasil perhitungan jarak antara 2 lokasi menggunakan teorema *haversine formula* pada aplikasi, selanjutnya dilakukan perhitungan manual menggunakan teorema *haversine formula*.

Pada kasus ini telah didapatkan *latitude* maupun *longitude* dari lokasi pengguna dan lokasi Masjid Raya Mujahidin. *Latitude* dan *longitude* pengguna adalah -0,0413262 dan 109,3360908 sedangkan *latitude* dan *longitude* Masjid Raya Mujahidin adalah -0,041581 dan 109,336172. *Latitude* dan *longitude* pengguna disimbolkan dengan *lat1*

dan *long2*, sedangkan *latitude* dan *longitude* disimbolkan dengan *lat1* dan *long1*.

Diketahui:

$$lat1 = -0,0413262$$

$$lat2 = -0,041581$$

$$long1 = 109,3360908$$

$$long2 = 109,336172$$

$$a = \sin^2 \left(\frac{lat2-lat1}{2} \right) + \cos(lat1) \cos(lat2) \sin^2 \left(\frac{long2-long1}{2} \right)$$

$$a = \sin^2 \left(\frac{-0,041581 - (-0,0413262)}{2} \right) + \cos(-0,0413262) \cos(-0,041581) \sin^2 \left(\frac{109,336172 - 109,3360908}{2} \right)$$

$$= \sin^2 (2,2235495670407715E-6) + \cos(-0,0413262) \cos(-0,041581) \sin^2 (7,086036764056446E-7)$$

$$= \sin(2,2235495670407715E-6) \sin(2,2235495670407715E-6) + \cos(-0,0413262) \cos(-0,041581) \sin(7,086036764056446E-7)$$

$$\sin(7,086036764056446E-7) \sin(7,086036764056446E-7)$$

$$= (2,2235494670389393E-6) (2,2235494670389393E-6) + (0,999999739878849) (0,9999997366613652) (7,086036764055853E-7) (7,086036764055853E-7)$$

$$= (2,2235494670389393E-6) (2,2235494670389393E-6) + (0,999999739878849) (0,9999997366613652) (7,086036764055853E-7) (7,086036764055853E-7)$$

$$= (2,2235494670389393E-6) (2,2235494670389393E-6) + (0,999999739878849) (0,9999997366613652) (7,086036764055853E-7) (7,086036764055853E-7)$$

$$= (2,2235494670389393E-6) (2,2235494670389393E-6) + (0,999999739878849) (0,9999997366613652) (7,086036764055853E-7) (7,086036764055853E-7)$$

$$= (2,2235494670389393E-6) (2,2235494670389393E-6) + (0,999999739878849) (0,9999997366613652) (7,086036764055853E-7) (7,086036764055853E-7)$$

$$= (2,2235494670389393E-6) (2,2235494670389393E-6) + (0,999999739878849) (0,9999997366613652) (7,086036764055853E-7) (7,086036764055853E-7)$$

$$= (2,2235494670389393E-6) (2,2235494670389393E-6) + (0,999999739878849) (0,9999997366613652) (7,086036764055853E-7) (7,086036764055853E-7)$$

$$= (2,2235494670389393E-6) (2,2235494670389393E-6) + (0,999999739878849) (0,9999997366613652) (7,086036764055853E-7) (7,086036764055853E-7)$$

$$= (2,2235494670389393E-6) (2,2235494670389393E-6) + (0,999999739878849) (0,9999997366613652) (7,086036764055853E-7) (7,086036764055853E-7)$$

$$= (2,2235494670389393E-6) (2,2235494670389393E-6) + (0,999999739878849) (0,9999997366613652) (7,086036764055853E-7) (7,086036764055853E-7)$$

$$= (2,2235494670389393E-6) (2,2235494670389393E-6) + (0,999999739878849) (0,9999997366613652) (7,086036764055853E-7) (7,086036764055853E-7)$$

$$= (2,2235494670389393E-6) (2,2235494670389393E-6) + (0,999999739878849) (0,9999997366613652) (7,086036764055853E-7) (7,086036764055853E-7)$$

$$= (2,2235494670389393E-6) (2,2235494670389393E-6) + (0,999999739878849) (0,9999997366613652) (7,086036764055853E-7) (7,086036764055853E-7)$$

$$= (2,2235494670389393E-6) (2,2235494670389393E-6) + (0,999999739878849) (0,9999997366613652) (7,086036764055853E-7) (7,086036764055853E-7)$$

$$= (2,2235494670389393E-6) (2,2235494670389393E-6) + (0,999999739878849) (0,9999997366613652) (7,086036764055853E-7) (7,086036764055853E-7)$$

$$= (2,2235494670389393E-6) (2,2235494670389393E-6) + (0,999999739878849) (0,9999997366613652) (7,086036764055853E-7) (7,086036764055853E-7)$$

$$= (2,2235494670389393E-6) (2,2235494670389393E-6) + (0,999999739878849) (0,9999997366613652) (7,086036764055853E-7) (7,086036764055853E-7)$$

$$= (2,2235494670389393E-6) (2,2235494670389393E-6) + (0,999999739878849) (0,9999997366613652) (7,086036764055853E-7) (7,086036764055853E-7)$$

$$= (2,2235494670389393E-6) (2,2235494670389393E-6) + (0,999999739878849) (0,9999997366613652) (7,086036764055853E-7) (7,086036764055853E-7)$$

$$= (2,2235494670389393E-6) (2,2235494670389393E-6) + (0,999999739878849) (0,9999997366613652) (7,086036764055853E-7) (7,086036764055853E-7)$$

$$= (2,2235494670389393E-6) (2,2235494670389393E-6) + (0,999999739878849) (0,9999997366613652) (7,086036764055853E-7) (7,086036764055853E-7)$$

$$= (2,2235494670389393E-6) (2,2235494670389393E-6) + (0,999999739878849) (0,9999997366613652) (7,086036764055853E-7) (7,086036764055853E-7)$$

$$= (2,2235494670389393E-6) (2,2235494670389393E-6) + (0,999999739878849) (0,9999997366613652) (7,086036764055853E-7) (7,086036764055853E-7)$$

$$= (2,2235494670389393E-6) (2,2235494670389393E-6) + (0,999999739878849) (0,9999997366613652) (7,086036764055853E-7) (7,086036764055853E-7)$$

$$= (2,2235494670389393E-6) (2,2235494670389393E-6) + (0,999999739878849) (0,9999997366613652) (7,086036764055853E-7) (7,086036764055853E-7)$$

$$= (2,2235494670389393E-6) (2,2235494670389393E-6) + (0,999999739878849) (0,9999997366613652) (7,086036764055853E-7) (7,086036764055853E-7)$$

$$= (2,2235494670389393E-6) (2,2235494670389393E-6) + (0,999999739878849) (0,9999997366613652) (7,086036764055853E-7) (7,086036764055853E-7)$$

$$= (2,2235494670389393E-6) (2,2235494670389393E-6) + (0,999999739878849) (0,9999997366613652) (7,086036764055853E-7) (7,086036764055853E-7)$$

$$= (2,2235494670389393E-6) (2,2235494670389393E-6) + (0,999999739878849) (0,9999997366613652) (7,086036764055853E-7) (7,086036764055853E-7)$$

$$= (2,2235494670389393E-6) (2,2235494670389393E-6) + (0,999999739878849) (0,9999997366613652) (7,086036764055853E-7) (7,086036764055853E-7)$$

$$= (2,2235494670389393E-6) (2,2235494670389393E-6) + (0,999999739878849) (0,9999997366613652) (7,086036764055853E-7) (7,086036764055853E-7)$$

$$= (2,2235494670389393E-6) (2,2235494670389393E-6) + (0,999999739878849) (0,9999997366613652) (7,086036764055853E-7) (7,086036764055853E-7)$$

$$= (2,2235494670389393E-6) (2,2235494670389393E-6) + (0,999999739878849) (0,9999997366613652) (7,086036764055853E-7) (7,086036764055853E-7)$$

Dari hasil perhitungan manual didapatkan jarak antara lokasi pengguna dengan lokasi Masjid Raya Mujahidin adalah 29,74 m. Hal ini menunjukkan bahwa penggunaan teorema *haversine formula* pada aplikasi dan perhitungan secara manual terbukti berhasil menghitung jarak antara 2 lokasi dengan hasil yang sama.

5.3 Pengujian

5.3.1 Pengujian Buka dan Tutup Portal Parkir

Pengujian buka dan tutup portal parkir bertujuan untuk melihat apakah motor DC yang terhubung dengan portal dapat membuka dan menutup portal secara terus-menerus atau tidak. Pengujian dilakukan sebanyak 5 kali percobaan dengan jeda beberapa menit untuk setiap percobaan. Pengujian dilakukan dengan memberikan nilai 1 dan 0 pada motor DC untuk melihat apakah motor DC dapat bergerak dan membuka portal parkir. Pengujian buka portal parkir dapat dilihat pada Gambar 13.



Gambar 13 Pengujian Buka Portal Parkir

Selanjutnya dilakukan pengujian tutup portal parkir dengan memberikan nilai 0 dan 1 pada motor DC untuk melihat apakah motor DC dapat bergerak menutup portal. Adapun pengujian tutup portal dapat dilihat pada Gambar 14.



Gambar 14 Pengujian Tutup Portal Parkir

Pada pengujian proses buka dan tutup portal parkir dilakukan sebanyak 5 kali percobaan dan masing-masing memiliki waktu respon yang berbeda tiap-tiap percobaannya dikarenakan faktor jaringan internet yang digunakan sangat mempengaruhi kecepatan waktu respon NodeMCU ESP32 untuk membaca nilai yang diberikan sebelum memberikan perintah ke motor DC untuk membuka maupun menutup portal parkir. Berikut data yang diperoleh dari hasil pengujian buka portal parkir dapat

dilihat pada Tabel 1 dan hasil pengujian tutup portal parkir dapat dilihat pada Tabel 2.

Tabel 1 Pengujian Buka Portal

Uji	Gerakan Motor DC	
	Buka Portal	Waktu (detik)
1	Berhasil	20,11
2	Berhasil	19,57
3	Berhasil	19,48
4	Berhasil	18,06
5	Berhasil	19,51
Rata-rata waktu buka portal		19,35 detik

Tabel 2 Pengujian Tutup Portal

Uji	Gerakan Motor DC	
	Tutup Portal	Waktu (detik)
1	Berhasil	18,23
2	Berhasil	18,67
3	Berhasil	19,98
4	Berhasil	19,17
5	Berhasil	18,37
Rata-rata waktu tutup portal		18,88 detik

5.3.2 Pengujian Kontrol Portal Parkir Pada Aplikasi *Android*

Pengujian pada halaman kontrol parkir dilakukan untuk mengetahui apakah aplikasi dapat memberikan perintah buka dan tutup portal dengan menekan menu halaman kontrol parkir. Pengujian ini dilakukan di 5 lokasi yang berbeda untuk melihat berapa lama waktu respon buka dan tutup portal. Untuk melihat berapa lama waktu respon buka dan tutup portal yaitu melalui hasil ukur dengan aplikasi seperti Gambar 15.



Gambar 15 Hasil Ukur Waktu Respon Aplikasi

Adapun data yang diperoleh dari hasil pengujian buka dan tutup portal parkir pada aplikasi *android* dapat dilihat pada Tabel 3 dan Tabel 4.

Tabel 3 Pengujian Buka Portal Parkir pada Aplikasi *Android*

Uji	Kontrol Portal Parkir dengan Aplikasi <i>Android</i>			
	Lokasi	Jarak (meter)	Respon Buka Portal Parkir (detik)	Keterangan
1	Sekitar Masjid Mujahidin	5,79	0,28	Berhasil membuka portal parkir
2	Sekitar Masjid Mujahidin	8,86	0,26	Berhasil membuka portal parkir
3	Jl. D.A. Hadi	29,74	0,27	Berhasil membuka portal parkir
4	Jl. Jenderal Ahmad Yani	72,57	-	Pengguna berada diluar area Masjid Mujahidin
5	Jl. Jenderal Ahmad Yani	205,77	-	Pengguna berada diluar area Masjid Mujahidin

Tabel 4 Pengujian Tutup Portal Parkir pada Aplikasi *Android*

Uji	Kontrol Portal Parkir dengan Aplikasi <i>Android</i>			
	Lokasi	Jarak (meter)	Respon Tutup Portal Parkir (detik)	Keterangan
1	Sekitar Masjid Mujahidin	5,79	0,26	Berhasil menutup portal parkir
2	Sekitar Masjid Mujahidin	8,86	0,27	Berhasil menutup portal parkir
3	Jl. D.A. Hadi	29,74	0,31	Berhasil menutup portal parkir
4	Jl. Jenderal Ahmad Yani	72,57	-	Pengguna berada diluar area Masjid Mujahidin
5	Jl. Jenderal Ahmad Yani	205,77	-	Pengguna berada diluar area Masjid Mujahidin

Pengujian dilakukan sebanyak 5 kali percobaan dengan 5 lokasi yang berbeda. Percobaan ke-1 dan ke-2 dilakukan disekitar Masjid Raya Mujahidin dengan jarak 5,79 dan 8,56 meter dari Masjid Raya Mujahidin, percobaan ke-3 pada Jl. D.A. Hadi dengan jarak 29,74 meter dari Masjid Raya

Mujahidin. Hasil percobaan ke-1 sampai percobaan ke-3 aplikasi dapat merespon untuk membuka maupun menutup portal parkir. Percobaan ke-4 dan ke-5 dilakukan pada Jl. Jenderal Ahmad Yani dengan jarak 72,57 dan 205,77 meter dari Masjid Raya Mujahidin, aplikasi tidak dapat merespon untuk membuka maupun menutup portal parkir dikarenakan lokasi pengguna berada diluar radius wilayah yang telah ditentukan yaitu ≤ 50 meter dari lokasi Masjid Raya Mujahidin. Selain untuk melakukan kontrol portal parkir, aplikasi ini juga akan mengirim *log* aktivitas pengguna kedalam *database* yang mana *log* aktivitas ini akan ditampilkan pada aplikasi bagian admin.

6. KESIMPULAN DAN SARAN

6.1 Kesimpulan

Kesimpulan yang dapat diambil dari penelitian ini adalah sebagai berikut:

1. Perancangan alat portal parkir berhasil dilakukan untuk membuka dan menutup portal parkir dengan menggunakan motor DC sebagai penggerak portal dan NodeMCU sebagai kendali portal.
2. Perancangan sistem kendali berhasil dilakukan dengan membuat sebuah aplikasi berbasis *Android* yang dapat membuka dan menutup portal parkir yang terkoneksi ke internet, serta dapat menampilkan *log* aktivitas pengguna lahan parkir khusus.
3. Penggunaan metode *lock* GPS pada sistem kontrol portal parkir berbasis *Internet of Things* di Masjid Raya Mujahidin Pontianak berhasil dilakukan untuk membatasi wilayah dan jarak akses pengguna dalam menggunakan aplikasi untuk membuka dan menutup portal parkir.
4. Rata-rata waktu respon yang diperlukan sistem dalam mengendalikan portal parkir yang dilakukan sebanyak 5 kali percobaan dengan 5 lokasi yang berbeda adalah selama 0,27 detik membuka portal parkir dan 0,28 detik untuk menutup portal parkir. Sedangkan rata-rata waktu yang diperlukan alat dalam mengendalikan portal parkir di Masjid Raya Mujahidin dalam 5 kali percobaan adalah selama 19,35 detik untuk membuka portal parkir dan 18,88 detik untuk menutup portal parkir.

6.2 Saran

Berikut beberapa saran yang dapat diambil dari hasil pengujian yang telah dilakukan adalah sebagai berikut:

1. Menambahkan fitur notifikasi pada aplikasi bagian admin agar memudahkan pengelola Masjid untuk mengetahui tamu-tamu yang datang berkunjung pada saat tamu memasuki lokasi Masjid Raya Mujahidin Pontianak.
2. Menghubungkan *database* yang digunakan pada aplikasi dengan *Running Text* yang ada pada Masjid Raya Mujahidin Pontianak agar dapat menampilkan data nama-nama tamu yang datang berkunjung.
3. Untuk penelitian selanjutnya dapat membuat sistem parkir yang otomatis dalam membuka dan menutup portal jika mobil atau kendaraan yang dikenali oleh sistem melewati portal parkir.

DAFTAR PUSTAKA

- [1] Andi Idham Ramadhan, "Pengembangan Sistem Parkir Otomatis Menggunakan Arduino Mega 2560 Berbasis Website," *Jurnal Coding, Sistem Komputer Untan*, pp. 184-194, 2016.
- [2] Arthur Daniel Limantara, "Pemodelan Sistem Pelacakan Lot Parkir Kosong Berbasis Sensor Ultrasonic dan Interbet of Things (Iot) Pada Lahan Parkir diluar Jalan," *Seminar Nasional Sains dan Teknologi 2017*, pp. 1-10, 2017.
- [3] Rudi, "Rancang Bangun Prototype Sistem Smart Parking Berbasis Arduino dan Pemantaun Melalui Smartphone," *Jurnal Ecotipe*, pp. 14-20, 2017.
- [4] Ronny Makhfuddin Akbar, "Aplikasi Absensi Menggunakan Metode Lock GPS dengan Android di PT. PLN (Persero) App Malang Basecamp Mojokerto," *Majapahit Techno*, pp. 55-63, 2015.
- [5] Muhammad Noor Fais, "Pengembangan Sistem Parkir di Universitas Muria Kudus dengan Menggunakan Enkripsi Data dan Teknologi Barcode," *Jurnal SIMETRIS*, pp. 173-180, 2014.
- [6] Yulianto, "Penerapan Formula Haversine Pada Sistem Informasi Geografis Pencarian Jarak Terdekat Lokasi Lapangan Futsal," *Jurnal Ilmiah Ilmu Komputer*, pp. 14-21, 2018.
- [7] Android. (2020, Februari) Developer Android. [Online]. <https://developer.android.com/studio/intro/index.html>
- [8] Firebase. (2019, Desember) Firebase. [Online]. <https://firebase.google.com/docs/database/index.html>
- [9] Farizqi Panduardi and Endi Sailu Haq, "Wireless Smart Home System Menggunakan Raspberry Pi Berbasis Android," *Jurnal Teknologi Informasi Dan Terapan*, pp. 320-325, 2016.
- [10] Yoyon Efendi, "Internet of Things (IoT) Sistem Pengendalian Lampu Menggunakan Raspberry PI Berbasis Mobile," *Jurnal Ilmiah Ilmu Komputer*, pp. 19-26, 2018.
- [11] Shatadru Bipasha Biswas and M. Tariq Iqbal, "Solar Water Pumping System Control Using a Low Cost ESP32 Microcontroller," in *2018 IEEE Canadian Conference on Electrical & Computer Engineering (CCECE)*, Quebec City, QC, Canada, 2018.
- [12] Anthoinete P.Y.Waroh, "Analisa dan Simulasi Sistem Pengendalian Motor DC," *Jurusan Teknik Elektro Politeknik Negeri Manado*, pp. 81-86, 2014.
- [13] Nalaprana Nugroho and Sri Agustina, "Analisa Motor DC (Direct Current) Sebagai Penggerak Mobil Listrik," *Mikrotiga*, pp. 28-34, 2015.