

PENERAPAN ALGORITMA DIJKSTRA UNTUK PENCARIAN LAHAN PARKIR TERDEKAT PADA SMART PARKING SYSTEM BERBASIS INTERNET OF THINGS

Kharin Ananda Gusti^{1*}, Ilham Ragil Arzufa¹, Chalidia Nurin Hamdani¹

¹Teknik Instrumentasi Kilang, Politeknik Energi dan Mineral Akamigas, Jl Gajah Mada No.38, Kabupaten Blora Jawa Tengah, 58315

**E-mail: kharinananda28@gmail.com*

ABSTRAK

Kurangnya informasi akan ketersediaan parkir dikarenakan padatnya lahan parkir membuat pengguna kendaraan merasa kesusahan, serta pengelola lahan parkir tidak dapat menjamin keamanan penuh terhadap kendaraan dikarenakan kurangnya sistem keamanan. Oleh karena itu, penelitian ini bertujuan untuk menyediakan informasi yang akurat dalam ketersediaan lahan parkir. Metode algoritma dijkstra digunakan dalam penelitian ini untuk mengalamatkan lahan parkir terdekat dari pintu masuk tanpa perlu mengelilingi lahan parkir. Sistem ini, menggunakan sensor RFID untuk mendeteksi kartu parkir sebagai akses masuk dan keluar lahan parkir, dengan penggunaan kartu parkir ini membuat kendaraan lebih terproteksi serta memudahkan pengemudi dalam melakukan pembayaran parkir, dan data dari pembacaan kartu parkir disimpan pada *google sheets database*, serta sensor inframerah digunakan untuk mendeteksi adakah kendaraan yang terparkir pada area kosong. Hasil penelitian pada *smart parking system* ini tidak hanya mengelola sistem keamanan, tapi juga mengelola tata letak penempatan kendaraan pada slot-slot parkir yang tersedia, tidak hanya itu informasi ketersediaan slot parkir juga dapat dilihat secara *offline* pada tampilan layar LCD, dan juga dapat dimonitoring secara online melalui web-server. Melalui uji coba prototipe *smart parking system* ini, diharapkan dapat meningkatkan efisiensi penggunaan lahan parkir, mengurangi antrian kendaraan pada saat mencari parkir, serta memberikan kemudahan dan kenyamanan bagi pengguna parkir.

Kata kunci: Algoritma Dijkstra, *Internet of Things*, *Smart Parkir*, Sensor RFID, Sensor Inframerah

1. PENDAHULUAN

Tempat parkir merupakan sarana prasarana yang sangat dibutuhkan masyarakat pada zaman sekarang dikarenakan banyaknya mobilitas dan meningkatnya pertumbuhan penduduk, maka jumlah penggunaan transportasi juga ikut meningkat terutama penggunaan mobil karena banyak masyarakat lebih memilih kendaraan pribadi dibandingkan transportasi umum [1]. Akibatnya, kebutuhan akan lahan parkir meningkat, namun tidak sebanding dengan lahan parkir yang tersedia [2]. Sebagian besar tempat parkir masih menerapkan sistem manual, sehingga pengemudi kesulitan menemukan tempat parkir yang kosong karena kurangnya informasi dari petugas parkir [3]. Selain itu, dengan mengelilingi lahan parkir juga sangat tidak efektif karena pemborosan waktu bagi pengemudi untuk menemukan lokasi parkir yang belum terisi dan dapat terjadi penumpukan kendaraan di area pintu keluar disebabkan oleh proses pemeriksaan identitas berupa STNK dan pembayaran masih menggunakan cara manual. Jika sistem parkir secara manual digantikan dengan menggunakan *smart* sistem yang dapat langsung menginformasikan lokasi parkir yang belum terisi dan terdekat dari pintu masuk secara langsung

kepada pengendara ketika menggunakan tag RFID, maka proses parkir akan jauh lebih efisien [4]. Dengan *smart parking system* ini, pengendara dapat langsung menuju tempat parkir yang ditunjukkan pada layar display tanpa harus berkeliling mencari tempat yang kosong. Penggunaan *smart* sistem pada lahan parkir juga menjamin keamanan kendaraan yang ditinggalkan di lahan parkir, serta tidak akan ada penumpukan kendaraan di pintu keluar karena pembayaran sudah melalui tag RFID secara *cashless*.

Penggunaan Algoritma Dijkstra pada sistem parkir dalam penelitian ini sangatlah tepat, didukung oleh penelitian sebelumnya yang membahas Penerapan Algoritma Dijkstra Untuk Menentukan Rute Terbaik Pada Mobile E-Parking Berbasis Sistem Informasi Geografis [5], Rancang Bangun *Prototype* Sistem Smart Parking Berbasis Arduino dan Pemantauan Melalui Smartphone [6], Pencarian Jalur terpendek dengan Algoritma Dijkstra [7]. Pada penelitian ini sensor inframerah digunakan sebagai pendeteksi pada lokasi parkir selanjutnya data sensor akan diolah oleh Algoritma Dijkstra untuk menentukan rute terdekat agar mendapatkan lahan parkir yang kosong dan tag RFID sebagai alat pembayaran digital, serta ID tag RFID sebagai sistem keamanan karena ID akan disimpan di database dan dikirim ke google sheets sebagai informasi pada pintu keluar [8]. Layar LCD akan memberikan pengemudi informasi lokasi parkir kosong terdekat di pintu masuk dan informasi harga dan lama waktu parkir di pintu keluar. Protokol komunikasi yang digunakan pada penelitian ini adalah HTTP/HTTPS dan Websocket. Dalam mengintegrasikan ESP32 dengan firebase, HTTPS digunakan untuk mengirimkan data dari ESP32 menuju Firebase dengan koneksi yang aman. Sedangkan Websockets digunakan oleh *Firebase realtime database* untuk menjaga komunikasi yang terus-menerus dalam mengirim data secara realtime sehingga terjadi komunikasi dua arah antara ESP32 dan server Firebase.

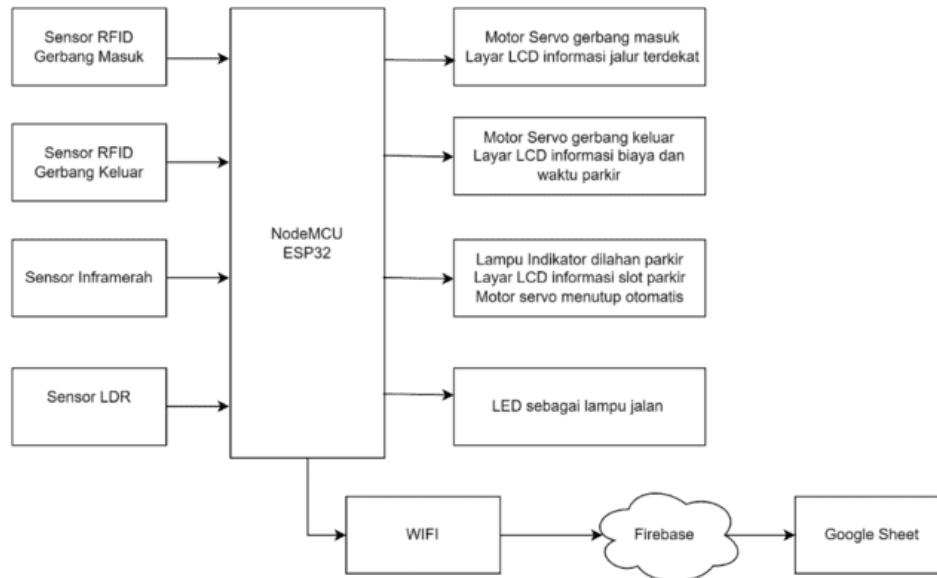
Penelitian ini bertujuan untuk merancang dan mengembangkan *smart parking system* yang dapat memudahkan pengendara dalam menemukan lokasi parkir dengan cepat tanpa perlu berkeliling mencari tempat yang kosong.

2. METODE

Pada penelitian ini, penulis akan membuat smart system parkir menggunakan Algoritma Dijkstra berbasis Internet of Things. Smart parking system yang dikembangkan pada penelitian ini merupakan sebuah prototipe yang terdiri dari beberapa slot parkir dan dilengkapi sensor inframerah. Dalam penelitian ini dilakukan pengujian kinerja Algoritma Dijkstra dalam mencari jalur terpendek menuju slot parkir kosong dan penelitian ini dilengkapi dengan Sensor Inframerah, Sensor RFID, Sensor LDR, Motor Servo, LCD, Database, dan Mikrokontroler yang mengimplementasikan smart system parking. Metode yang diterapkan pada penelitian ini dapat dikonsepkan seperti Gambar 1 berikut.

Pada Gambar 1, perancangan dan pembuatan penelitian ini secara keseluruhan dibutuhkan block diagram yang saling terkait. Kontroler yang digunakan sebagai pusat kerja alat ini berupa NodeMCU ESP32 yang berfungsi untuk menerima dan mengirimkan data hasil pembacaan sensor, kemudian hasil pembacaan akan dikirimkan ke Database dan data akan dikumpulkan dalam bentuk google sheets [9]. Penelitian ini dilengkapi dengan teknologi RFID yang mendeteksi sebuah tag RFID berupa kartu parkir, sensor RFID ini alat yang memancarkan sinyal radio yang digunakan untuk berkomunikasi dengan tag RFID setelah tag menerima sinyal dari sensor RFID maka sensor RFID akan mengirimkan data ke mikrokontroler untuk dilakukan pengelolaan data dan akan memberikan perintah untuk menggerakkan motor servo dan menampilkan informasi hasil algoritma dijkstra di gate masuk, serta informasi biaya dan lama waktu parkir di gate keluar pada layar LCD [10]. Kemudian sensor Inframerah pada sistem ini bertujuan untuk mendeteksi adanya mobil yang melintasi palang pintu atau mengisi lahan parkir kosong dengan memancarkan gelombang inframerah kepada bagian badan mobil, yang

mana hasil dari pendeteksian sensor inframerah yang dipancarkan dari bagian badan mobil akan di kirimkan ke mikrokontroler untuk memberikan perintah pada lampu indikator dan layar LCD untuk memperbarui informasi terkait lahan parkir [11]. Terakhir penggunaan sensor LRD pada penelitian ini hanya untuk mengaktifkan LED yang berperan sebagai lampu jalan secara otomatis ketika cuaca gelap.

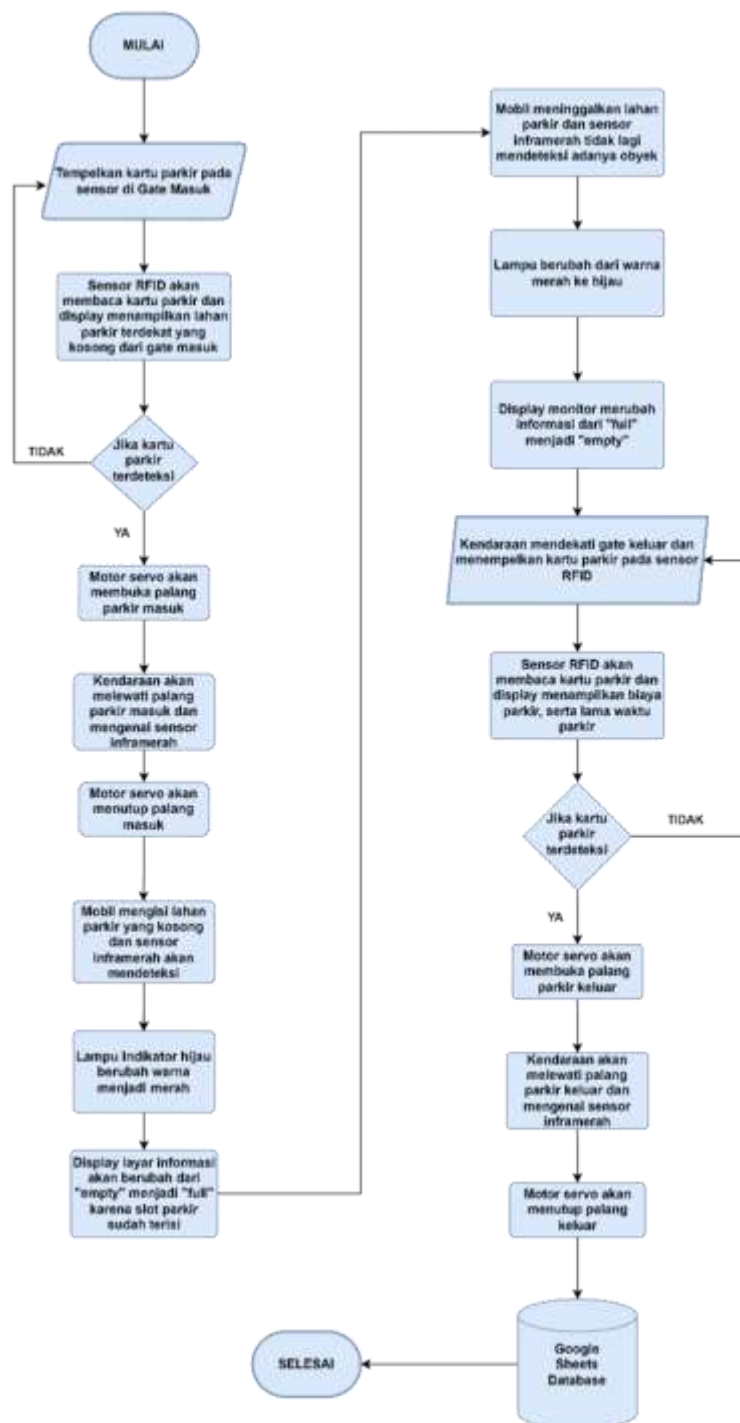


Gambar 1. Block Diagram Sistem

A. Flowchart Sistem

Gambar 2 menunjukkan *flowchart* sistem menjelaskan sistem parkir otomatis menggunakan sensor RFID, motor servo, dan sensor inframerah. Proses dimulai ketika pengguna menempelkan kartu parkir di Gate Masuk. Jika kartu terdeteksi, motor servo membuka palang, kendaraan masuk, dan sensor inframerah mencatat bahwa tempat parkir telah terisi. Lampu indikator berubah dari hijau ke merah, dan layar informasi menampilkan "full". Saat kendaraan ingin keluar, pengguna menempelkan kartu parkir di Gate Keluar, dan sistem menghitung biaya parkir. Setelah pembayaran, palang terbuka, dan kendaraan keluar. Sistem kemudian memperbarui database dan layar kembali menampilkan "empty," menandakan tempat parkir kosong dan siap digunakan.

Sistem parkir ini menggunakan sebuah metode yang mengarahkan atau mengalamatkan calon pengguna parkir pada slot parkir terdekat. Slot parkir akan terisi secara berurutan, dimulai dari slot kosong yang paling dekat dengan pintu masuk, sehingga slot yang kosong dan yang terisi tetap dapat dipetakan dan dinformasikan kepada calon pengguna parkir.

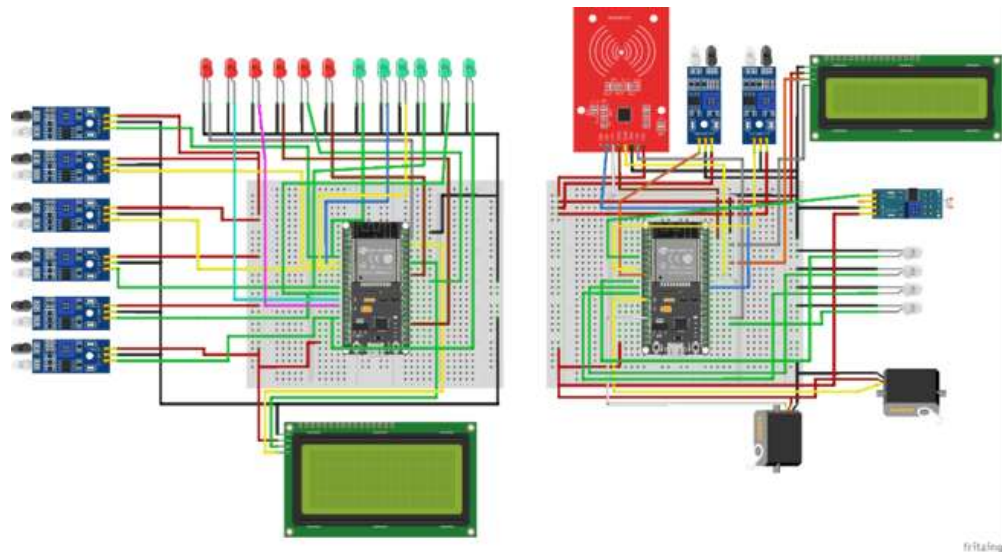


Gambar 2. Flowchart Sistem

B. Rangkaian Sistem

Rangkaian sistem ini untuk mengetahui parameter dalam perancangan hardware dari komponen-komponen yang akan digunakan dengan diuji terlebih dahulu. Rangkaian dari smart sistem parkir diintegrasikan dengan dua NodeMCU ESP32 yang mana hasil pembacaan dan penerimaan mikrokontroler akan dikirim ke *Realtime Database* dalam bentuk Google Sheets, pada gambar juga terlihat beberapa sensor dan komponen yang dibutuhkan dalam rangkaian smart sistem parkir. Dapat dilihat rangkaian sistem penelitian ini pada Gambar 3 berikut.

Berdasarkan Gambar 3 terkait skema rangkaian sistem dapat dijelaskan bahwa penelitian *smart parking system* ini menggunakan dua buah mikrokontroler ESP32 yang diintegrasikan dengan berbagai komponen-komponen sensor. Mikrokontroler ESP32 merupakan sebagai pusat kontrol atau otak dari komponen yang digunakan. ESP32 akan mengatur sistem kerja alat pada prototipe ini dengan memberikan aksi kepada komponen yang digunakan. Komponen-komponen sensor ini untuk merancang kinerja sistem parkir, mulai dari sensor inframerah yang digunakan untuk mendeteksi keberadaan mobil pada slot parkir yang diintegrasikan dengan led untuk menyatakan kondisi slot parkir. Adapun antarmuka RFID sebagai sistem akses masuk dan keluar dari parkir yang diintegrasikan dengan motor servo sebagai buka atau tutup palang pintu otomatis.



Gambar 3 Skema Rangkaian Sistem

Dalam membuat proyek penelitian ini menggunakan NodeMCU ESP32 berbasis *Internet of Things* (IoT). Beberapa komponen yang digunakan dalam pembuatan alat ini sebagai berikut:

1. NodeMCU ESP32

ESP32 adalah modul mikrokontroler yang diperkenalkan oleh Espressif Systems sebagai penerus ESP8266. Dengan fitur dual-mode, yaitu WiFi dan Bluetooth, serta didukung dengan konsumsi daya yang lebih rendah. ESP32 memudahkan pengguna dalam mengembangkan berbagai proyek dan aplikasi berbasis *Internet of Things* (IoT).

2. Sensor RFID (*Radio Frequency Identification*)

RFID adalah sistem yang memanfaatkan frekuensi radio untuk membaca dan mengirimkan informasi dari suatu objek melalui perangkat kecil yang disebut tag. Setiap tag berisi informasi unik berupa nomor seri, yang memungkinkan tag tersebut berkomunikasi dengan sensor RFID ketika mendeteksi sinyal frekuensi dari sensor tersebut.

3. Sensor Inframerah

Sensor Inframerah adalah sensor yang digunakan untuk mendeteksi sebuah objek dengan memanfaatkan pantulan cahaya inframerah [12]. Sensor inframerah akan memancarkan cahaya inframerah, kemudian ketika cahaya yang dipancarkan mengenai suatu objek, sinar akan terpantul kembali dan detektor akan menerima sinyal untuk mendeteksi keberadaan objek.

4. Sensor LDR

Sensor LDR adalah jenis resistor yang peka terhadap cahaya. Nilai resistansi sensor LDR akan mengalami perubahan berdasarkan intensitas cahaya yang diterima. Sensor LDR akan mengubah energi dari foton menjadi elektron.

5. Motor Servo

Motor servo merupakan aktuator elektromekanis yang dimana perputaran secara kontinu tidak dibutuhkan, tetapi motor servo digunakan pada sesuatu yang bergerak ke posisi tertentu dan harus berhenti atau bertahan pada posisi tertentu sesuai dengan perintah yang diberikan.

6. *Light Emitting Diode (LED)*

LED merupakan komponen elektronika yang memancarkan cahaya ketika diberikan tegangan. Konsumsi daya yang dibutuhkan LED tidak terlalu besar, karena hasil intensitas cahaya yang dihasilkan LED tidak termasuk besar.

7. *Liquid Crystal Display (LCD)*

Liquid Crystal Display (LCD) adalah perangkat yang sering digunakan untuk menampilkan data komunikasi antara sistem dan pengguna dalam bentuk teks atau karakter pada berbagai proyek berbasis *Internet of Things* [13].

8. *Firebase*

Firebase adalah *platform realtime database* yang memungkinkan sinkronisasi data di berbagai klien secara cepat melalui cloud. Dengan arsitektur berbeda dari database tradisional, *Firebase* menawarkan solusi andal untuk aplikasi yang membutuhkan penyampaian informasi secara instan.

C. Algoritma Dijkstra

Algoritma Dijkstra, yang ditemukan oleh Edger Wybe Dijkstra, merupakan algoritma greedy yang sering digunakan untuk menyelesaikan masalah pemilihan rute [14]. Algoritma ini digunakan untuk menemukan rute terpendek atau rute paling optimal antara dua titik dengan menghitung nilai minimum dari setiap rute yang tersedia [15]. Dalam penelitian ini, algoritma Dijkstra diterapkan pada *smart parking system* untuk membantu pengemudi menemukan tempat parkir kosong terdekat. Proses penentuan rute terpendek dimulai dengan mengidentifikasi dua titik, yaitu titik awal dan titik tujuan, kemudian membentuk grafik yang mewakili berbagai kemungkinan rute yang dapat diambil, dan menghitung bobot minimum dari setiap titik [16][17]. Variabel dependen dalam penelitian ini adalah waktu tempuh kendaraan dari pintu masuk ke slot parkir yang ditentukan oleh algoritma, sementara variabel independennya adalah jumlah slot parkir yang sudah terisi.

Pada algoritma Dijkstra dalam menentukan jarak terpendek antara dua vertex pada graf berbobot, di mana bobot antar vertex merupakan nilai dari setiap edge dan harus bernilai positif. Algoritma ini bekerja dengan menghitung jarak minimum dari vertex asal ke vertex terdekat, kemudian dilanjutkan ke vertex berikutnya hingga mencapai tujuan [18]. Prosesnya dimulai dengan mengidentifikasi dua titik, yaitu titik awal dan titik tujuan, membentuk graf yang mewakili berbagai kemungkinan rute, dan menghitung bobot minimum dari setiap titik.

Langkah pertama yang harus dilakukan adalah membandingkan setiap titik yang akan dilalui. Setiap titik memiliki bobot yang berbeda, sehingga rute yang diambil akan menghasilkan nilai yang berbeda-beda [19].

$$Ke\ Titik\ Tujuan = Titik1 \rightarrow Titik2 \rightarrow Titik3 \rightarrow \dots \rightarrow Titik_n \quad (1)$$

Setelah menentukan rute dari titik awal hingga titik tujuan, peneliti akan menjumlahkan bobot setiap titik yang dilalui [20].

$$Ke\ Titik\ Tujuan = Titik1 + Titik2 + Titik3 + \dots + Titik_n \quad (2)$$

Hasil penjumlahan setiap rute yang paling minimum merupakan hasil dari pencarian rute terdekat yang dicari [20]. Pada Gambar 4, setiap perpindahan antar titik memiliki nilai jarak, dan langkah pertama yang harus dilakukan adalah menghitung jarak pada setiap simpul dari semua kemungkinan rute. Dalam contoh ini, terdapat tiga rute yang bisa dilalui oleh kendaraan. Rute Pertama = Titik 0 → Titik 1 → Titik 2 → Titik 3 → Titik 4 → Titik 5 → Titik 7.

Rute Kedua = Titik 0 → Titik 6 → Titik 10 → Titik 11 → Titik 7.

Rute Ketiga = Titik 0 → Titik 6 → Titik 8 → Titik 9 → Titik 10 → Titik 11 → Titik 7.

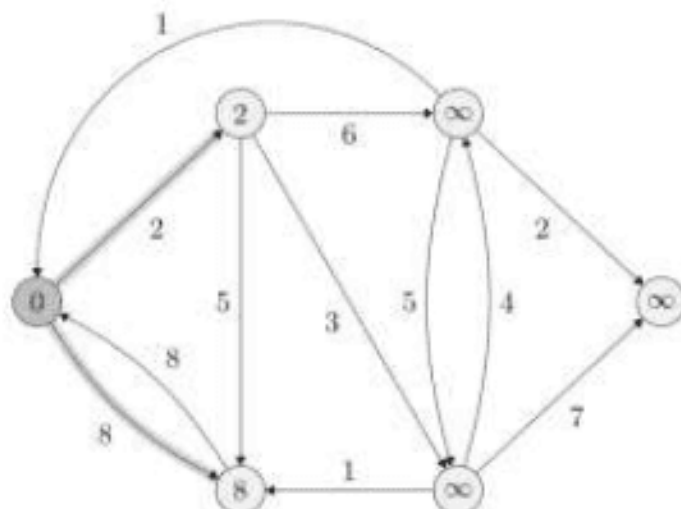
Setelah itu, jarak total dari masing-masing rute dihitung dalam satuan meter:

Rute Pertama = $420 + 450 + 1200 + 4500 + 120 + 160 = 6850$ meter (6.850 km).

Rute Kedua = $420 + 190 + 3900 + 3500 + 180 = 8190$ meter (8.190 km).

Rute Ketiga = $420 + 190 + 1676 + 2330 + 3900 + 180 = 8696$ meter (8.696 km).

Berdasarkan contoh kasus di atas, dapat disimpulkan bahwa rute pertama adalah rute terdekat, karena memiliki jarak yang paling dekat untuk mencapai titik tujuan.

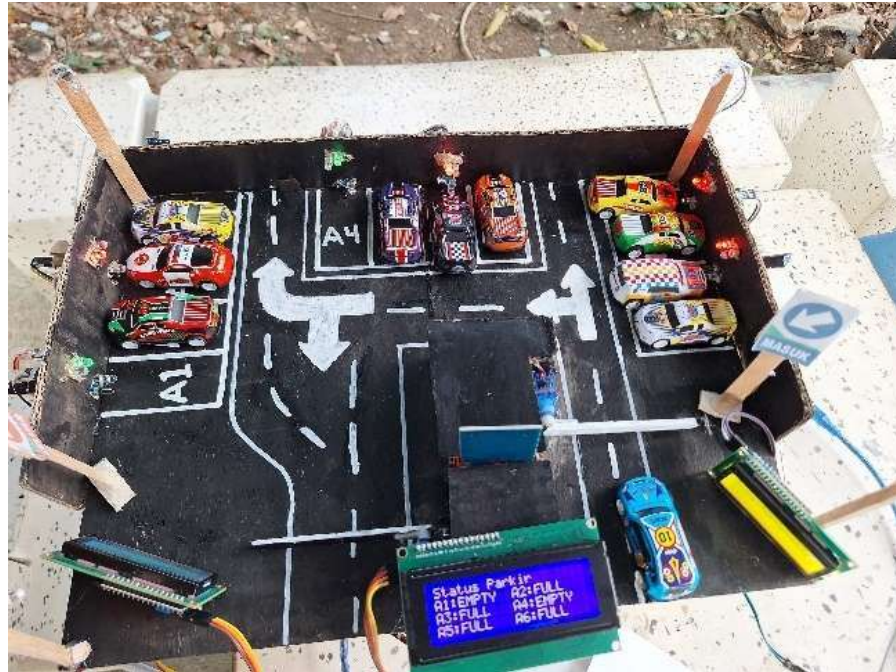


Gambar 4 Rute Algoritma Dijkstra

3. PEMBAHASAN

A. Desain Prototipe Keseluruhan

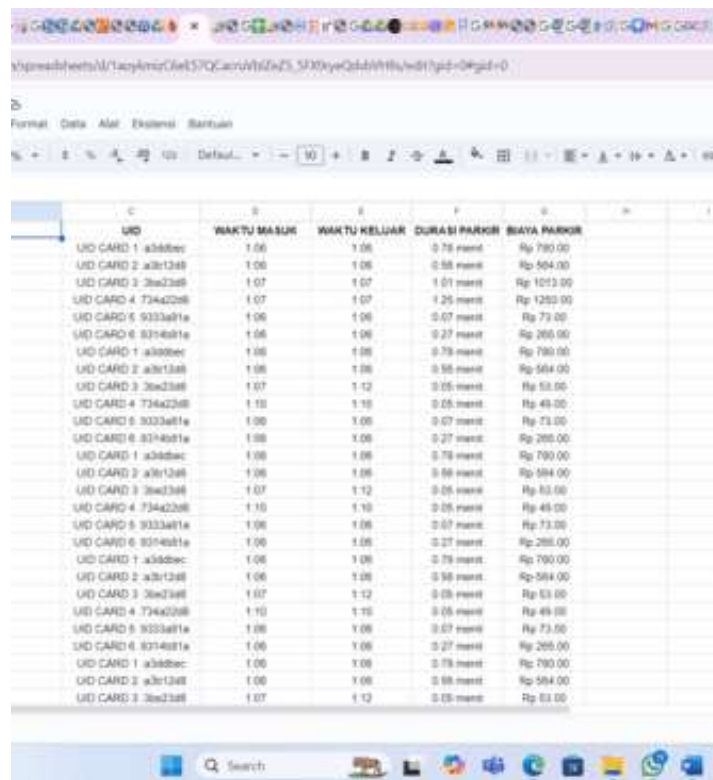
Setelah melakukan perancangan sistem dan juga skema *wiring* sistem, selanjutnya pembuatan rancangan *hardware*. Gambar 5 adalah desain prototipe *smart parking system* yang sudah dirancang. Gambar 5 menjelaskan bahwa saat mobil akan memasuki area parkir, pengemudi harus terlebih dahulu menempelkan kartu parkir. Setelah itu, palang pintu akan terbuka secara otomatis, dan layar LCD akan menampilkan informasi tentang slot parkir terdekat yang kosong. Setelah kendaraan melewati sensor inframerah, palang pintu akan tertutup kembali, dan mobil dapat melanjutkan ke area parkir yang kosong.



Gambar 5 Desain Prototipe

B. Pengujian Alat

Pengujian dilakukan dengan memeriksa fungsi setiap sensor dan display yang digunakan, kemudian dilanjutkan dengan pengecekan wiring rangkaian. Pada tahap ini, akan dilakukan *troubleshooting* untuk mengidentifikasi kesalahan atau error yang mungkin terjadi. Pengujian yang dilakukan peneliti berupa menguji arah peputaran motor servo. Ketika sensor RFID membaca tag RFID pada gerbang masuk dan keluar maka motor servo akan berputar sebanyak beberapa derajat untuk membuka palang pintu. Dan pengujian sensor Inframerah dalam mendeteksi adanya mobil. Setelah rangkaian dan sensor sudah berjalan dengan baik maka komponen-komponen sudah dapat di tempatkan sesuai rancangan alat yang telah dibuat. Selanjutnya proses monitoring melalui *google sheets database* terdapat 4 hal yang dimonitoring yaitu, biaya parkir, durasi parkir, waktu keluar, dan waktu masuk kendaraan. Adapun tampilan monitoring Google Sheet seperti Gambar 6 berikut.



UID	WAKTU MASUK	WAKTU KELUAR	DURASI PARKIR	BIAYA PARKIR
UID CARD 1 a3d8bec	1:06	1:06	0:76 menit	Rp 780.00
UID CARD 2 a3e12a8	1:06	1:06	0:58 menit	Rp 504.00
UID CARD 3 36a23a8	1:07	1:07	1:01 menit	Rp 1013.00
UID CARD 4 734a22a8	1:07	1:07	1:26 menit	Rp 1282.00
UID CARD 5 9333a81e	1:06	1:06	0:07 menit	Rp 73.00
UID CARD 6 8314a81e	1:06	1:06	0:27 menit	Rp 266.00
UID CARD 1 a3d8bec	1:06	1:06	0:76 menit	Rp 780.00
UID CARD 2 a3e12a8	1:06	1:06	0:58 menit	Rp 504.00
UID CARD 3 36a23a8	1:07	1:12	0:05 menit	Rp 53.00
UID CARD 4 734a22a8	1:10	1:10	0:05 menit	Rp 49.00
UID CARD 5 9333a81e	1:06	1:06	0:07 menit	Rp 73.00
UID CARD 6 8314a81e	1:06	1:06	0:27 menit	Rp 266.00
UID CARD 1 a3d8bec	1:06	1:06	0:76 menit	Rp 780.00
UID CARD 2 a3e12a8	1:06	1:06	0:58 menit	Rp 504.00
UID CARD 3 36a23a8	1:07	1:12	0:05 menit	Rp 53.00
UID CARD 4 734a22a8	1:10	1:10	0:05 menit	Rp 49.00
UID CARD 5 9333a81e	1:06	1:06	0:07 menit	Rp 73.00
UID CARD 6 8314a81e	1:06	1:06	0:27 menit	Rp 266.00
UID CARD 1 a3d8bec	1:06	1:06	0:76 menit	Rp 780.00
UID CARD 2 a3e12a8	1:06	1:06	0:58 menit	Rp 504.00
UID CARD 3 36a23a8	1:07	1:12	0:05 menit	Rp 53.00
UID CARD 4 734a22a8	1:10	1:10	0:05 menit	Rp 49.00
UID CARD 5 9333a81e	1:06	1:06	0:07 menit	Rp 73.00
UID CARD 6 8314a81e	1:06	1:06	0:27 menit	Rp 266.00
UID CARD 1 a3d8bec	1:06	1:06	0:76 menit	Rp 780.00
UID CARD 2 a3e12a8	1:06	1:06	0:58 menit	Rp 504.00
UID CARD 3 36a23a8	1:07	1:12	0:05 menit	Rp 53.00

Gambar 6. Tampilan Google Sheets

Pada pengujian prototipe ini area parkir memiliki 2 lahan parkir kosong dan ketika pengendara menempelkan kartu parkir pada RFID, maka algoritma dijkstra akan bekerja untuk mendapatkan lahan parkir mana yang terdekat dari pintu masuk dan didapatkanlah bahwa lahan parkir A4 adalah yang terdekat dari pintu masuk, kemudian informasi itu akan di tampilkan pada display LCD untuk diberitahukan pada pengendara. Dapat dilihat pada Gambar 7 berikut.



Gambar 7 Pengujian Hardware dengan Algoritma Dijkstra

Ketika kendaraan sudah mengisi lahan parkir yang diberitahukan oleh display lcd di gate masuk dari hasil pencarian algoritma dijkstra, maka sensor inframerah yang berada di lahan parkir A4 akan mendeteksi adanya kendaraan yang mengisi lahan parkir tersebut, lalu akan mengubah indikator lampu berwarna hijau yang menandakan bahwa lahan parkir itu kosong

berubah menjadi warna merah karena lahan parkir tersebut telah terisi dan terjadi perubahan juga pada display LCD yang menginformasikan semua status lahan parkir, bahwa lahan parkir A4 yang sebelumnya statusnya EMPTY menjadi status FULL. Perubahan yang terjadi pada LCD akan terus update berdasarkan perubahan yang terjadi secara realtime. Dapat dilihat pada Gambar 8 berikut.



Gambar 8 Perubahan Tampilan Kondisi Slot Parkir

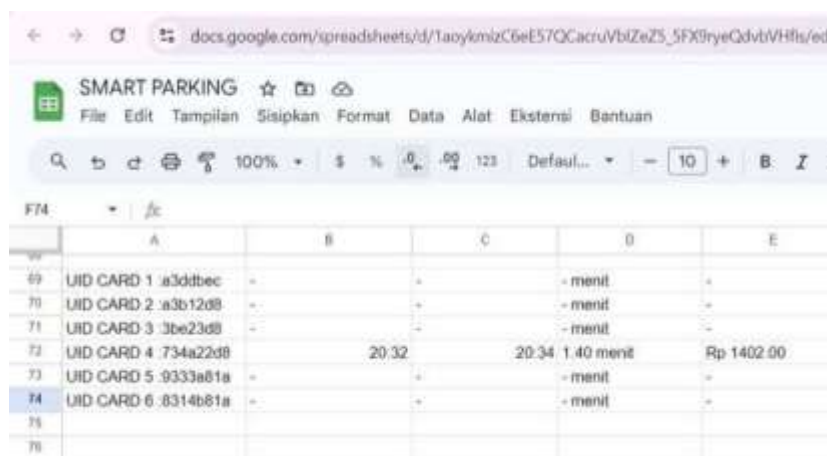
Pada saat kendaraan akan meninggalkan lahan parkir, pengendara harus menempelkan kembali kartu parkir pada sensor RFID, dan ketika sensor RFID sudah membaca kartu parkir maka palang keluar otomatis terbuka dan display LCD akan menampilkan durasi dan biaya parkir dari kendaraan tersebut. Penerapan ini dapat dilihat pada Gambar 9 berikut.



Gambar 9 Tampilan Biaya Parkir pada LCD

Dilakukan pengujian apakah lama waktu parkir mobil pada tampilan layar LCD sesuai dengan yang berapa pada *google sheets database* dan perhitungan *stopwatch*. Penguji melakukan perhitungan waktu dengan stopwatch dimulai ketika kartu parkir bersentuhan dengan

sensor parkir hingga kartu parkir kembali bersentuhan dengan sensor RFID pada gate keluar. Terlihat dari Gambar 10 hasil dari pembacaan kartu parkir yang terdata pada google sheets database bahwa pengendara masuk kelahan parkir di jam 20:32 dan keluar dari lahan parkir di jam 20:34, pengendara hanya berada pada lahan parkir selama 1 menit 40 detik.



	A	B	C	D	E
69	UID CARD 1 :a3ddbdc	-	-	- menit	-
70	UID CARD 2 :a3b12d8	-	-	- menit	-
71	UID CARD 3 :3be23d8	-	-	- menit	-
72	UID CARD 4 :734a22d8	20:32	20:34	1:40 menit	Rp 1402.00
73	UID CARD 5 :9333a81a	-	-	- menit	-
74	UID CARD 6 :8314b81a	-	-	- menit	-
75					
76					

Gambar 10 Tampilan Durasi Waktu Pada Google Sheets

Dapat dilihat pada Gambar 11 merupakan hasil pengujian durasi sistem alat kerja dengan *stopwatch*. Percobaan ini melakukan perbandingan antara nilai variabel yang dihasilkan oleh prototipe dengan *stopwatch* pada *smartphone*. Hasil perbandingan yang didapatkan dari pengujian durasi waktu antara prototipe dengan *stopwatch* memiliki perbedaan yang tidak jauh. Sedangkan pada Gambar 12 tampak hasil tampilan slot parkir melalui web server, tampilan ini untuk mengetahui ketersediaan slot parkir yang ditampilkan secara online. Tampilan web server ini dapat memudahkan pengguna parkir untuk mengetahui keberadaan slot parkir yang tersedia.



Gambar 11. Tampilan Durasi Waktu pada LCD dan Stopwatch

Rute 3: $20\text{ cm} + 29,5\text{ cm} + 30\text{ cm} = 79,5\text{ cm}$

Rute 4: $20\text{ cm} + 29,5\text{ cm} + 44\text{ cm} = 93,5\text{ cm}$

Rute 5: $20\text{ cm} + 29,5\text{ cm} + 30\text{ cm} + 29,5\text{ cm} = 109\text{ cm}$

Rute 6: $20\text{ cm} + 29,5\text{ cm} + 30\text{ cm} + 27,5\text{ cm} + 27,5 = 134,5\text{ cm}$

Berdasarkan perhitungan yang telah dilakukan didapatkan rute tercepat yaitu dari gate langsung menuju slot A4 tanpa melalui slot-slot lainnya. Dari antara 6 rute, rute tercepat di dapatkan pada rute 1 sebesar 37,5 cm lebih pendek dibandingkan jarak rute lain.

4. SIMPULAN

Dari hasil penelitian ini, prototipe yang dirancang untuk mengimplementasikan Algoritma Dijkstra pada smart parking system berbasis Internet of Things telah sesuai dengan desain yang ditentukan berdasarkan diagram alir. Semua komponen, termasuk sensor RFID, sensor inframerah, sistem kendali, dan layar LCD, berfungsi dengan baik setelah melalui proses troubleshooting yang intensif. Hasil uji coba menunjukkan bahwa prototipe dapat mendeteksi kendaraan dengan akurat, memberikan informasi yang tepat mengenai lahan parkir terdekat, serta memperbarui informasi secara real-time melalui lampu indikator dan layar LCD. Keberhasilan uji coba ini menunjukkan bahwa seluruh alur proses, dari deteksi hingga penyimpanan data ke Google Sheets, berjalan sesuai harapan dan hasil perbandingan durasi waktu yang telah diujikan pada prototipe juga memiliki perbedaan yang sangat sedikit hanya berbeda sepersekian detik dengan hasil perhitungan stopwatch. Namun, terdapat kekurangan berupa waktu delay yang lama pada motor servo dan akses webserver yang terbatas pada jaringan lokal. Oleh karena itu, disarankan untuk melakukan penelitian lebih lanjut guna mengatasi masalah keterlambatan ini dan meningkatkan aksesibilitas sistem agar dapat diakses secara lebih luas.

5. DAFTAR PUSTAKA

- [1] K. P. R. INDONESIA, "MENGUPAS PERPARKIRAN DI PERKOTAAN Kementerian Perhubungan Republik Indonesia." Accessed: Oct. 14, 2024. [Online]. Available: <https://dephub.go.id/post/read/mengupas-perparkiran-di-perkotaan-3094>
- [2] A. A. Batabyal and P. Nijkamp, "A probabilistic analysis of two university parking issues," *Ann. Reg. Sci.*, vol. 44, no. 1, pp. 111–120, Jan. 2009, doi: 10.1007/S00168-008-0237-8.
- [3] R. Kurniawan and A. Zulus, "Sistem Smart Parking Menggunakan Ultrasonik Sensor," *J. Sist. Komput. Musirawas*, vol. 3, no. 1, p. 22, 2018, doi: 10.32767/jusikom.v3i1.309.
- [4] R. Winarsih, I. and Mahendra, "Sistem Parkir Otomatis Menggunakan Rfid Berbasis Mikrokontroler AT 89S51 | PDF." Accessed: Oct. 14, 2024. [Online]. Available: <https://id.scribd.com/doc/264521487/79-234-1-PB>
- [5] C. Prianto and M. Kusnadi, "Penerapan Algoritma Dijkstra Untuk Menentukan Rute Terbaik Pada Mobile E-Parking Berbasis Sistem Informasi Geografis," *J. Inform. J. Pengemb. IT*, vol. 3, no. 3, pp. 329–335, Oct. 2018, doi: 10.30591/JPIT.V3I3.941.
- [6] R. Rudi, I. Dinata, and R. Kurniawan, "RANCANG BANGUN PROTOTYPE SISTEM SMART PARKING BERBASIS ARDUINO DAN PEMANTAUAN MELALUI SMARTPHONE," *J. ECOTIPE*, vol. 4, no. 2, pp. 14–20, Oct. 2017, doi: 10.33019/ECOTIPE.V4I2.7.
- [7] M. K. Harahap and N. Khairina, "Pencarian Jalur Terpendek dengan Algoritma Dijkstra," *Sinkron*, vol. 2, no. 2, p. 18, Oct. 2017, doi: 10.33395/SINKRON.V2I2.61.
- [8] T. S. P. baharsyah Gino, "SMART PARKING BASEMENT UNIFA MENGGUNAKAN RFID PADA KARTU MAHASISWA BERBASIS MIKROKONTROLER," *Sustain.*, vol. 11, no. 1, pp. 1–78, 2019.
- [9] S. Keputusan Dirjen Penguatan Riset dan Pengembangan Ristek Dikti and B. Rahman, "Control System prototype Smart Home IoT based with MQTT method using Google Assistant," *J. RESTI*

- (*Rekayasa Sist. dan Teknol. Informasi*), vol. 4, no. 2, pp. 303–310, Apr. 2020, doi: 10.29207/RESTI.V4I2.1721.
- [10] N. Maghfirah, “Pengimplementasian RFID dalam Perkembangan Teknologi.” Accessed: Oct. 12, 2024. [Online]. Available: http://reslab.sk.fti.unand.ac.id/index.php?option=com_k2&view=item&id=245:pengimplementasian-rfid-dalam-perkembangan-teknologi&Itemid=342
 - [11] E. Sunandar *et al.*, “PROTOTYPE MONITORING AREA PARKIR MOBIL BERBASIS ARDUINO UNO UNTUK MENDETEKSI KETERSEDIAAN SLOT PARKIR SECARA OTOMATIS,” vol. 10, no. 1, 2017.
 - [12] S. Sadi, “Rancang Bangun Sistem Eskalator Otomatis Menggunakan Sensor Photodioda Dan Infrared (Ir) Berbasis Mikrokontroler Atmega32,” *J. Din. UMT*, vol. 1, no. 1, p. 71, 2015, doi: 10.31000/dinamika.v1i1.511.
 - [13] M. Yusuf, M. Yusuf, I. Isnawaty, and R. Ramadhan, “IMPLEMENTASI ROBOT LINE FOLLOWER PENYIRAM TANAMAN OTOMATIS MENGGUNAKAN METODE PROPORTIONAL–INTEGRAL–DERIVATIVE CONTROLLER (PID),” *Semant. Tek. Inf.*, vol. 2, no. 1, Apr. 2016, doi: 10.55679/semantik.v2i1.724.
 - [14] I. Arthalia Wulandari and P. Sukmasetyan, “Implementasi Algoritma Dijkstra untuk Menentukan Rute Terpendek Menuju Pelayanan Kesehatan,” *J. Ilm. Sist. Inf.*, vol. 1, no. 1, pp. 30–37, 2022, doi: 10.24127/jisi.v1i1.1953.
 - [15] E. P. Aprilianingsih, R. Primananda, and A. Suharsono, “Analisis Fail Path Pada Arsitektur Software Defined Network Menggunakan Dijkstra Algorithm,” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 1, no. 3, pp. 174–183, May 2017, Accessed: Oct. 12, 2024. [Online]. Available: <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/59>
 - [16] D. B. Johnson, “A Note on Dijkstra’s Shortest Path Algorithm,” *J. ACM*, vol. 20, no. 3, pp. 385–388, Jul. 1973, doi: 10.1145/321765.321768.
 - [17] B. Golden, “Technical Note—Shortest-Path Algorithms: A Comparison,” <https://doi.org/10.1287/opre.24.6.1164>, vol. 24, no. 6, pp. 1164–1168, Dec. 1976, doi: 10.1287/OPRE.24.6.1164.
 - [18] M. F. Rkt, “Sistem informasi geografis pariwisata kota medan menggunakan metode Algoritma Dijkstra,” 2021.
 - [19] J. Talunohi, “Implementasi Algoritma Dijkstra Untuk Menentukan Rute Terpendek Ke Wisata Pantai Pulau Nias,” 2023.
 - [20] Bunaen Chatrin, Pratiwi Hanna, and Riti Yosefina, “Penerapan Algoritma Dijkstra Untuk Menentukan Rute Terpendek Dari Pusat Kota Surabaya Ke Tempat Bersejarah,” *J. Teknol. Dan Sist. Inf. Bisnis*, vol. 4, no. 1, pp. 213–223, 2022.