

# **RANCANG BANGUN SISTEM MONITORING KETERSEDIAAN TEMPAT PARKIR MENGUNAKAN ARDUINO DAN SENSOR ULTRASONIK**

**SKRIPSI**

Diajukan untuk Memenuhi Tugas Akhir dan Melengkapi  
Syarat Guna Memperoleh Gelar Sarjana Strata Satu (S-1)  
dalam Teknologi Informasi



Diajukan oleh:

**FARIS NUR RAHMAN**

NIM : 1908096012

**PROGRAM STUDI TEKNOLOGI INFORMASI  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI WALISONGO SEMARANG  
2023**



# PERNYATAAN KEASLIAN

## PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini :

Nama : **Faris Nur Rahman**

NIM : 1908096012

Jurusan : Teknologi Informasi

Menyatakan bahwa skripsi yang berjudul :

**RANCANG BANGUN SISTEM MONITORING KETERSEDIAAN  
TEMPAT PARKIR MENGGUNAKAN ARDUINO DAN SENSOR  
ULTRASONIK**

Secara keseluruhan adalah hasil penelitian / karya saya sendiri, kecuali bagian tertentu yang dirujuk sumbernya.

Semarang, 5 Oktober 2023

Pembuat Pernyataan,



**Faris Nur Rahman**

NIM. 1908096012



# LEMBAR PENGESAHAN



KEMENTERIAN AGAMA  
UNIVERSITAS ISLAM NEGERI WALISONGO  
FAKULTAS SAINS DAN TEKNOLOGI  
Jl. Prof. Dr. Hamka Ngalyan Semarang  
Telp. 024-7601295 Fax.7615387

## LEMBAR PENGESAHAN

Naskah proposal berikut ini :

Judul : RANCANG BANGUN SISTEM MONITORING KETERSEDIAAN TEMPAT  
PARKIR MENGGUNAKAN ARDUINO DAN SENSOR ULTRASONIK

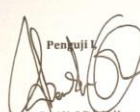
Penulis : Faris Nur Rahman


NIM : 1908096012

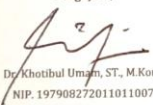
Jurusan : Teknologi Informasi


Telah diujikan dalam sidang *munaqasyah* oleh Dewan Penguji Fakultas Sains dan Teknologi UIN Walisongo Semarang dan dapat diterima sebagai salah satu syarat memperoleh gelar sarjana dalam Teknologi Informasi.

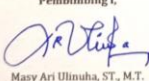
Semarang, 2 Oktober 2023


Penguji I,  
  
Wendy Daji Y, S.Pd., M.Kom.  
NIP. 197706222006042005

DEWAN PENGUJI  
Penguji II,  
  
Hery Mustofa, M.Kom.  
NIP. 198703172019031007

Penguji III,  
  
Dr. Khotibul Umam, ST., M.Kom.  
NIP. 197908272011011007

Penguji IV,  
  
Siti Nur Aini, S.Kom., M.Kom.  
NIP. 198401312018012001

Pembimbing I,  
  
Masy Ari Ulinuha, ST., M.T.  
NIP. 198108122011011007

Pembimbing II,  
  
Hery Mustofa, M.Kom.  
NIP. 198703172019031007

## NOTA DINAS

Semarang, 16 September 2023

Yth. Ketua Program Studi Teknologi Informasi

Fakultas Sains dan Teknologi

UIN Walisongo Semarang

Assalamu'alaikum wr. wb.

Dengan ini diberitahukan bahwa saya telah melakukan bimbingan, arahan dan koreksi naskah skripsi dengan;

Judul : RANCANG BANGUN SISTEM MONITORING  
KETERSEDIAAN TEMPAT PARKIR MENGGUNAKAN ARDUINO  
DAN SENSOR ULTRASONIK

Penulis : Faris Nur Rahman

NIM : 1908096012

Jurusan : Teknologi Informasi

Saya memandang bahwa naskah skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo untuk diajukan dalam Sidang Munaqosah.

Wassalamu'alaikum wr.wb.

Pembimbing I,



Masy Ari Ulinuha, ST., M.T

NIP. 19810812 201101 1007



## NOTA DINAS

Semarang, 18 September 2023

Yth. Ketua Program Studi Teknologi Informasi

Fakultas Sains dan Teknologi

UIN Walisongo Semarang

Assalamu'alaikum wr. wb.

Dengan ini diberitahukan bahwa saya telah melakukan bimbingan, arahan dan koreksi naskah skripsi dengan;

Judul : RANCANG BANGUN SISTEM MONITORING  
KETERSEDIAAN TEMPAT PARKIR MENGGUNAKAN ARDUINO  
DAN SENSOR ULTRASONIK

Penulis : Faris Nur Rahman

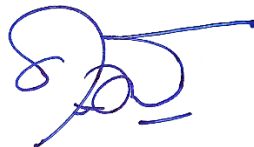
NIM : 1908096012

Jurusan : Teknologi Informasi

Saya memandang bahwa naskah skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo untuk diajukan dalam Sidang Munaqosah.

Wassalamu'alaikum wr.wb.

Pembimbing II,



Hery Mustofa, M.Kom.

NIP. 19870317 201903 1 007





## ABSTRAK

Penelitian ini fokus pada merancang dan membangun sistem monitoring ketersediaan tempat parkir yang inovatif. Tujuan utama penelitian ini adalah untuk menciptakan sebuah sistem yang mampu secara efektif dan akurat mengidentifikasi ketersediaan slot parkir dalam waktu nyata. Sistem ini beroperasi dengan memasang sensor ultrasonik di atas setiap slot parkir untuk mendeteksi status ketersediaannya. Ketika kendaraan bergerak masuk atau keluar dari slot parkir, sensor tersebut secara otomatis mengukur jarak antara kendaraan dan sensor. Hasil pengukuran ini digunakan untuk menentukan apakah suatu slot parkir kosong atau sudah terisi. Informasi status ketersediaan slot parkir kemudian ditampilkan secara real-time melalui lampu LED yang terletak di atas masing-masing slot parkir dan juga melalui tampilan *web*.

Selain itu, penelitian ini juga akan mengevaluasi performa kinerja sistem yang telah dikembangkan. Hal ini mencakup analisis akurasi dalam mendeteksi ketersediaan tempat parkir, respons waktu sistem terhadap perubahan status parkir, serta kemampuan sistem dalam menangani volume kendaraan yang tinggi. Hasilnya, peneliti mendapatkan data bahwa performa kinerja sistem monitoring ini mencapai tingkat keberhasilan sempurna pada pengujian sistem monitoring, yaitu sebesar 100%. Penerapan sistem monitoring ini diharapkan akan mempermudah pengemudi dalam menemukan slot parkir yang kosong dan mengurangi waktu yang terbuang dalam mencari tempat parkir yang sesuai. Dengan demikian, diharapkan sistem ini dapat meningkatkan efisiensi penggunaan lahan parkir dan memberikan pengalaman parkir yang lebih baik bagi pengguna lahan parkir.

**Kata Kunci** : Arduino, NodeMCU ESP8266, sensor ultrasonik HC-SR04, Monitoring Slot Parkir



## ABSTRACT

This research focuses on designing and building an innovative parking space availability monitoring system. The main objective of this research is to create a system that is able to effectively and accurately identify the availability of parking slots in real time. This system operates by installing ultrasonic sensors above each parking slot to detect its availability status. When a vehicle moves into or out of a parking slot, the sensor automatically measures the distance between the vehicle and the sensor. The results of this measurement are used to determine whether a parking slot is empty or filled. Information on the status of parking slot availability is then displayed in real-time via LED lights located above each parking slot and also via a web display.

Apart from that, this research will also evaluate the performance of the system that has been developed. This includes analysis of accuracy in detecting parking space availability, the system's response time to changes in parking status, and the system's ability to handle high vehicle volumes. As a result, researchers obtained data that the performance of this monitoring system achieved a perfect success rate in testing the monitoring system, namely 100%. It is hoped that the implementation of this monitoring system will make it easier for motorists to find empty parking slots and reduce the time wasted in looking for a suitable parking space. Thus, it is hoped that this system can increase the efficiency of parking space use and provide a better parking experience for parking space users.

**Keywords:** Arduino, NodeMCU ESP8266, HC-SR04 ultrasonic sensor, Parking Slot Monitoring



## KATA PENGANTAR

Puji dan syukur saya panjatkan kepada Allah SWT. atas ridhonya saya dapat menyelesaikan penyusunan skripsi ini. Adapun judul skripsi yang saya ajukan adalah “Sistem Monitoring Ketersediaan Tempat Parkir Menggunakan Arduino dan Sensor Ultrasonik”

Tidak dapat disangkal bahwa butuh usaha yang keras dalam penyelesaian pengerjaan skripsi ini. Namun, karya ini tidak akan selesai tanpa orang-orang tercinta di sekeliling saya yang mendukung dan membantu. Terima kasih saya sampaikan kepada:

1. Bapak Prof. Dr. Imam Taufiq, M.Ag. selaku Rektor Universitas Islam Negeri Walisongo Semarang.
2. Bapak Dr. H. Ismail, M.Ag, selaku Dekan Fakultas Teknologi Informasi Universitas Islam Negeri Walisongo Semarang.
3. Bapak Nur Cahyo Hendro Wibowo, S.T., M.Kom, selaku ketua program studi Teknologi Informasi Universitas Islam Negeri Walisongo Semarang.
4. Bapak Masy Ari Ulinuha, ST., M.T, dan Bapak Hery Mustofa, M.Kom. selaku dosen pembimbing skripsi saya yang dengan sabar telah meluangkan waktu untuk memberikan arahan yang sangat berguna bagi penulis sehingga skripsi ini dapat terselesaikan dengan baik.

5. Staff, karyawan dan dosen di lingkungan Universitas Islam Negeri Walisongo Semarang.
6. Orang tua tercinta dan keluarga yang sudah memberi dukungan secara moril maupun materil serta kasih sayang yang tak terbatas.
7. Abdur Rofi Maulidin, Roman Fauzan, Jhorgi Arya Pratama, Zainab Hauro Aisyah, dan Vidya Setianti Wulandari yang sudah membantu penelitian dan memberikan dukungan hingga skripsi ini terselesaikan.
8. Teman-teman Teknologi Informasi yang selalu memberi dukungan serta semua pihak yang telah membantu dan tidak dapat disebutkan satu persatu.
9. *Lastly, I want to thank myself. I want to thank myself for believing in me. I want to thank myself for doing all this hard work. I want to thank myself for not having a day off.*

Semoga segala kebaikan dan pertolongan semuanya mendapat berkah dari Allah SWT. dan akhirnya saya menyadari bahwa skripsi ini masih jauh dari kata sempurna, karena keterbatasan ilmu yang saya miliki. Untuk itu saya dengan kerendahan hati mengharapkan saran dan kritik yang sifatnya membangun dari semua pihak demi membangun laporan penelitian ini.

## DAFTAR ISI

PERNYATAAN KEASLIAN .....	iii
LEMBAR PENGESAHAN.....	v
NOTA DINAS.....	vi
NOTA DINAS.....	viii
ABSTRAK .....	x
KATA PENGANTAR.....	xiv
DAFTAR ISI .....	xvi
DAFTAR TABEL.....	xx
DAFTAR GAMBAR.....	xxii
DAFTAR LAMPIRAN.....	xxvi
BAB I.....	1
A.    Latar Belakang.....	1
B.    Rumusan Masalah .....	3
C.    Tujuan Penelitian.....	4
D.    Batasan Masalah .....	4
E.    Manfaat Penelitian .....	4
BAB II .....	7



A.	Kajian Pustaka .....	7
1.	Parkir .....	7
2.	Jenis Parkir .....	7
3.	Arduino Uno.....	10
4.	Arduino IDE.....	12
5.	Firebase .....	13
6.	Sensor Ultrasonik HC-SR04.....	15
7.	Light Emiting Diode (LED).....	18
8.	Kabel Jumper .....	20
9.	NodeMCU Esp8266 .....	21
B.	Kajian Penelitian yang Relevan.....	22
BAB III.....		27
A.	Model Pengembangan .....	27
B.	Prosedur Pengembangan .....	29
BAB IV .....		51
A.	Hasil Analisis .....	51
B.	Hasil Desain .....	52
C.	Hasil Pengembangan.....	53
D.	Hasil Implementasi .....	56
E.	Hasil Evaluasi .....	57

F.	Rekapitulasi Hasil Pengujian .....	75
BAB V.....		79
A.	Kesimpulan .....	79
B.	Saran.....	80
DAFTAR PUSTAKA.....		81
LAMPIRAN .....		89



## DAFTAR TABEL

Tabel 2. 1 Penelitian yang relevan .....	22
Tabel 4. 1 Hasil Uji Sensor 1. ....	57
Tabel 4. 2 Hasil Uji Sensor 2. ....	58
Tabel 4. 3 Hasil Uji Sensor 3. ....	59
Tabel 4. 4 Hasil Uji Sensor 4. ....	60
Tabel 4. 5 Hasil pengujian dengan satu objek.....	75
Tabel 4. 6 Hasil pengujian dengan dua objek.....	76
Tabel 4. 7 Hasil pengujian dengan tiga objek.....	76
Tabel 4. 8 Hasil pengujian dengan empat objek.....	77



## DAFTAR GAMBAR

Gambar 2. 1 Model-model pola parkir .....	10
Gambar 2. 2 Arduino UNO .....	11
Gambar 2. 3 Arduino IDE .....	13
Gambar 2. 4 Firebase Database.....	14
Gambar 2. 5 Sensor Ultrasonik HC-SR04 .....	15
Gambar 2. 6 Ilustrasi Cara Kerja Sensor Ultrasonik .....	16
Gambar 2. 7 Light Emitting Diode.....	19
Gambar 2. 8 NodeMCU ESP8266 .....	21
Gambar 3. 1 Prosedur Pengembangan ADDIE.....	28
Gambar 3. 2 Desain Rancangan Sistem Parkir .....	32
Gambar 3. 3 Flowchart Sistem Parkir.....	34
Gambar 3. 4 Ilustrasi Sistematis Rangkaian Hardware .....	36
Gambar 3. 6 Gambar Program Arduino UNO .....	38
Gambar 3. 7 Gambar Konfigurasi Databae Firebase.....	39
Gambar 3. 8 Ilustrasi Tampilan Antarmuka Web.....	40
Gambar 3. 9 Gambar Miniatur Tempat Parkir .....	42
Gambar 3. 10 Gambar Serial Monitor .....	44
Gambar 3. 11 Pengujian Komunikasi Sensor dengan LED.....	45
Gambar 3. 12 Gambar Pemantuan di Serial Monitor .....	45
Gambar 3. 13 Gambar Pengujian dengan Beberapa Objek....	46
Gambar 3. 14 Gambar serial komunikasi antara Arduino dengan NodeMCU.....	47

Gambar 3. 15 Gambar sinkronisasi antara Firebase dengan NodeMCU.....	47
Gambar 3. 16 Gambar pengujian web .....	48
Gambar 3. 17 Gambar Pengujian Secara Realtime .....	49
Gambar 4. 1 Denah Desain Miniatur .....	53
Gambar 4. 2 Rangkaian Hardware Sederhana .....	54
Gambar 4. 3 Gambar Tampilan Web pada Browser .....	56
Gambar 4. 4 Gambar Miniatur yang Telah Dibuat.....	56
Gambar 4. 5 Sistem Hardware Satu Objek Kondisi 1 .....	62
Gambar 4. 6 Sistem Software Satu Objek Kondisi 1.....	62
Gambar 4. 7 Sistem Hardware Satu Objek Kondisi 2 .....	63
Gambar 4. 8 Sistem Software Satu Objek Kondisi 2.....	63
Gambar 4. 9 Sistem Hardware Satu Objek Kondisi 3 .....	64
Gambar 4. 10 Sistem Software Satu Objek Kondisi 3 .....	65
Gambar 4. 11 Sistem Hardware Satu Objek Kondisi 4.....	66
Gambar 4. 12 Sistem Software Satu Objek Kondisi 4.....	66
Gambar 4. 13 Sistem Hardware Dua Objek Kondisi 1 .....	67
Gambar 4. 14 Sistem Software Dua Objek Kondisi 1.....	67
Gambar 4. 15 Sistem Hardware Dua Objek Kondisi 2 .....	68
Gambar 4. 16 Sistem Software Dua Objek Kondisi 2.....	69
Gambar 4. 17 Sistem Hardware Dua Objek Kondisi 3 .....	70
Gambar 4. 18 Sistem Software Dua Objek Kondisi 3.....	70
Gambar 4. 19 Sistem Hardware Tiga Objek Kondisi 1.....	71
Gambar 4. 20 Sistem Software Tiga Objek Kondisi 1 .....	72
Gambar 4. 21 Sistem Hardware Tiga Objek Kondisi 2.....	73

Gambar 4. 22 Sistem Software Tiga Objek Kondisi 2 .....	73
Gambar 4. 23 Sistem Hardware Empat Objek.....	74
Gambar 4. 24 Sistem Software Empat Objek.....	75





## DAFTAR LAMPIRAN

Lampiran 1 Source Code .....	89
------------------------------	----



# BAB I

## PENDAHULUAN

### A. Latar Belakang

Parkir dapat didefinisikan sebagai tindakan menghentikan sementara suatu kendaraan di lokasi yang ditentukan, baik dengan rambu lalu lintas resmi maupun tanpa rambu tersebut. Aktivitas ini bukan hanya untuk menaikturunkan penumpang atau barang, tetapi juga untuk tujuan lain (Nawawi, 2015 dalam Ridwan, 2017). Peningkatan jumlah toko dan pusat perbelanjaan yang terus berkembang di kota-kota besar umumnya terfokus pada satu lokasi tertentu. Dampaknya adalah peningkatan lalu lintas kendaraan yang tinggi, sehingga masyarakat semakin membutuhkan tempat parkir (Fauzan, 2019).

Pada penelitian yang dilakukan oleh Ilham et al. (2022), terdapat peningkatan indeks parkir sebesar 21,33% dengan rincian indeks parkir saat *weekday* 23,17% dan *weekend* 44,5%. Peningkatan indeks parkir dapat menyebabkan berbagai masalah, seperti kesulitan dalam mencari tempat parkir yang mengakibatkan kemacetan dan gangguan pada aktivitas. Area parkir di pusat perbelanjaan, hotel, dan gedung perkantoran, termasuk lahan parkir di *basement*, kesulitan dalam mengatasi sistem parkir dengan mengandalkan petugas parkir secara

konvensional. Memantau dan mengatur setiap lantai parkir secara tradisional membutuhkan jumlah petugas parkir yang banyak agar pemantauan parkir dapat dilakukan dengan efektif. Dalam situasi di mana jumlah petugas parkir terbatas, masalah timbul ketika pengguna parkir baru masuk ke area parkir yang luas, karena sulit bagi mereka untuk menemukan slot parkir yang kosong di setiap lantai. Akibatnya, pengguna parkir harus berputar mengelilingi area parkir untuk mencari slot parkir yang tersedia (Fauzan, 2019).

Untuk mengatasi masalah parkir tersebut, peneliti sedang mencari solusi dengan memanfaatkan mikrokontroler. Arduino, sebagai salah satu jenis mikrokontroler *open source* dengan tipe AVR yang diproduksi oleh perusahaan Atmel, adalah papan elektronik yang berisi mikrokontroler. Arduino dapat digunakan untuk membuat rangkaian elektronik sederhana maupun kompleks (Kadir, 2012).

Dalam penggunaan mikrokontroler, diperlukan perangkat keras yang berfungsi sebagai alat pemantauan ketersediaan slot parkir. Sensor ultrasonik HC-SR04 adalah jenis sensor yang menggunakan gelombang ultrasonik untuk mengukur jarak. Prinsip kerja sensor ini mirip dengan radar ultrasonik, di mana gelombang ultrasonik dipancarkan dan kemudian diterima kembali

oleh *receiver* ultrasonik. Perbedaan waktu antara saat pancaran dan saat penerimaan gelombang ultrasonik mewakili jarak objek. Sensor ini cocok digunakan dalam aplikasi elektronik yang membutuhkan deteksi jarak, termasuk penggunaan sebagai sensor pada robot (Kianpisheh, 2012).

Berangkat dari masalah yang telah disebutkan, penulis berupaya mengembangkan sebuah prototipe sistem pemantauan parkir otomatis yang bertujuan untuk memberikan kemudahan kepada pengguna kendaraan empat roda. Dalam hal ini, teknologi otomatisasi digunakan dengan memanfaatkan mikrokontroler berbasis *open source*, yaitu Arduino Uno, dan sensor ultrasonik HC-SR04. Dengan menggunakan teknologi ini, keadaan tempat parkir dapat dengan mudah dipantau, baik dalam hal jumlah tempat parkir yang tersedia maupun posisi tempat parkir yang masih kosong.

## **B. Rumusan Masalah**

Rumusan masalah yang diangkat dari penelitian ini yaitu :

1. Bagaimana merancang dan membangun sistem monitoring ketersediaan tempat parkir menggunakan Arduino uno?

2. Bagaimana performa sistem monitoring ketersediaan tempat parkir?

### **C. Tujuan Penelitian**

Tujuan dari penelitian ini sebagai berikut :

1. Merancang dan membangun sistem monitoring ketersediaan tempat parkir.
2. Mengetahui performa kinerja sistem monitoring ketersediaan tempat parkir.

### **D. Batasan Masalah**

Dalam rangka memfokuskan pembahasan pada topik yang telah diuraikan dan menghindari pembahasan yang melebar, batasan penelitian ini adalah :

1. Objek penelitian berupa sebuah prototipe miniatur tempat parkir.
2. Hanya admin yang dapat melakukan *create, read, update, dan delete*.
3. Jumlah slot parkir yang dibuat dalam prototipe hanya empat slot.
4. Pengujian menggunakan *black box testing*.

### **E. Manfaat Penelitian**

Manfaat teoritis dari penelitian ini sebagai berikut :

1. Memberikan informasi ilmiah terhadap mahasiswa dan masyarakat pada umumnya tentang sistem

monitoring ketersediaan tempat parkir menggunakan sensor ultrasonik HC-SR04 dan Arduino.

2. Sebagai bahan penelitian lanjutan bagi civitas akademika.

Manfaat praktis dari penelitian ini sebagai berikut :

1. Memberikan informasi kepada masyarakat dan *stakeholder* mengenai cara yang tepat dalam penanganan masalah sulitnya pencarian ketersediaan slot parkir.
2. Meringankan petugas parkir dalam hal dari yang sebelumnya manual menjadi otomatis dengan pemanfaatan sistem monitoring ketersediaan tempat parkir.





## **BAB II**

### **LANDASAN PUSTAKA**

#### **A. Kajian Pustaka**

##### **1. Parkir**

Pengertian mengenai parkir dapat dibagi menjadi dua, yaitu sebagai tempat untuk menghentikan sementara kendaraan dan juga sebagai tempat untuk menghentikan kendaraan dalam jangka waktu yang panjang atau singkat sesuai dengan kebutuhan (Setijowarno, 2001).

Menurut Direktorat Jenderal Perhubungan Darat (1998), parkir adalah saat kendaraan berhenti untuk sementara tanpa adanya pergerakan, sedangkan berhenti mengacu pada saat kendaraan tidak bergerak untuk sementara waktu dengan pengemudi masih berada di dalam kendaraan. Parkir merupakan kebutuhan yang diinginkan oleh pemilik kendaraan yang ingin memarkirkan kendaraannya di tempat yang mudah dijangkau. Salah satu kemudahan tersebut adalah parkir di badan jalan.

##### **2. Jenis Parkir**

- a. Parkir di Badan Jalan (On street Parking)

Menurut Ditjen Perhubungan Darat (1998), pengertian fasilitas parkir pada badan jalan memiliki kesamaan dengan pengertian kawasan parkir. Fasilitas parkir pada badan jalan merujuk pada fasilitas parkir yang menggunakan sisi/tepi badan jalan. Fasilitas parkir pada badan jalan merupakan area parkir yang memanfaatkan badan jalan sebagai tempat parkir, namun hanya pada kawasan parkir tersebut terdapat pengendalian parkir melalui pintu masuk.

Meskipun hanya sedikit kendaraan yang diparkir di badan jalan, namun secara efektif kendaraan tersebut telah menyebabkan penyempitan badan jalan. Kendaraan yang diparkir di sisi jalan menjadi faktor utama dari 50% kecelakaan yang terjadi di tengah ruas jalan di daerah pertokoan. Hal ini terutama disebabkan oleh gangguan pandangan yang berkurang, ketika kendaraan berhenti atau keluar dari tempat parkir secara tiba-tiba di depan kendaraan yang sedang melintas. (Ditjen Perhubungan Darat, 1998).

b. Parkir di luar Badan Jalan (Off Street Parking)

Menurut Ditjen Perhubungan Darat (1998), fasilitas parkir di luar badan jalan merujuk pada

fasilitas parkir kendaraan yang tidak berada di dalam badan jalan atau tidak langsung berada di badan jalan, melainkan berada di area di luar badan jalan yang telah disediakan secara khusus. Jenis parkir ini mencakup area parkir umum, tempat parkir khusus yang juga terbuka untuk umum, serta tempat parkir khusus yang dibatasi untuk keperluan pribadi seperti kantor, pusat perbelanjaan, dan sebagainya.

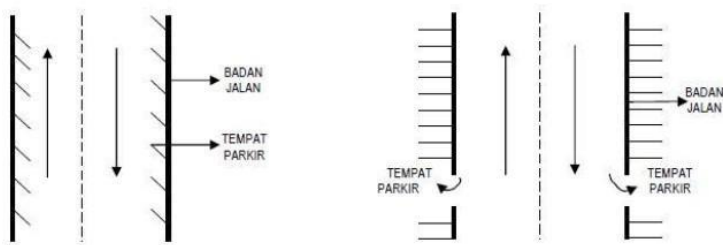
Sistem parkir di luar badan jalan dapat berbentuk area parkir terbuka atau bangunan parkir bertingkat khusus. Idealnya, lokasi parkir di luar badan jalan (off street parking) harus dibangun dengan jarak yang tidak terlalu jauh dari tujuan pemarkir. Jarak terjauh antara tempat parkir dan tujuan tidak boleh melebihi 300-400 meter. Jika jaraknya lebih dari itu, pemarkir akan mencari tempat parkir lain karena enggan untuk berjalan jauh (Warpani,1990). Dalam penempatan fasilitas parkir di luar badan jalan dapat dikelompokkan atas dua bagian, yakni:

1. Fasilitas untuk umum yaitu tempat parkir berupa gedung parkir atau taman parkir

untuk umum yang diusahakan sebagai kegiatan sendiri.

2. Fasilitas parkir penunjang yaitu berupa gedung parkir atau taman parkir yang disediakan untuk menunjang kegiatan pada bangunan utama.

Pada Gambar 2.1 menunjukkan model pola parkir di tepi jalan dan di luar jalan.



Gambar 2. 1 Model-model pola parkir

- a. Parkir di tepi jalan  
(on street parking)
- b. Parkir di luar jalan  
(off street parking)

Sumber: Miro (1997)

### 3. Arduino Uno

Arduino adalah sebuah papan mikrokontroler serba guna yang dapat diprogram dan bersifat *open source*. Platform Arduino telah menjadi sangat populer dan terus menarik jumlah pengguna baru yang terus meningkat. Popularitasnya disebabkan oleh

kemudahan penggunaan dan penulisan kode program. Berbeda dengan kebanyakan papan sirkuit pemrograman sebelumnya, Arduino tidak memerlukan perangkat keras terpisah (seperti *programmer* atau *downloader*) untuk memuat atau meng-upload kode baru ke mikrokontroler. Cukup menggunakan kabel USB, pengguna dapat dengan mudah memulai menggunakan Arduino (Yusro, 2016). Pada Gambar 2.2 menunjukkan bentuk dari papan mikrokontroler Arduino Uno.



Gambar 2. 2 Arduino UNO

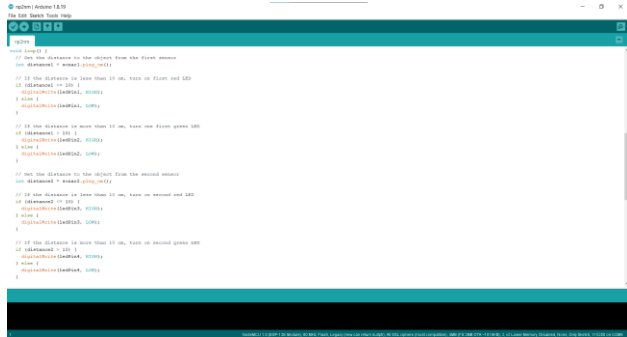
Sumber: P. M. Astra (2020)

Arduino UNO adalah sebuah papan mikrokontroler yang berbasis pada ATmega328. Arduino UNO memiliki 14 pin *input/output* digital (6 di antaranya dapat digunakan sebagai *output* PWM), 6 *input* analog, sebuah osilator Kristal 16 MHz, koneksi USB, jack daya,

header ISCP, dan tombol reset. Arduino UNO menyediakan semua yang diperlukan untuk mendukung mikrokontroler, dengan mudah dapat terhubung ke komputer melalui kabel USB atau mendapatkan pasokan daya melalui adaptor AC ke DC atau menggunakan baterai sebagai sumber daya (Sembiring, 2020).

#### **4. Arduino IDE**

Arduino memiliki bahasa pemrograman khusus yang disebut bahasa pemrograman Arduino. Bahasa pemrograman Arduino merupakan implementasi yang disederhanakan dari bahasa C dan C++. Hal ini dilakukan agar pengguna dapat berinteraksi dengan Arduino lebih mudah. Arduino juga dilengkapi dengan perangkat lunak sendiri yang disebut Arduino IDE. Arduino IDE adalah alat yang berguna untuk menulis program (yang dalam konteks Arduino disebut sebagai "sketsa"), mengkompilasi program tersebut, dan mengunggahnya ke papan Arduino (Kadir, 2016). Tampilan antarmuka Arduino IDE ditunjukkan dalam Gambar 2.3.



Gambar 2. 3 Arduino IDE

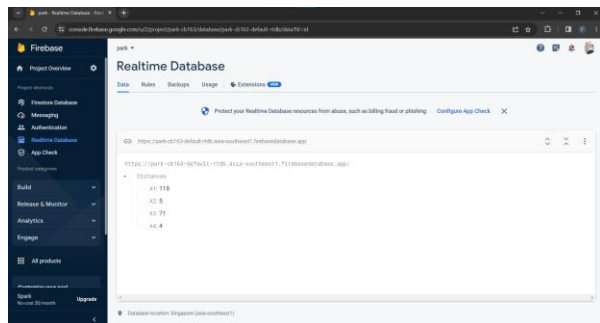
## 5. Firebase

Firebase adalah layanan dari Google yang berguna bagi pengembang aplikasi dalam menyederhanakan proses pengembangan aplikasi. Dengan Firebase, para pengembang dapat lebih fokus pada pengembangan aplikasi tanpa harus mengeluarkan usaha yang besar. Terdapat dua fitur menarik yang ditawarkan oleh Firebase, yaitu Firebase Remote Config dan Firebase Realtime Database. Selain itu, Firebase juga menyediakan fitur tambahan untuk aplikasi yang membutuhkan pemberitahuan, seperti Firebase Notification (Firebase Realtime Database, 2023).

Firebase menyediakan layanan Realtime Database dan *backend* sebagai layanan. Layanan ini menawarkan antarmuka pemrograman aplikasi yang memungkinkan data aplikasi untuk disinkronkan di



berbagai klien dan disimpan di Firebase cloud. Firebase juga menyediakan pustaka klien yang memungkinkan integrasi dengan aplikasi Android, iOS, JavaScript, Java, Swift, dan Node.js. Selain itu, *database* dapat diakses melalui REST API dan dapat diikat ke beberapa kerangka kerja JavaScript seperti AngularJS, React, Ember.js, dan Backbone.js. Pengembang yang menggunakan Realtime Database Firebase dapat mengamankan data dengan menerapkan aturan keamanan yang dikelola oleh server perusahaan. Firebase Realtime Database adalah *database* yang di-*host* di *cloud*, di mana data disimpan dalam format JSON dan disinkronkan secara realtime ke setiap klien yang terhubung (Firebase Realtime Database, 2023). Tampilan antarmuka konsol pada Firebase ditunjukkan pada Gambar 2.4.



Gambar 2. 4 Firebase Database

## 6. Sensor Ultrasonik HC-SR04

Sensor ultrasonik adalah jenis sensor yang bekerja dengan memanfaatkan prinsip pantulan gelombang suara. Sensor ini digunakan untuk mendeteksi keberadaan objek atau benda tertentu di depannya dalam rentang frekuensi di atas gelombang suara, mulai dari 20kHz hingga 2MHz. Sensor ultrasonik terdiri dari dua unit, yaitu unit pemancar dan penerima (Bakhtiyar, 2017). Gambar 2.5 menunjukkan bentuk dari sensor ultrasonic HC-SR04.



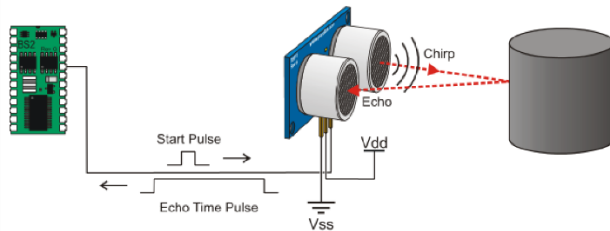
*Gambar 2. 5 Sensor Ultrasonik HC-SR04*

Sumber: Muhammad (2020)

Kristal piezoelektrik terhubung dengan mekanisme jangkar dan terhubung dengan diafragma yang bergetar tegangan bolak-balik dengan frekuensi kerja antara 20kHz hingga 2MHz. Struktur atom dalam kristal piezoelektrik menyebabkan kontraksi atau

ekspansi, serta polaritas yang menghasilkan efek piezoelektrik pada sensor ultrasonik (Rissa, 2019).

Pantulan ultrasonik terjadi ketika ada objek tertentu dan gelombang ultrasonik yang dipancarkan akan diterima kembali oleh unit sensor penerima. Setelah itu, unit sensor penerima akan menyebabkan diafragma bergetar, dan efek piezoelektrik akan menghasilkan tegangan bolak-balik dengan frekuensi yang sama. Prinsip cara kerja ultrasonik dapat dilihat pada Gambar 2.6.



*Gambar 2. 6 Ilustrasi Cara Kerja Sensor Ultrasonik*

Sumber: Heru (2019)

Besar amplitudo sebuah sinyal listrik yang dihasilkan oleh sensor penerima bergantung pada jarak dekat atau jauhnya objek yang akan dideteksi, serta kualitas sensor pemancar dan penerima. Proses penginderaan yang dilakukan oleh sensor menggunakan metode pantulan untuk mengukur jarak

antara sensor dan objek yang menjadi target (Heru, 2019).

HC-SR04 adalah jenis sensor jarak yang menggunakan gelombang ultrasonik. Sensor ultrasonik ini terdiri dari dua transduser, yaitu pemancar (*transmitter*) yang menghasilkan gelombang ultrasonik dan penerima (*receiver*) yang menerima pantulan gelombang tersebut. Prinsip kerja sensor ultrasonik ini adalah *transmitter* mengirimkan pulsa gelombang ultrasonik dengan frekuensi 40 kHz, kemudian waktu yang dibutuhkan untuk pantulan dari objek diukur (Heru, 2019).

Menurut Bakhtiyar & Bambang (2017) Prinsip kerja HC-SR04 melibatkan pemancaran sinyal ultrasonik (20 KHz) dalam bentuk pulsa oleh *transmitter*. Jika ada objek padat di depan HC-SR04, *receiver* akan menerima pantulan sinyal ultrasonik tersebut. *Receiver* membaca lebar pita gelombang yang dipantulkan oleh objek dalam bentuk PWM dan mengukur selisih waktu pemancaran. Dengan pengukuran ini, jarak objek di depan sensor dapat diketahui. Pin *trigger* dan *echo* dihubungkan ke mikrokontroler. Untuk memulai pengukuran jarak, mikrokontroler mengeluarkan sinyal *high* pada pin *trigger* selama minimal 10  $\mu$ S, yang menyebabkan HC-

SR04 memancarkan gelombang suara ultrasonik. Ketika pantulan bunyi kembali ke HC-SR04, sinyal *high* akan diterima oleh pin *echo* dan menjadi input pada mikrokontroler. HC-SR04 akan menghasilkan pulsa gelombang dengan durasi 100  $\mu$ s - 18 ms pada *output*-nya, tergantung pada informasi jarak pantulan objek yang diterima. Durasi sinyal *high* dari pin *echo* digunakan untuk menghitung jarak antara HC-SR04 dan objek yang memantulkan suara di depan sensor.

## **7. Light Emitting Diode (LED)**

LED (Light Emitting Diode) adalah sebuah lampu indikator yang digunakan dalam perangkat elektronik untuk menampilkan status perangkat tersebut (Nayomi, 2013). Lampu LED dibuat dari bahan semikonduktor yang akan menghasilkan cahaya dengan satu warna (monokromatik) berdasarkan prinsip pemberian tegangan maju (*forward bias*). Ketika diberikan tegangan, aliran elektron terjadi dari lapisan tipe p ke lapisan tipe n melalui lapisan *depletion* yang tipis dalam kondisi *loop* tertutup. (Saputro, 2013). Gambar 2.7 menunjukkan bentuk dari perangkat elektronik LED.



*Gambar 2. 7 Light Emitting Diode*

LED memiliki umur pemakaian yang lebih lama dibandingkan dengan sumber penerangan lainnya. Namun, untuk memastikan cahaya yang dihasilkan tetap optimal, diperlukan manajemen panas yang baik dalam proses pengemasan. Oleh karena itu, beberapa industri telah mengembangkan LED yang tahan lama dan dapat diandalkan dengan memperhatikan faktor ini (Narendran, 2005).

LED memiliki karakteristik berbeda-beda menurut warna yang dihasilkan. Semakin tinggi arus yang mengalir pada karakteristik listrik dan optik pada LED dan laser 204 LED maka semakin terang pula cahaya yang dihasilkan, namun perlu diperhatikan bahwa LED memiliki batasan arus yang diperbolehkan, yaitu sekitar 10mA hingga 20mA, dan tegangan yang berbeda-beda tergantung pada warna yang dihasilkan, berkisar antara 1,6V hingga 3,5V. Jika arus yang

melewati LED melebihi 20mA, LED dapat terbakar. Oleh karena itu, untuk mencegah terjadinya kerusakan, resistor digunakan sebagai penghambat arus pada rangkaian LED (Hsieh and Wang, 2017).

## **8. Kabel Jumper**

Kabel jumper merupakan kabel elektrik yang digunakan untuk menghubungkan komponen-komponen di *breadboard* tanpa perlu menggunakan solder. Kabel jumper biasanya dilengkapi dengan konektor atau pin di kedua ujungnya. Konektor yang digunakan untuk menusuk disebut *male connector*, sementara konektor yang digunakan untuk ditusuk disebut *female connector*. Kabel jumper terbagi menjadi tiga jenis, yaitu: *Male to Male*, *Male to Female*, dan *Female to Female* (Wibowo, 2020).

Kabel jumper digunakan sebagai penghubung antara komponen-komponen dalam pembuatan perangkat prototipe. Kabel jumper dapat terhubung dengan kontroler seperti Raspberry Pi dan Arduino melalui *breadboard*. Karakteristik kabel jumper ini termasuk memiliki panjang antara 10 hingga 20 cm. Jenis kabel jumper ini menggunakan kabel serat tunggal dan memiliki *housing* yang berbentuk bulat. (Wibowo, 2020).

## 9. NodeMCU Esp8266

NodeMCU adalah sebuah *platform IoT* dan kit pengembangan *open source* yang menggunakan bahasa pemrograman Lua. Media ini digunakan untuk membantu dalam pembuatan prototipe produk IoT dan juga dapat digunakan dengan menggunakan *sketch* melalui Arduino IDE. Kit pengembangan ini didasarkan pada modul ESP8266, yang mengintegrasikan fitur GPIO, PWM (*Pulse Width Modulation*), IIC, 1-Wire, dan ADC (*Analog to Digital Converter*) dalam satu papan. NodeMCU memiliki ukuran panjang 4.83cm, lebar 2.54cm, dan berat 7 gram. Papan ini sudah dilengkapi dengan fitur WiFi dan *firmware* yang bersifat *open source*(Purwanto, 2019). Gambar 2.8 menunjukkan bentuk dari modul wifi NodeMCU ESP 8266.



Gambar 2. 8 NodeMCU ESP8266

Sumber: Derwin (2020)

NodeMCU dapat dianggap sebagai versi board Arduino dari ESP8266. Memprogram ESP8266 sedikit rumit karena membutuhkan beberapa teknik



kabelisasi dan modul tambahan untuk mengunggah program. Namun, NodeMCU telah mengemas ESP8266 ke dalam sebuah board yang kompak dengan berbagai fitur seperti yang dimiliki oleh mikrokontroler, termasuk kemampuan akses WiFi dan *chip* komunikasi *USB to serial*. Dengan demikian, untuk memprogramnya, hanya diperlukan kabel data USB yang sama seperti yang digunakan untuk mengisi daya smartphone. NodeMCU ESP8266 adalah sebuah *board* interaktif yang menggunakan bahasa pemrograman LUA. NodeMCU ESP8266 memiliki kapasitas memori 4Mb dan kecepatan clock 80-160 Mhz (Hidayat, 2017).

## B. Kajian Penelitian yang Relevan

Ada beberapa hasil penelitian terdahulu yang relevan atau berhubungan dengan penelitian yang dilakukan oleh peneliti, yaitu sebagai berikut :

Tabel 2. 1 Penelitian yang relevan

No	Judul & Penulis	Hasil Penelitian	Perbedaan dengan penelitian ini
1.	Sistem Monitoring Tempat Parkir Dengan Sensor	Hasil penelitian ini adalah aplikasi yang akan <i>update</i> melalui twitter	Penelitian di samping membutuhkan koneksi internet

	<p>Ultrasonik Berbasis Arduino Uno Pada Cibinong City Mall (Muhajir, 2016)</p>	<p>secara otomatis setiap jangka waktu yang ditentukan.</p>	<p>yang stabil, tidak update secara <i>realtime</i> dan tidak menunjukkan slot parkir yang kosong.</p>
2.	<p>PROTOTYPE SISTEM MONITORING PARKIR MOBIL BERBASIS MIKROKONTROLER (Ramadhan, 2019)</p>	<p>Hasil penelitian ini adalah aplikasi monitoring menggunakan mikrokontroler yang akan menutup palang di depan area parkir jika slot parkir sudah penuh.</p>	<p>Penelitian di samping tidak ada layar monitor sebagai penunjuk informasi penunjuk lokasi slot yang kosong.</p>
3.	<p>RANCANG BANGUN SISTEM MONITORING TEMPAT PARKIR BERBASIS ARDUINO (Ridwan, 2017)</p>	<p>Hasil penelitian bahwa aplikasi sistem monitoring tempat parkir dapat berjalan pada sistem operasi windows 7, 8, dan 10. Dan</p>	<p>Penelitian di samping masih menggunakan kabel. Sedangkan penelitian ini menggunakan wireless.</p>

		mendeteksi mobil pada tempat parkir pada jarak maksimal 7cm.	
4.	Sistem Monitoring Tempat parkir Berbasis Arduino Mega Dengan modul Komunikasi XBee Pro S2c (Pardede, 2019)	Hasil penelitian ini menggunakan modul komunikasi nirkabel XBeePro S2C maka lokasi parkir yang dipantau dapat lebih luas dimana jarak pancar komunikasi mencapai 95m.	Penelitian di samping terbatas jarak dalam melihat hasil pemantauan pada lokasi parkir.
5.	Sistem Monitoring Parkir Mobil Berbasis Mikrokontroler Arduino Uno (Mustaziri, 2020)	Hasil penelitian ini akan bekerja saat kartu RFID yang terdaftar melakukan scanner akan menghasilkan tampilan pada LCD	Penelitian di samping tidak menggunakan tampilan web sebagai penunjuk informasi penunjuk lokasi slot yang kosong.

		dan palang parkir akan terbuka tetapi jika area parkir sudah penuh maka LCD akan menampilkan "Parkir FULL".	
--	--	---	--

Dari tabel di atas dapat disimpulkan bahwa penelitian ini adalah membuat sistem monitoring tempat parkir yang telah dilakukan oleh Muhajir, Ramadhan, dan Ridwan. Tetapi perbedaan mendasar pada penelitian ini adalah digungkannya layar monitor sebagai sumber informasi utama letak slot parkir yang kosong dan menggunakan sistem *wireless*.



## **BAB III**

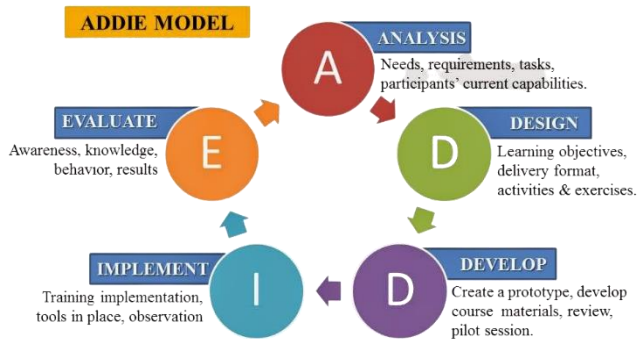
### **METODE PENELITIAN**

#### **A. Model Pengembangan**

Penelitian pengembangan merupakan sebuah metode penelitian yang digunakan untuk menciptakan dan menguji efektivitas produk tertentu. Salah satu bentuk penelitian pengembangan yang umum adalah yang melibatkan tahap pengembangan produk, dilanjutkan dengan analisis, dan terakhir evaluasi terhadap produk akhir yang dihasilkan (Sugiyono 2011).

Model pengembangan yang digunakan dalam penelitian pengembangan ini adalah model ADDIE (*Analysis, Design, Development, Implementation, and Evaluation*). Model ADDIE dipilih karena memiliki tahapan yang sistematis dan sederhana. Model ini dapat beradaptasi dengan baik dalam berbagai situasi, sehingga masih relevan digunakan hingga saat ini. Model ADDIE memiliki tingkat fleksibilitas yang tinggi dalam menangani berbagai permasalahan. Meskipun fleksibel, model ADDIE tetap efektif dan banyak yang akrab dengan singkatannya. Selain itu, model ADDIE menyediakan kerangka kerja terstruktur untuk pengembangan intervensi instruksional, serta melibatkan evaluasi dan revisi dalam setiap tahapnya

(Angko 2013). Gambar 3.1 menunjukkan prosedur model pengembangan ADDIE.



*Gambar 3. 1* Prosedur Pengembangan ADDIE

Sumber: [grafispaten.wordpress.com](http://grafispaten.wordpress.com)

Desain model ADDIE dikembangkan oleh Dick & Carry pada tahun 1996 untuk mengembangkan produk pembelajaran (Cahyadi, 2019). Berikut langkah atau tahapan dari desain model ADDIE :

1. Analisis merupakan tahapan dalam model ADDIE yang melibatkan pengidentifikasian kebutuhan, masalah yang ada, serta pemikiran tentang produk yang akan dikembangkan sesuai dengan sasaran yang ditetapkan.
2. Tahap desain, dalam konteks model ADDIE, merupakan tahap di mana dilakukan perancangan konsep produk yang akan dikembangkan.

3. Tahap pengembangan, dalam model ADDIE, adalah proses mengubah desain awal menjadi produk yang telah selesai.
4. Implementasi, dalam konteks model ADDIE, adalah tahap di mana produk yang sedang dikembangkan diuji coba sebagai langkah konkret untuk menerapkan produk tersebut.
5. Evaluasi merupakan proses untuk mengevaluasi keberhasilan produk yang telah dibuat dan menilai sejauh mana produk tersebut mencapai harapan awal yang ditetapkan.

## **B. Prosedur Pengembangan**

Prosedur pengembangan ADDIE terdiri dari 5 tahap yang meliputi Analisis, Desain, Pengembangan, Implementasi, dan Evaluasi. Model ADDIE sering digunakan dalam penelitian pengembangan karena memiliki tahapan yang sistematis. Namun, dalam penelitian ini, model pengembangan tersebut dimodifikasi sehingga hanya mencakup tahap implementasi saja, sesuai dengan kebutuhan penelitian.

### **1. Tahap Analisis (*Analyze*)**

Tahap analisis dalam pengembangan sistem pembantu monitoring ketersediaan tempat parkir



merupakan langkah kunci dalam merancang solusi yang efektif. Tujuan dari tahap ini adalah untuk memahami masalah di lapangan dan mengidentifikasi kebutuhan yang harus dipenuhi oleh produk yang akan dikembangkan.

Berdasarkan hasil penelitian yang dilakukan oleh Ilham et al. (2022), terlihat bahwa terjadi peningkatan indeks parkir yang signifikan, terutama selama akhir pekan. Peningkatan ini telah mengakibatkan masalah seperti kesulitan mencari tempat parkir, kemacetan, dan gangguan dalam aktivitas sehari-hari. Oleh karena itu, diperlukan penanganan yang efektif untuk mengatasi masalah ini.

Solusi yang akan diimplementasikan adalah sistem monitoring ketersediaan tempat parkir menggunakan Arduino dan sensor ultrasonik. Dalam proses analisis kebutuhan, telah diidentifikasi bahwa komponen *hardware* yang diperlukan meliputi Sensor Ultrasonik HC-SR04 untuk mendeteksi keberadaan kendaraan, modul wifi NodeMCU Esp8266 untuk mengirim data ke *database*, dan Arduino Uno untuk mengendalikan perangkat keras.

Di sisi perangkat lunak, kebutuhan *software* mencakup Arduino IDE yang digunakan untuk pemrograman perangkat keras, Visual Studio Code

untuk mengembangkan tampilan *web*, dan Firebase sebagai platform penyimpanan data *online*. Semua komponen ini akan bekerja bersama-sama untuk menciptakan sistem yang dapat memberikan informasi *realtime* tentang ketersediaan tempat parkir kepada pengguna.

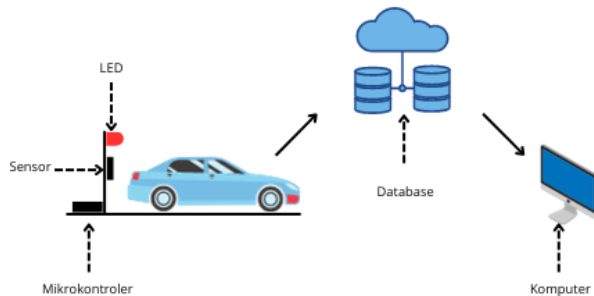
Pada analisis kebutuhan untuk pembuatan prototipe miniatur lahan parkir melibatkan perlengkapan alat tulis kantor dalam menciptakan miniatur yang representatif. Styrofoam digunakan sebagai bahan dasar untuk membentuk struktur fisik miniatur lahan parkir. Untuk mengolah styrofoam dengan presisi, alat pemotong, seperti cutter, akan digunakan. Pengukuran yang akurat dilakukan dengan penggunaan pengaris sebagai alat pengukur dan pemandu. Pulpen dan spidol berfungsi untuk memberikan tanda atau garis pada styrofoam, memastikan pemotongan dan penempatan komponen berjalan sesuai rencana. Terakhir, soldier digunakan untuk melubangi styrofoam demi menempatkan sensor ultrasonik dengan presisi.

## 2. Tahap Desain (*Design*)

Tahap desain ini berfungsi sebagai panduan dalam merancang aplikasi setelah mendapatkan informasi

dari tahap analisis. Fokus desain ini adalah pada pengembangan sistem aplikasi yang dapat memonitor ketersediaan tempat parkir dengan efektif.

Dalam desain ini, sistem akan bekerja sebagai berikut: sensor yang dipasang di setiap slot parkir akan berfungsi untuk mendeteksi apakah slot tersebut terisi oleh mobil atau tidak. Informasi yang diperoleh oleh sensor akan diteruskan ke Arduino, yang akan mengendalikan komunikasi dengan modul wifi. Modul wifi digunakan untuk mengirimkan data ke aplikasi melalui jaringan wifi. Gambar 3.2 menunjukkan desain dari rancangan sistem monitoring parkir.

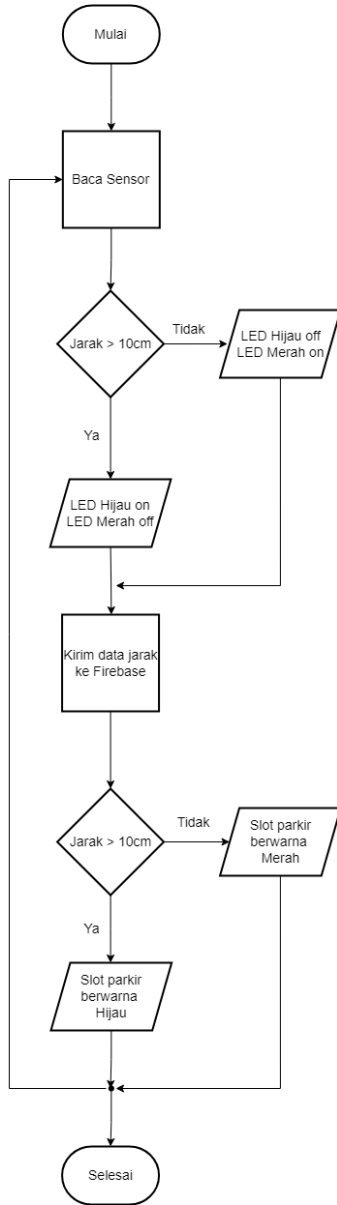


Gambar 3. 2 Desain Rancangan Sistem Parkir

Aplikasi ini akan memiliki antarmuka pengguna yang dapat diakses melalui perangkat berbasis *web*. Antarmuka ini akan menampilkan status ketersediaan tempat parkir secara *realtime* kepada pengguna. Setiap slot parkir akan ditampilkan dengan lampu LED yang

akan berubah warna berdasarkan informasi dari sensor. Pengguna dapat melihat status ini melalui tampilan *web* dengan mudah.

Data ketersediaan tempat parkir yang dikirim oleh modul wifi akan disimpan dalam *database* yang di-*hosting* di Firebase. Aplikasi akan mengakses data ini dari Firebase dan menampilkan informasi yang relevan kepada pengguna. Selama tahap desain ini, program akan dirancang untuk mikrokontroler yang akan mengendalikan sensor dan modul wifi. Penggunaan Arduino IDE dan Visual Studio Code akan mendukung pengembangan perangkat keras dan perangkat lunak. Pada Gambar 3.3 menunjukkan flowchart dari sistem monitoring parkir, dimana sensor membaca data jarak yang kemudian warna led akan menunjukkan status slot parkir, setelah itu data jarak dikirim ke Firebase dan status slot parkir akan ditampilkan pada tampilan web (Yuniarti, W.D., 2019).



Gambar 3. 3 Flowchart Sistem Parkir

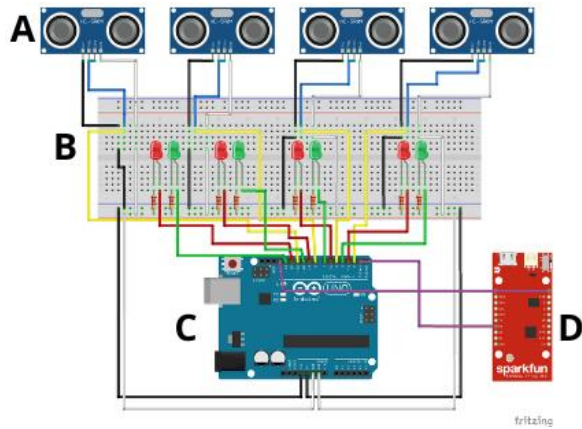
### 3. Tahap Pengembangan (*Development*)

Setelah menyelesaikan tahap analisis dan tahap desain, langkah selanjutnya adalah tahap pengembangan. Tahap ini merupakan fase di mana konsep yang telah dirancang dalam tahap desain akan diwujudkan menjadi produk konkret. Pada tahap ini, produk akan dikembangkan dengan implementasi komponen *hardware*, *software* dan sistem antarmuka yang telah direncanakan.

#### a) Pengembangan *Hardware*

Tahap pengembangan perangkat keras adalah langkah penting dalam pengembangan sistem monitoring ketersediaan tempat parkir. Proses ini melibatkan rangkaian komponen utama seperti Sensor Ultrasonik HC-SR04, modul WiFi NodeMCU Esp8266, dan LED. Rangkaian ini akan dirancang dan dirakit dengan teliti sesuai dengan desain yang telah direncanakan. Gambar 3.4 menunjukkan ilustrasi sistematis dari rangkaian yang akan dirancang, dimana objek yang ditunjukkan dengan huruf A adalah sensor ultrasonik yang akan kemudian dihubungkan ke breadboard yang ditunjukkan pada huruf B, breadboard di sini

berfungsi sebagai tempat komponen hardware bertemu. Kemudian pada huruf C ditunjukkan Arduino sebagai mikrokontroler yang mengatur semua rangkaian *hardware*, dan kemudian data yang diterima oleh Arduino dikirim ke NodeMCU yang ditunjukkan pada huruf C.



*Gambar 3. 4 Ilustrasi Sistematis Rangkaian Hardware*

Selanjutnya, rangkaian tersebut akan diintegrasikan dengan Arduino Uno, yang berfungsi sebagai otak utama sistem. Arduino Uno akan mengoordinasikan operasi komponen-komponen tersebut, termasuk pengolahan data dari sensor ultrasonik, pengiriman data melalui modul WiFi, dan kontrol indikasi visual menggunakan LED.

Setelah komponen-komponen terhubung dan terintegrasi dengan baik, tahap selanjutnya adalah penempatan rangkaian *hardware* ini di atas tiap-tiap slot parkir. Sensor ultrasonik akan bertanggung jawab untuk mendeteksi keberadaan kendaraan di setiap slot parkir. NodeMCU akan mengambil data dari sensor dan mengirimkannya ke *database online* melalui koneksi WiFi. LED akan ditempatkan di atas sensor untuk memberikan indikasi visual mengenai status ketersediaan tempat parkir, sehingga pengguna dapat dengan mudah mengetahui apakah slot parkir tersebut kosong atau terisi.

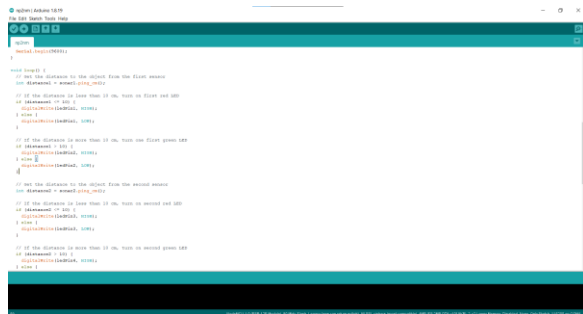
#### b) Pengembangan *Software*

Tahap pengembangan perangkat lunak dalam proyek ini memiliki peran sentral dalam memastikan keberhasilan sistem monitoring ketersediaan tempat parkir. Langkah pertama adalah pembuatan program menggunakan Arduino IDE. Program ini dirancang khusus untuk mengontrol dan mengoordinasikan interaksi antara berbagai komponen perangkat keras, termasuk sensor ultrasonik, NodeMCU, LED, dan Arduino Uno. Fungsi utama program adalah



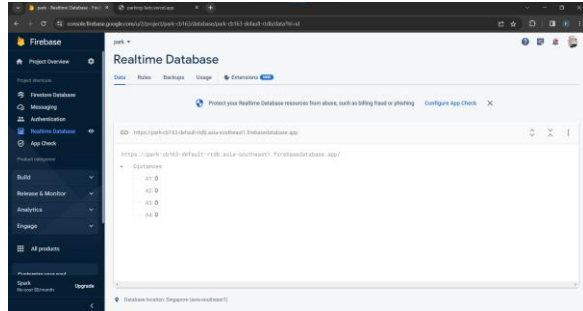
mengambil data yang diberikan oleh sensor ultrasonik, mengatur status LED sesuai kondisi parkir, dan mentransmisikan data ke *database* melalui koneksi WiFi yang ada.

Selain itu, program juga harus dioptimalkan untuk memastikan pengiriman data yang akurat dan responsif. Sensor ultrasonik harus mampu membaca data dengan tepat, NodeMCU harus efisien dalam mengirim data ke database, dan seluruh sistem harus mampu memberikan respons dalam waktu nyata. Gambar 3.6 menunjukkan tampilan dari program Arduino yang telah dibuat. Sedangkan pada Gambar 3.7 menunjukkan tampilan konfigurasi pada *database* Firebase.



```
1 // Arduino IDE
2 // File Name: Ultrasonic
3
4 #include <Ultrasonic.h>
5 #include <WiFi.h>
6 #include <Firebase.h>
7 #include <Arduino.h>
8
9 #define TRIG_PIN 12
10 #define ECHO_PIN 11
11 #define LED_PIN 13
12 #define DB_HOST "firebaseio.com"
13 #define DB_PATH "/parking"
14 #define DB_NAME "parking"
15 #define DB_USER "root"
16 #define DB_PASS "password"
17
18 Ultrasonic sensor;
19 WiFiClient client;
20 FirebaseData firebaseData;
21
22 void setup() {
23   // Set the distance to the object from the first sensor
24   sensor = Ultrasonic(12, 11);
25
26   // If the distance is less than 10 cm, turn on first red LED
27   pinMode(LED_PIN, OUTPUT);
28   digitalWrite(LED_PIN, LOW);
29 }
30
31 void loop() {
32   // If the distance is more than 10 cm, turn on first green LED
33   digitalWrite(LED_PIN, HIGH);
34   delay(100);
35 }
36
37 // If the distance is less than 10 cm, turn on second green LED
38 digitalWrite(LED_PIN, LOW);
39 delay(100);
40 }
```

Gambar 3. 5 Gambar Program Arduino UNO

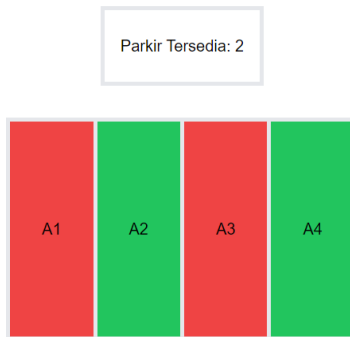


Gambar 3. 6 Gambar Konfigurasi Databae Firebase

### c) Pengembangan Sistem Web

Pengembangan sietem *web* mencakup bebrapa tahapan penting yaitu. Pertama, pembuatan *database* dengan Firebase. *Database online* akan dibuat menggunakan Firebase. Data yang dikirim oleh mikrokontroler akan disimpan di sini untuk diakses secara *realtime* melalui tampilan web.

Kedua, pembuatan tampilan *web* melalui Visual Studio Code. Dengan bantuan Visual Studio Code, tampilan *web* akan dibangun untuk menampilkan status ketersediaan tempat parkir secara *realtime*. Data yang diambil dari *database* Firebase akan diintegrasikan ke dalam tampilan *web* ini. Gambar 3. 8 menunjukkan ilustrasi tampilan antarmuka pada *web* yang akan dibuat.



*Gambar 3. 7 Ilustrasi Tampilan Antarmuka Web*

#### 4. Tahap Implementasi (*Implementation*)

Pada tahap implementasi ini, produk yang telah dikembangkan akan dibuat ke dalam sebuah miniatur lahan parkir. Setiap komponen *hardware*, termasuk sensor ultrasonik, modul WiFi NodeMCU Esp8266, LED, dan Arduino Uno akan dipasang secara fisik di setiap slot parkir yang ada dalam miniatur.

Pertama-tama, dilakukan pemotongan styrofoam menggunakan *cutter*. Hal ini dilakukan untuk membentuk miniatur lahan parkir sesuai dengan desain yang telah direncanakan. Proses pemotongan harus dilakukan dengan presisi agar miniatur tampak representatif.

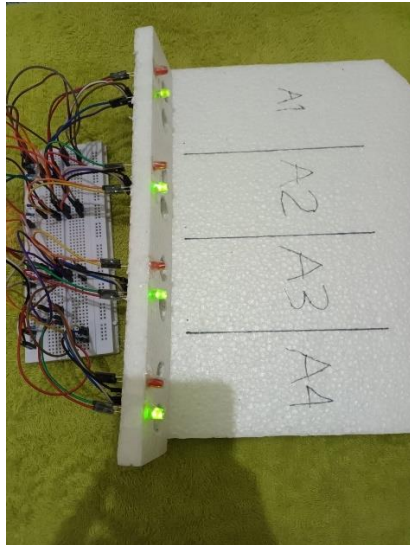
Selanjutnya, akan dilakukan pembuatan tempat untuk meletakkan sensor ultrasonik dengan bantuan solder. Ini akan memastikan hasil yang serapih mungkin dalam penempatan sensor.

Kemudian, LED akan ditempatkan di atas tiap sensor ultrasonik. Langkah ini bertujuan untuk memberikan indikasi visual tentang status ketersediaan tempat parkir kepada pengguna. Pencerahan LED akan berubah sesuai dengan kondisi sensor, memudahkan pengguna dalam menentukan ketersediaan tempat parkir.

Terakhir, pembuatan garis pemandu pada miniatur lahan parkir akan dilakukan. Garis-garis ini akan membantu dalam menunjukkan lokasi tiap slot parkir dengan lebih jelas. Penggunaan pulpen atau spidol akan memastikan pembuatan garis pemandu yang presisi.

Dengan menyelesaikan tahap implementasi ini, miniatur lahan parkir akan siap digunakan sebagai bagian dari sistem monitoring ketersediaan tempat parkir. Ini memungkinkan pengujian dan pengujian sistem sebelum implementasi dalam skenario yang sebenarnya, serta memastikan bahwa miniatur tersebut memenuhi standar yang diperlukan. Gambar

3. 9 menunjukkan miniatur tempat parkir yang telah dibuat.



*Gambar 3. 8 Gambar Miniatur Tempat Parkir*

#### 5. Tahap Evaluasi (*Evaluation*)

Tahap evaluasi merupakan langkah penting dalam pengembangan proyek ini. Evaluasi bertujuan untuk menilai kualitas produk yang telah diimplementasikan pada tahap implementasi dan untuk mengidentifikasi potensi kekurangan dan kelemahan yang mungkin masih ada dalam sistem. (Branch, 2009). Ini membantu dalam memastikan bahwa produk yang dikembangkan memenuhi standar kualitas dan efektivitas yang diharapkan.

Peneliti melakukan uji fungsional terhadap produk yang telah dikembangkan dengan fokus pada hasil *output* yang dihasilkan dari *input* yang diberikan. Metode evaluasi yang digunakan adalah metode *black box*, yang melibatkan pengujian perkomponen dan perfitur baik dari *hardware* maupun *software*.

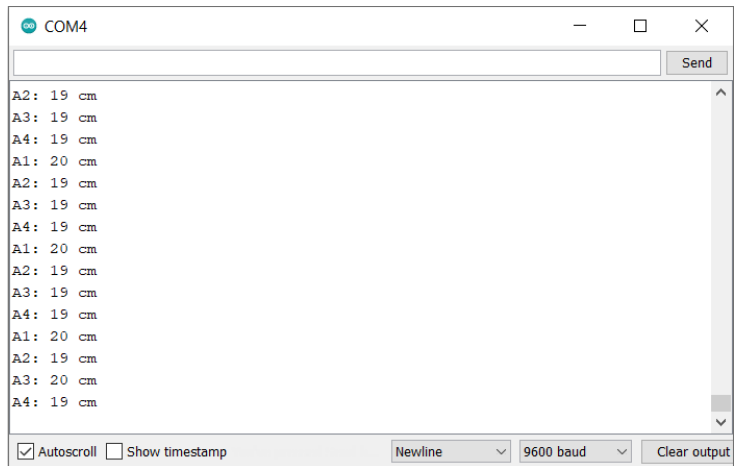
Uji *black box* pada *hardware* melibatkan pengujian kinerja Sensor Ultrasonik HC-SR04. Uji ini akan memverifikasi apakah sensor mampu membaca objek pada jarak tertentu atau tidak. Selain itu, uji kinerja NodeMCU Esp8266 juga akan dilakukan untuk memastikan apakah modul ini dapat mengirimkan data melalui koneksi WiFi ke *database* dengan baik.

Pada sisi *software*, uji *black box* akan mencakup uji fitur pada aplikasi yang digunakan. Uji ini akan memverifikasi apakah aplikasi dapat tersambung dengan sukses ke *database* melalui koneksi WiFi. Seluruh proses pengiriman data dan interaksi antara aplikasi dan *database* akan dievaluasi untuk memastikan bahwa sistem berjalan sesuai dengan yang direncanakan.

#### 1. Pengujian *Hardware*

- a) Pengujian pertama yaitu menguji apakah tiap sensor bisa bekerja secara optimal dalam

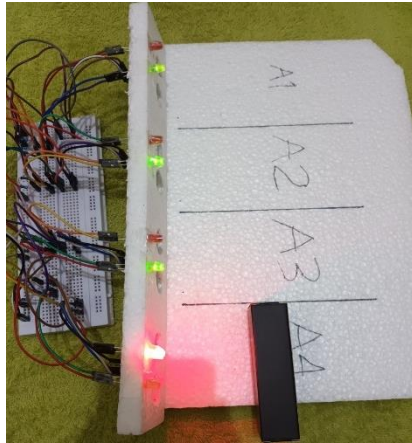
membaca jarak objek, dalam pengujian ini peneliti memanfaatkan fitur serial monitor pada aplikasi Arduino IDE, dan sensor akan meng-*update* tiap detik. Dapat dilihat pada Gambar 3. 10 fitur serial monitor yang menunjukkan data pembacaan jarak pada sensor.



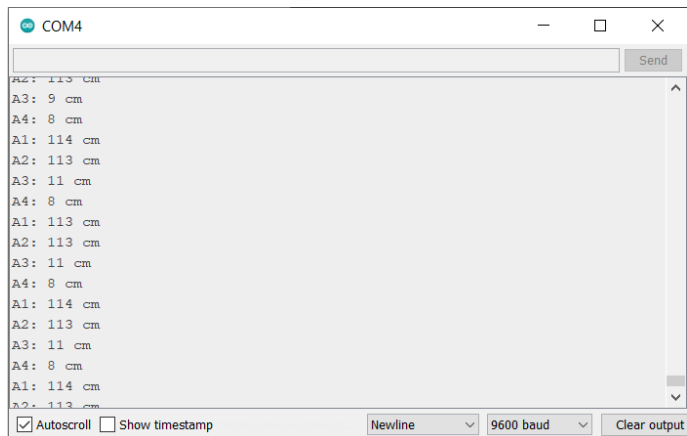
*Gambar 3. 9 Gambar Serial Monitor*

b) Pengujian kedua yaitu apakah tiap sensor dapat berkomunikasi dengan LED, dalam pengujian ini peneliti meletakkan sebuah objek di masing-masing sensor secara bergantian. Peneleiti menetapkan dimana jika jarak objek dengan sensor kurang dari atau sama dengan 10 cm, maka LED pada sensor harus berubah merah dan jika lebih dari 10 maka LED harus berwarna hijau.

Gambar 3. 11 menunjukkan komunikasi sensor dengan LED. Dan pada gambar 3. 12 menunjukkan data pembacaan pada serial monitor.



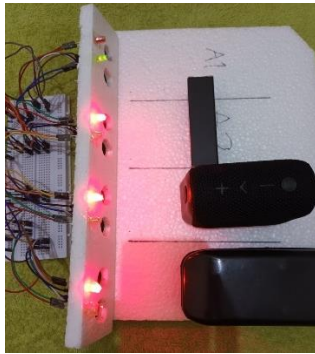
*Gambar 3. 10 Pengujian Komunikasi Sensor dengan LED*



*Gambar 3. 11 Gambar Pemantauan di Serial Monitor*

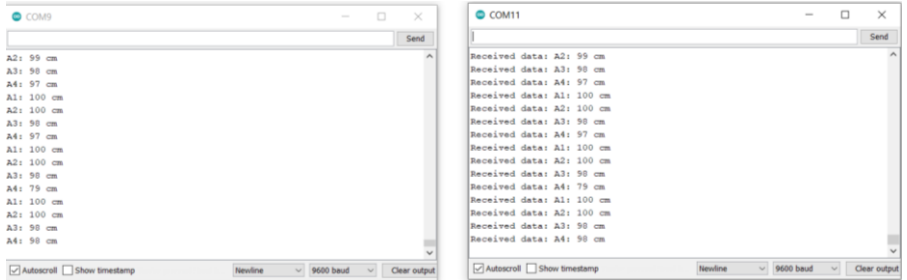


Penguji juga melihat dari serial monitor untuk memastikan jaraknya akurat. Kemudian peneliti meletakkan beberapa objek sekaligus di depan rangkaian *hardware* dan didapatkan hasil sebagai berikut. Gambar 3. 13 menunjukkan pengujian komunikasi sensor dengan LED menggunakan beberapa objek.



*Gambar 3. 12 Gambar Pengujian dengan Beberapa Objek*

c) Pengujian ketiga yaitu mengirim hasil dari Arduino ke Nodemcu atau yang bisa disebut dengan komunikasi serial. Penguji akan memanfaatkan serial monitor di masing-masing program. Pada Gambar 3. 14 menunjukkan serial komunikasi antara Arduino dengan NodeMCU.

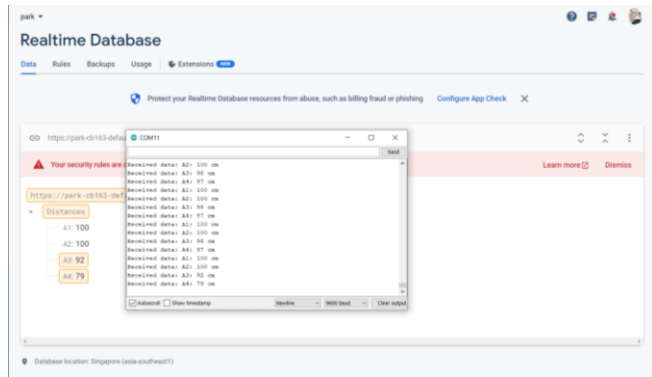


Gambar 3. 13 Gambar serial komunikasi antara Arduino dengan NodeMCU

a. Hasil output Arduino.

b. Hasil output dari NodeMCU.

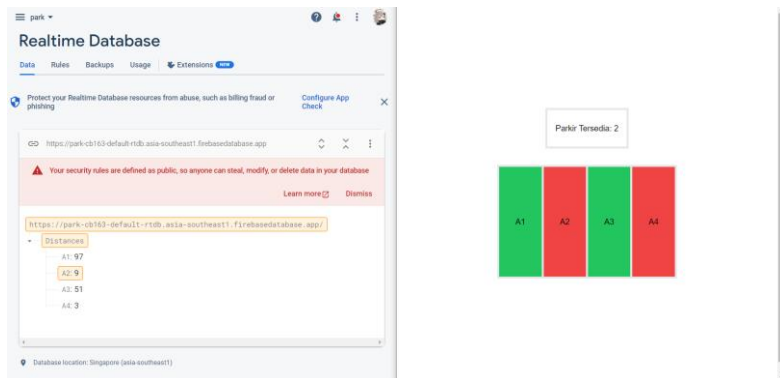
d) Pengujian keempat yaitu mengirim data yang diterima oleh NodeMcu dari Arduino ke *database* Firebase. Peneliti akan memastikan langsung sinkronisasi data dari NodeMCU maupun Firebase. Gambar 3.15 menunjukkan sinkronisasi data antara NodeMCU dengan Firebase.



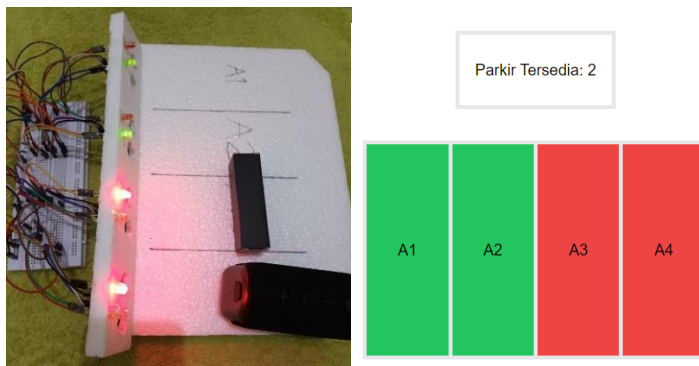
Gambar 3. 14 Gambar sinkronisasi antara Firebase dengan NodeMCU

## 2. Pengujian Tampilan *Web*

Pengujian ini berupa pengambilan data dari *database* Firebase untuk tampilan *web*. Peneliti akan memastikan bahwa data dapat terkirim dan *web* akan bekerja seperti seharusnya, dimana jika jika data di *database* menunjukkan kurang dari atau sama dengan 10 cm maka slot parkir yang mendeteksi objek tersebut akan berwarna merah, serta akan menunjukkan jumlah slot parkir yang tersedia. Pada Gambar 3. 16 menunjukkan pengujian *web* sedangkan Gambar 3. 17 menunjukkan pengujian seara *realtime*.



Gambar 3. 15 Gambar pengujian *web*



*Gambar 3. 16 Gambar Pengujian Secara Realtime*



## BAB IV

### HASIL PENELITIAN DAN PEMBAHASAN

#### A. Hasil Analisis

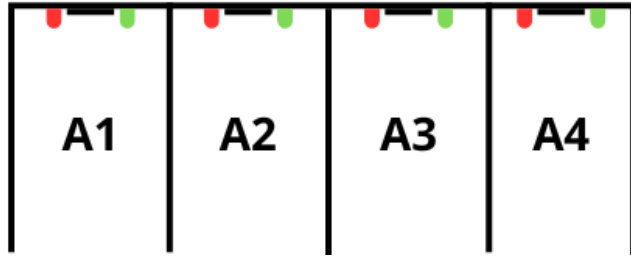
Untuk komponen *hardware*, kita memerlukan beberapa elemen kunci. Pertama adalah Sensor Ultrasonik HC-SR04 yang akan digunakan untuk mendeteksi keberadaan kendaraan di setiap slot parkir. Selanjutnya, kita memerlukan modul WiFi NodeMCU Esp8266 yang akan bertanggung jawab untuk mengirimkan data yang dikumpulkan oleh sensor ke *database online*. Terakhir, Arduino Uno akan bertindak sebagai otak utama dalam sistem, mengendalikan operasi seluruh perangkat keras dan berkoordinasi dengan perangkat lunak.

Dalam aspek perangkat lunak, beberapa perangkat dan media yang dibutuhkan. Pertama, kita menggunakan Arduino IDE untuk memprogram perangkat keras, terutama dalam mengembangkan program untuk Arduino Uno. Selain itu, Visual Studio Code digunakan untuk mengembangkan tampilan *web* yang akan menjadi antarmuka pengguna dalam proyek ini. Terakhir, Firebase berperan sebagai media penyimpanan data *online* untuk menyimpan data yang dikumpulkan oleh sistem.

Untuk membuat prototipe miniatur lahan parkir yang representatif, perlengkapan alat tulis kantor diperlukan. Styrofoam menjadi bahan dasar untuk membentuk struktur fisik miniatur, yang akan dibentuk dengan presisi menggunakan *cutter*. Pengukuran yang akurat dilakukan dengan bantuan pengaris, sedangkan pulpen dan spidol digunakan untuk memberikan tanda atau garis pada styrofoam agar pemotongan dan penempatan komponen berjalan sesuai rencana. Terakhir, soldier digunakan untuk melubangi styrofoam demi menempatkan sensor ultrasonik dengan presisi.

## **B. Hasil Desain**

Dalam desain akhir ini, sistem akan bekerja sebagai berikut: sensor yang dipasang di belakang setiap slot parkir akan berfungsi untuk mendeteksi apakah slot tersebut terisi oleh mobil atau tidak. Lalu hasilnya akan ditampilkan dengan LED yang menyala warna merah ataupun hijau. Pada Gambar 4.1 menunjukkan denah desain pada miniature yang dibuat.



*Gambar 4. 1 Denah Desain Miniatur*

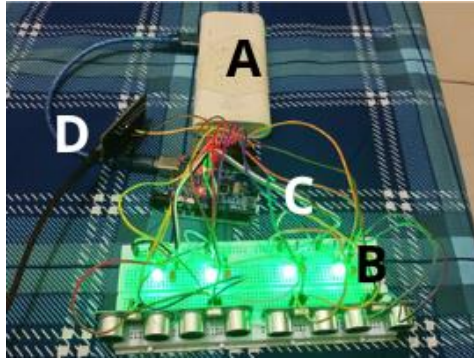
Yang dimana alur kerja sistem ini adalah yang pertama sensor yang berada di belakang tiap slot parkir akan membaca slot parkiri terisi atau tidak. Kemudian hasilnya akan ditunjukkan dengan nyala lampu LED. Setelah itu data akan diunggah ke Firebase yang nantinya sebagai data yang akan ditampilkan pada tampilan *web*.

### **C. Hasil Pengembangan**

#### *1. Pengembangan Hardware*

Hasil yang didapatkan dari pengembangan hardware ini berupa rangkaian sederhana yang kemudian akan diimplementasikan pada miniatur tempat parkir.





*Gambar 4. 2 Rangkaian Hardware Sederhana*

Pada Gambar 4.2 menunjukkan rangkaian hardware sederhana yang dimana pada huruf A adalah power bank yang digunakan sebagai sumber listrik utama pada rangkaian ini, kemudian pada huruf B adalah breadboard sebagai media yang menyatukan berbagai komponen hardware, pada huruf C adalah Arduino sebagai pengendali utama pada rangkaian ini, dan huruf D adalah NodeMCU sebagai modul wifi yang akan mengunggah data ke Firebase.

## 2. Pengembangan Software

Pada pengembangan software, peneliti membuat fungsi yang dimana jika jarak objek dari sensor kurang dari atau sama dengan 10 cm, maka sensor akan mengartikan kalau di slot parkir tersebut terisi dan akan menghidupkan LED merah.

```
// If the distance is less than 10 cm, turn on first red LED
if (distance1 <= 10) {
  digitalWrite(ledPin1, HIGH);
} else {
  digitalWrite(ledPin1, LOW);
}
```

```
// If the distance is more than 10 cm, turn one first green LED
if (distance1 > 10) {
  digitalWrite(ledPin2, HIGH);
} else {
  digitalWrite(ledPin2, LOW);
}
```

### 3. Pengembangan Sistem *Web*

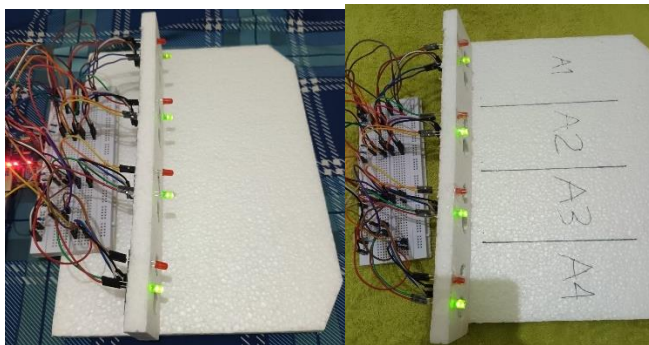
Pada pengembangan *web* peneliti berhasil membuat tampilan *web* yang akan berguna sebagai media pembantu yang akan memudahkan pengguna untuk menemukan slot parkir yang kosong dari mana saja, karena *web* ini sudah *online* dan bisa diakses di *browser* dengan memasukkan *link* <https://parking-faris.vercel.app>. Pada Gambar 4. 3 menunjukkan tampilan *web* pada *browser*.



*Gambar 4. 3 Gambar Tampilan Web pada Browser*

#### **D. Hasil Implementasi**

Pada tahap implementasi ini, produk yang telah dikembangkan akan dibuat ke dalam sebuah miniatur lahan parkir. Setiap komponen *hardware*, termasuk sensor ultrasonik, modul WiFi NodeMCU Esp8266, LED, dan Arduino Uno akan dipasang secara fisik di setiap slot parkir yang ada dalam miniatur. Pada Gambar 4. 4 menunjukkan miniatur tempat parkir yang telah dibuat.



*Gambar 4. 4 Gambar Miniatur yang Telah Dibuat*

## E. Hasil Evaluasi

### 1. Pengujian Kinerja Sensor

Metode pengujian yang akan dilakukan adalah peneliti meletakkan objek di depan sensor dengan jarak 20 cm menggunakan bantuan penggaris untuk lebih akurat, kemudian melihat hasil pembacaan data jarak yang terbaca pada tiap sensor dan dibandingkan dengan jarak sebenarnya.

Tabel 4. 1 Hasil Uji Sensor 1.

No	Jarak Sebenarnya	Hasil Pada Sensor	Error
1	20	20	0
2	20	20	0
3	20	20	0
4	20	20	0
5	20	20	0
6	20	20	0
7	20	20	0
8	20	20	0
9	20	20	0
10	20	20	0

Pada pengujian sensor 1 tidak ada perbedaan hasil pada sensor dengan jarak sebenarnya, sehingga didapatkan akurasi sebagai berikut:

Menggunakan rumus :

$$\frac{\text{Rata – rata hasil pada sensor}}{\text{Jarak sebenarnya}} \times 100\%$$

$$\frac{20}{20} \times 100\% = 100\%$$

Dengan presentase *error* sebesar :

$$\frac{\text{Rata - rata error}}{\text{Jarak sebenarnya}} \times 100\%$$

$$\frac{0}{20} \times 100\% = 100\%$$

Tabel 4. 2 Hasil Uji Sensor 2.

No	Jarak Sebenarnya	Hasil Pada Sensor	Error
1	20	20	0
2	20	20	0
3	20	20	0
4	20	20	0
5	20	20	0
6	20	20	0
7	20	20	0
8	20	20	0
9	20	20	0
10	20	20	0

Pada pengujian sensor 2 tidak ada perbedaan hasil pada sensor dengan jarak sebenarnya, sehingga didapatkan akurasi sebagai berikut:

Menggunakan rumus :

$$\frac{\text{Rata - rata hasil pada sensor}}{\text{Jarak sebenarnya}} \times 100\%$$

$$\frac{20}{20} \times 100\% = 100\%$$

Dengan presentase *error* sebesar :

$$\frac{\text{Rata - rata error}}{\text{Jarak sebenarnya}} \times 100\%$$

$$\frac{0}{20} \times 100\% = 100\%$$

Tabel 4. 3 Hasil Uji Sensor 3.

No	Jarak Sebenarnya	Hasil Pada Sensor	Error
1	20	20	0
2	20	20	0
3	20	19	1
4	20	19	1
5	20	20	0
6	20	19	1
7	20	20	0
8	20	20	0
9	20	20	0
10	20	20	0

Pada pengujian sensor 3 ada perbedaan hasil pada sensor dengan jarak sebenarnya, sehingga didapatkan akurasi sebagai berikut:

Menggunakan rumus :

$$\frac{\text{Rata - rata hasil pada sensor}}{\text{Jarak sebenarnya}} \times 100\%$$

$$\frac{19,7}{20} \times 100\% = 98,5\%$$

Dengan presentase *error* sebesar :

$$\frac{\text{Rata - rata error}}{\text{Jarak sebenarnya}} \times 100\%$$

$$\frac{0,3}{20} \times 100\% = 1.5\%$$

Tabel 4. 4 Hasil Uji Sensor 4.

No	Jarak Sebenarnya	Hasil Pada Sensor	Error
1	20	19	1
2	20	20	0
3	20	19	1
4	20	19	1
5	20	20	0
6	20	20	0
7	20	19	1
8	20	20	0
9	20	20	0
10	20	20	0

Pada pengujian sensor 4 ada perbedaan hasil pada sensor dengan jarak sebenarnya, sehingga didapatkan akurasi sebagai berikut:

Menggunakan rumus :

$$\frac{\text{Rata - rata hasil pada sensor}}{\text{Jarak sebenarnya}} \times 100\%$$

$$\frac{19,6}{20} \times 100\% = 98\%$$

Dengan presentase *error* sebesar :

$$\frac{\text{Rata - rata error}}{\text{Jarak sebenarnya}} \times 100\%$$

$$\frac{0,4}{20} \times 100\% = 2\%$$

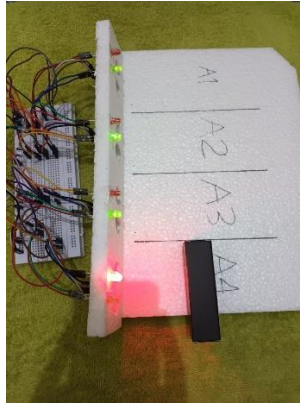
## 2. Pengujian Sistem Monitoring

### a. Pengujain sistem dengan satu objek

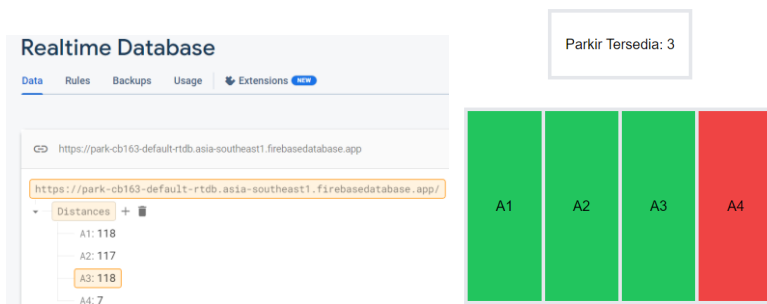
#### 1) Kondisi 1

Pada kondisi 1 dimana objek diletakkan pada slot A4 dan didapatkan hasil berupa pada Gambar 4.1 yang menunjukkan jika LED pada slot A4 berwarna merah yang dimana menunjukkan bahwa slot A4 sedang terisi. Gambar 4. 2 menunjukkan data pembacaan jarak pada Firebase yang kemudian ditampilkan pada *web* yang menunjukkan bahwa slot A4 juga sedang terisi.





*Gambar 4. 5 Sistem Hardware Satu Objek Kondisi 1*

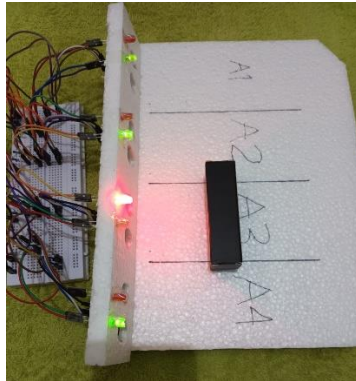


*Gambar 4. 6 Sistem Software Satu Objek Kondisi 1*

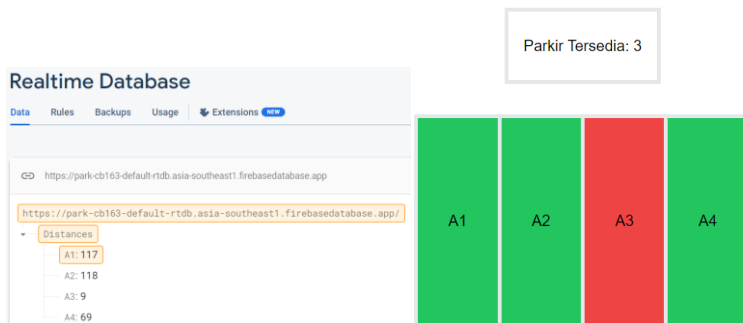
2) Kondisi 2

Pada kondisi 2 dimana objek diletakkan pada slot A3 dan didapatkan hasil berupa pada Gambar 4.3 yang menunjukkan jika LED pada slot A3 berwarna merah yang dimana menunjukkan bahwa slot A3

sedang terisi. Gambar 4. 4 menunjukkan data pembacaan jarak pada Firebase yang kemudian ditampilkan pada *web* yang menunjukkan bahwa slot A3 juga sedang terisi.



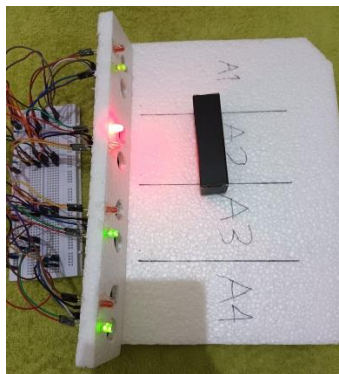
*Gambar 4. 7 Sistem Hardware Satu Objek Kondisi 2*



*Gambar 4. 8 Sistem Software Satu Objek Kondisi 2*

### 3) Kondisi 3

Pada kondisi 3 dimana objek diletakkan pada slot A2 dan didapatkan hasil berupa pada Gambar 4.5 yang menunjukkan jika LED pada slot A2 berwarna merah yang dimana menunjukkan bahwa slot A2 sedang terisi. Gambar 4. 6 menunjukkan data pembacaan jarak pada Firebase yang kemudian ditampilkan pada *web* yang menunjukkan bahwa slot A2 juga sedang terisi.



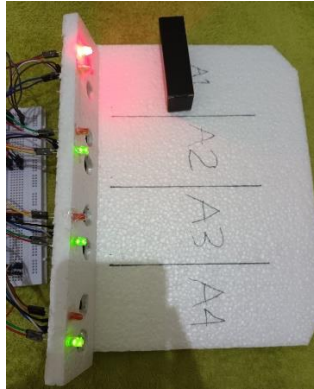
*Gambar 4. 9 Sistem Hardware Satu Objek Kondisi 3*



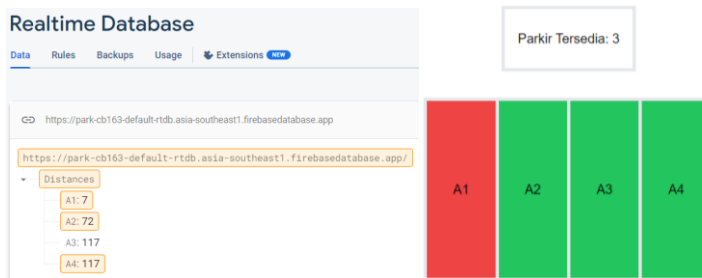
Gambar 4. 10 Sistem Software Satu Objek Kondisi 3

#### 4) Kondisi 4

Pada kondisi 4 dimana objek diletakkan pada slot A1 dan didapatkan hasil berupa pada Gambar 4.7 yang menunjukkan jika LED pada slot A1 berwarna merah yang dimana menunjukkan bahwa slot A1 sedang terisi. Gambar 4. 8 menunjukkan data pembacaan jarak pada Firebase yang kemudian ditampilkan pada *web* yang menunjukkan bahwa slot A1 juga sedang terisi.



Gambar 4. 11 Sistem Hardware Satu Objek Kondisi 4



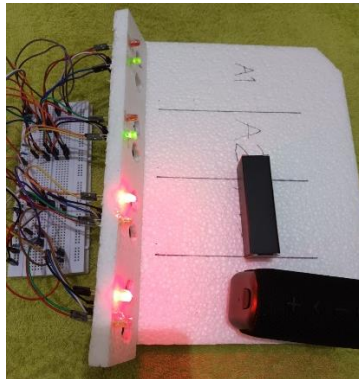
Gambar 4. 12 Sistem Software Satu Objek Kondisi 4

b. Pengujian sistem dengan dua objek

a) Kondisi 1

Pada kondisi 1 dimana objek diletakkan pada slot A4 dan A3 lalu didapatkan hasil berupa pada Gambar 4.9 yang menunjukkan jika LED pada slot A4 dan A3 berwarna merah yang dimana

menunjukkan bahwa slot A4 dan A3 sedang terisi. Gambar 4. 10 menunjukkan data pembacaan jarak pada Firebase yang kemudian ditampilkan pada *web* yang menunjukkan bahwa slot A4 dan A3 juga sedang terisi.



*Gambar 4. 13 Sistem Hardware Dua Objek Kondisi 1*

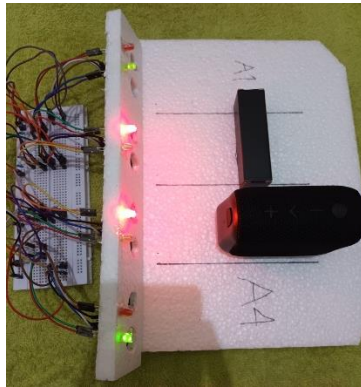
Slot	Status
A1	Available (Green)
A2	Available (Green)
A3	Occupied (Red)
A4	Occupied (Red)

Parkir Tersedia: 2

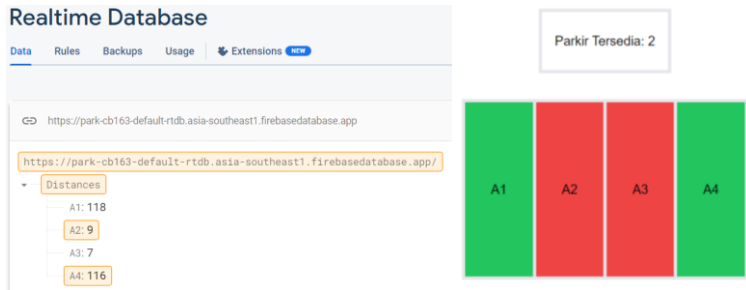
*Gambar 4. 14 Sistem Software Dua Objek Kondisi 1*

b) Kondisi 2

Pada kondisi 2 dimana objek diletakkan pada slot A3 dan A2 lalu didapatkan hasil berupa pada Gambar 4.11 yang menunjukkan jika LED pada slot A3 dan A2 berwarna merah yang dimana menunjukkan bahwa slot A3 dan A2 sedang terisi. Gambar 4. 12 menunjukkan data pembacaan jarak pada Firebase yang kemudian ditampilkan pada *web* yang menunjukkan bahwa slot A3 dan A2 juga sedang terisi.



*Gambar 4. 15 Sistem Hardware Dua Objek Kondisi 2*

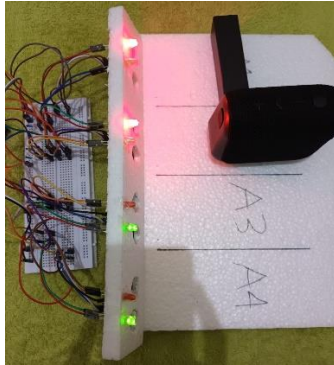


Gambar 4. 16 Sistem Software Dua Objek Kondisi 2

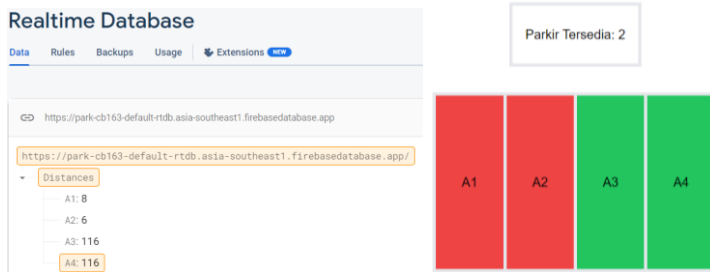
c) Kondisi 3

Pada kondisi 3 dimana objek diletakkan pada slot A2 dan A1 lalu didapatkan hasil berupa pada Gambar 4.13 yang menunjukkan jika LED pada slot A2 dan A1 berwarna merah yang dimana menunjukkan bahwa slot A2 dan A1 sedang terisi. Gambar 4. 14 menunjukkan data pembacaan jarak pada Firebase yang kemudian ditampilkan pada *web* yang menunjukkan bahwa slot A2 dan A1 juga sedang terisi.





*Gambar 4. 17 Sistem Hardware Dua Objek Kondisi 3*

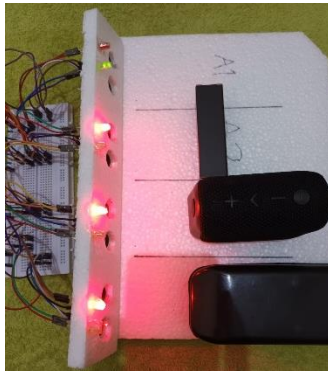


*Gambar 4. 18 Sistem Software Dua Objek Kondisi 3*

- c. Pengujian sistem dengan tiga objek
  - a) Kondisi 1

Pada kondisi 1 dimana objek diletakkan pada slot A4, A3, dan A2 lalu didapatkan hasil berupa pada Gambar 4.15 yang

menunjukkan jika LED pada slot A4, A3, dan A2 berwarna merah yang dimana menunjukkan bahwa slot A4, A3, dan A2 sedang terisi. Gambar 4. 16 menunjukkan data pembacaan jarak pada Firebase yang kemudian ditampilkan pada *web* yang menunjukkan bahwa slot A4, A3, dan A2 juga sedang terisi.



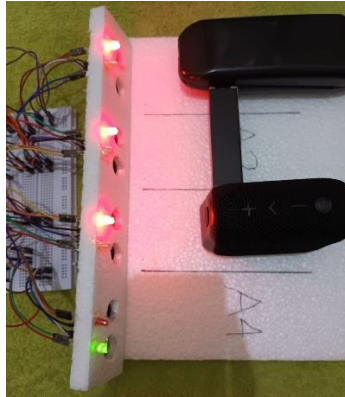
*Gambar 4. 19 Sistem Hardware Tiga Objek Kondisi 1*



Gambar 4. 20 Sistem Software Tiga Objek Kondisi 1

#### b) Kondisi 2

Pada kondisi 2 dimana objek diletakkan pada slot A3, A2, dan A1 lalu didapatkan hasil berupa pada Gambar 4.17 yang menunjukkan jika LED pada slot A3, A2, dan A1 berwarna merah yang dimana menunjukkan bahwa slot A3, A2, dan A1 sedang terisi. Gambar 4. 18 menunjukkan data pembacaan jarak pada Firebase yang kemudian ditampilkan pada *web* yang menunjukkan bahwa slot A3, A2, dan A1 juga sedang terisi.



Gambar 4. 21 Sistem Hardware Tiga Objek Kondisi 2

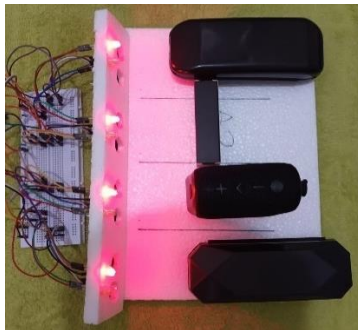


Gambar 4. 22 Sistem Software Tiga Objek Kondisi 2

d. Pengujian dengan empat objek

Objek diletakkan pada slot A4, A3, A2, dan A1 lalu didapatkan hasil berupa pada Gambar 4.19 yang menunjukkan jika LED

pada slot A4, A3, A2, dan A1 berwarna merah yang dimana menunjukkan bahwa slot A4, A3, A2, dan A1 sedang terisi. Gambar 4. 20 menunjukkan data pembacaan jarak pada Firebase yang kemudian ditampilkan pada *web* yang menunjukkan bahwa slot A4, A3, A2, dan A1 juga sedang terisi.



*Gambar 4. 23 Sistem Hardware Empat Objek*



Gambar 4. 24 Sistem Software Empat Objek

## F. Rekapitulasi Hasil Pengujian

### 1. Hasil Pengujian Sistem Monitoring

Tabel 4. 5 Hasil pengujian dengan satu objek.

	Kondisi 1	Kondisi 2	Kondisi 3	Kondisi 4
Slot 1	Kosong	Kosong	Kosong	Isi
Slot 2	Kosong	Kosong	Isi	Kosong
Slot 3	Kosong	Isi	Kosong	Kosong
Slot 4	Isi	Kosong	Kosong	Kosong
LED 1	Hijau on Merah off	Hijau on Merah off	Hijau on Merah off	Hijau off Merah on
LED 2	Hijau on Merah off	Hijau on Merah off	Hijau off Merah on	Hijau on Merah off
LED 3	Hijau on Merah off	Hijau off Merah on	Hijau on Merah off	Hijau on Merah off
LED 4	Hijau off Merah on	Hijau on Merah off	Hijau on Merah off	Hijau on Merah off

A1	Hijau	Hijau	Hijau	Merah
A2	Hijau	Hijau	Merah	Hijau
A3	Hijau	Merah	Hijau	Hijau
A4	Merah	Hijau	Hijau	Hijau
KETERANGAN	Berhasil	Berhasil	Berhasil	Berhasil

Tabel 4. 6 Hasil pengujian dengan dua objek.

	Kondisi 1	Kondisi 2	Kondisi 3
Slot 1	Kosong	Kosong	Isi
Slot 2	Kosong	Isi	isi
Slot 3	Isi	isi	Kosong
Slot 4	isi	Kosong	Kosong
LED 1	Hijau on Merah off	Hijau on Merah off	Hijau off Merah on
LED 2	Hijau on Merah off	Hijau off Merah on	Hijau off Merah on
LED 3	Hijau off Merah on	Hijau off Merah on	Hijau on Merah off
LED 4	Hijau off Merah on	Hijau on Merah off	Hijau on Merah off
A1	Hijau	Hijau	Merah
A2	Hijau	Merah	Merah
A3	Merah	Merah	Hijau
A4	Merah	Hijau	Hijau
KETERANGAN	Berhasil	Berhasil	Berhasil

Tabel 4. 7 Hasil pengujian dengan tiga objek.

	Kondisi 1	Kondisi 2
Slot 1	Kosong	Isi
Slot 2	Isi	Isi
Slot 3	Isi	Isi
Slot 4	Isi	Kosong
LED 1	Hijau on	Hijau off

	Merah off	Merah on
LED 2	Hijau off Merah on	Hijau off Merah on
LED 3	Hijau off Merah on	Hijau off Merah on
LED 4	Hijau off Merah on	Hijau on Merah off
A1	Hijau	Merah
A2	Merah	Merah
A3	Merah	Merah
A4	Merah	Hijau
KETERANGAN	Berhasil	Berhasil

Tabel 4. 8 Hasil pengujian dengan empat objek.

	Kondisi
Slot 1	Merah
Slot 2	Merah
Slot 3	Merah
Slot 4	Merah
LED 1	Hijau off Merah on
LED 2	Hijau off Merah on
LED 3	Hijau off Merah on
LED 4	Hijau off Merah on
A1	Merah
A2	Merah
A3	Merah
A4	Merah
KETERANGAN	Berhasil



Pada pengujian sitem monitoring, peneliti akhirnya bisa mendapatkan presensate keberhasilan tiap skenario sebagai berikut:

Menggunakan rumus presentase keberhasilan :

$$\frac{\text{Jumlah berhasil}}{\text{Jumlah pengujian}} \times 100\%$$

$$\frac{10}{10} \times 100\% = 100\%$$

Diketahui bahwa presentase keberhasilan pengujian sitem monitoring ini mencapai 100% yang berarti peneliti telah berhasil membuat sistem monitoring tempat parkir yang lebih efisien.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **A. Kesimpulan**

Dalam penelitian ini, peneliti berhasil merancang dan membangun prototipe sistem monitoring ketersediaan tempat parkir yang dapat mengidentifikasi lokasi slot parkir yang kosong secara efisien dan memberikan kenyamanan kepada pengguna parkir.

Hasilnya, peneliti mendapatkan data bahwa performa kinerja sistem monitoring ini mencapai tingkat keberhasilan sempurna pada pengujian sistem monitoring, yaitu sebesar 100%. Hasil penelitian ini mencerminkan pencapaian yang signifikan dalam upaya meningkatkan efisiensi dan pengelolaan tempat parkir. Sistem ini memiliki potensi besar untuk mengatasi masalah kemacetan parkir dan memberikan pengalaman yang lebih baik bagi pengguna parkir. Penelitian ini membuktikan bahwa inovasi dalam sistem monitoring parkir adalah langkah yang relevan dan bermanfaat dalam menghadapi tantangan mobilitas perkotaan modern.

## **B. Saran**

Berdasarkan hasil penelitian yang telah dilakukan, peneliti ingin memberikan beberapa saran yang dapat meningkatkan kualitas dan kinerja sistem ini:

1. Penggunaan sensor dan hardware yang lebih berkualitas guna meningkatkan serta memastikan pengukuran ketersediaan tempat parkir lebih akurat.
2. Pemilihan power supply yang sesuai dengan kebutuhan hardware guna memastikan pasokan daya yang stabil dan memadai agar sistem berjalan dengan baik tanpa gangguan.
3. Integrasi yang lancar antara hardware dan software untuk memastikan bahwa perangkat keras dan perangkat lunak berfungsi bersama dengan baik.
4. Pengujian lebih detail terhadap komponen perangkat keras dan perangkat lunak untuk memastikan kinerja sistem dalam berbagai situasi.
5. Memahami setiap langkah dalam prosedur penelitian kunci untuk mendapatkan hasil yang sesuai dengan harapan.
6. Untuk penelitian selanjutnya, dapat meningkatkan jumlah sensor slot parkir yang dapat dimuat, dan menambahkan fungsi identifikasi jika objek yang berada di slot parkir itu bukan mobil.

## DAFTAR PUSTAKA

- Angko, Nancy dan Mustaji. 2013. Pengembangan Bahan Ajar dengan Model ADDIE untuk Mata Pelajaran Matematika Kelas 5 SDS Mawar Sharon Surabaya. *Jurnal KWANGSAN*. 1 (1): 1-15.
- A. Purwanto and S. Lutfi, "Pengendalian Lampu Rumah Berbasis Google Asisstant Melalui Smartphone Menggunakan NodeMCU-12E ESP8266 di Nuke Komputer Service," *J. Himsya Tech*, vol. 20, no. 2, pp. 1–6, 2019.
- Arsada, Bakhtiyar. 2017. "Aplikasi Sensor Ultrasonik Untuk Deteksi Posisi Jarak Pada Ruang Menggunakan Arduino Uno." *Jurnal Teknik Elektro* 6 (2): 1–8.
- Branch, R. M. (2009). *Instructional Design-The ADDIE Approach*. New York: Springer.
- Cahyadi, R. A. H. (2019). Pengembangan Bahan Ajar Berbasis Addie Model. *Halaqa: Islamic Education Journal*, 3(1), 35–42.  
<https://doi.org/10.21070/halaqa.v3i1.2124>

Derwin Mahardika. 2020. Perbedaan NodeMCU, Wemos, dan ESP8266 Wifi Module untuk Perangkat IoT Mikrokontroler. Online di <https://www.teknodika.com/2020/04/perbedaan-nodemcu-wemos-dan-esp8266.html>, diakses pada 20 Oktober 2022

Ditjen Perhubungan Darat. 1998. Pedoman Perencanaan dan Pengoperasian Fasilitas Parkir. Jakarta: Departemen Perhubungan.

D. Wibowo, T. Al Fit, R. Baihaqi, W. H. Sugiharto, and M. I. Ghozali. 2020. "Sistem Parkir Mobil Cerdas Menggunakan Citra Digital Dan Microcontroller Atmega328," J. Dialekt. Inform., vol. 1, no. 1, pp. 13–17.

Firestore Realtime Database. 2023. Dokumentasi Firestore Realtime Database. Online di <https://firebase.google.com/docs/database?hl=id>, diakses pada 9 September 2023.

Heru Purwanto, Malik Riyadi, Destiana Windi Widi Astuti, I Wayan Angga Wijaya Kusuma. 2019. "Komparasi Sensor Ultrasonik HC-SR04 Dan JSN-SR04T Untuk Aplikasi Sistem Deteksi Ketinggian Air." *Jurnal SIMETRIS* 10 (2): 717–24.

- H. I Hsieh and H Wang. "LED Current Balance Using a Variable Voltage. Regulator with Low Dropout DS Control", *Applied. Sciences.* , pp 1- 14, 2017
- H. Nayomi, *Peluang Pemanfaatan Lampu LED Sebagai Sumber Penerangan*, Depok: Universitas Indonesia, 2013
- J. H. Saputro, T. Sukmadi and Karnoto, "Analisa Penggunaan Lampu LED Pada Penerangan Dalam Rumah," *Transmisi*, vol. 1, no. 15, pp. 19-27, 2013.
- Kadir, Abdul. 2012. *Panduan Praktis Mempelajari Aplikasi Mikrokontroler dan Pemrogramannya Menggunakan Arduino*. Yogyakarta: Andi offset.
- Kianpisheh, Amin. 2012. *Smart Parking System (SPS) Architecture Using Ultrasonik Detection*, *Universiti Saints Malaysia. International Journal of Software Engineering and Its Applications* Vol. 6. No. 3. July. 2012.
- K, Shihabudin Achmad Muhajir A, and Safrina Amini. 2016. "Sistem Monitoring Tempat Parkir Dengan Sensor Ultrasonik Berbasis Arduino Uno Pada Cibinong City Mall." *Seniati*, 350-55.

Miro, F. 1997. Sistem Transportasi Kota Bandung: Bandung: Tarsito Bandung

Mirza, Y., Deviana, H., & Teknik Komputer Politeknik Negeri Sriwijaya Palembang, J. (2020). Sistem Monitoring Parkir Mobil Berbasis Mikrokontroler Arduino Uno. *12 Jurnal JUPITER*, 12(2), 12–25.

Muhammad Habib Al Khairi. 2020. Tutorial Menggunakan Sensor Ultrasonik HC-SR04 dengan Arduino. Online di <https://www.mahirelektro.com/2020/04/Tutorial-Menggunakan-Sensor-Ultrasonik-HC-SR04-Arduino.html>, diakses pada 20 Oktober 2022

N. Narendran and Y. Gu, "Life of LED-Based White Light Sources," *Journal Of Display Technology*, vol. 1, no. 1, pp. 167-171, 2005.

Pasaribu, R. A. 2019. Perancangan dan Pembuatan Alat Peringatan Jarak Aman pada Kendaraan Bermotor Menggunakan Sensor Ultrasonik (HC-SR04) Berbasis Mikrokontroler Arduino [Skripsi]. Medan: Universitas Pembangunan Panca Budi. 63 hlm.

Pardede, M., Hutajulu, E., & Sundawa, B. V. (2019). Sistem Monitoring Tempat parkir Berbasis Arduino Mega

Dengan modul Komunikasi XBee Pro S2c. *RELE (Rekayasa Elektrikal Dan Energi) : Jurnal Teknik Elektro*, 1(2), 48–56.

<https://doi.org/10.30596/rele.v1i2.3012>

P. M. Astra, “Prototype Pengunci Pintu Otomatis Menggunakan Rfid (Radio Frequency Identification) Berbasis Mikrokontroler Arduino Uno,” *Technologic*, vol. 11, no. 8, pp. 1–5, 2020.

Ramadhan, F, R Nashrullah, and ... 2019. “Prototipe Sistem Monitoring Parkir Mobil Berbasis Mikrokontroler.” *Autocracy: Jurnal Otomasi ...*, 6–12.  
<https://doi.org/10.21009/autocracy.06.1.2>.

Ridwan, Nur Muhammad. 2017. “Rancang Bangun Sistem Monitoring Tempat Parkir Berbasis Arduino ( Smart Parking ).” *JATI (Jurnal Mahasiswa Teknik Informatika)* 1 (1): 533–39.

Ryan Johan Sembiring, S.T., M.M., M.T. 2020. Jenis-jenis Arduino. Online di <https://podomorouniversity.ac.id/jenis-jenis-arduino/>, diakses 16 September 2022



- Setijowarno, D. dan Frazila, R. B. 2001. Pengantar Sistem Transportasi. Semarang: Universitas Katolik Soegijopranata.
- Suci, Ilham, Nur Alfiansyah, Ramadhan Musaffa, Ichsan Mudjiastuti, and Handajani Agus. 2022. "Analisis Kebutuhan Parkir Pada Gedung Parkir Java Supermall Peterongan Semarang" 18 (1): 1-10.
- Sugiyono. 2011. Metode Penelitian Kuantitatif, Kualitatif dan R&D. Bandung: Alfabeta
- Warpani, S. 1990. Merencanakan Sistem Perangkutan. Bandung: Penerbit ITB
- Wicaksono, M.F., Hidayat. (2017). Mudah Belajar Mikrokontroler Arduino. Bandung: Informatika.
- Winaya, Putu Preantjaya. 2017. "Analisis Karakteristik Dan Solusi Parkir Di Badan Jalan ( Studi Kasus: Jalan Sumatera , Denpasar , Bali )," 41.
- Yuniarti, W.D. (2019). Dasar-dasar Pemrograman dengan Python. Deepublish Publisher ISBN: 9786230203503.

Yusro Muhammad, .M.T., 2017. "Modul teori dan praktikum mikrokontroler platform arduino".

Zuhal. Dasar Teknik Tenaga Listrik dan Elektronika Daya. Jakarta : Gramedia Pustaka Utama, 1995.



## LAMPIRAN

### Lampiran 1 Source Code

Code untuk Arduino komunikasi dengan sensor dan LED.

```
#include <NewPing.h>
```

```
// Define the ultrasonic sensor pins
```

```
const int trigPin1 = 2;
```

```
const int echoPin1 = 2;
```

```
const int trigPin2 = 5;
```

```
const int echoPin2 = 5;
```

```
const int trigPin3 = 8;
```

```
const int echoPin3 = 8;
```

```
const int trigPin4 = 11;
```

```
const int echoPin4 = 11;
```

```
// Define the LED pins
```

```
const int ledPin1 = 3;
```

```
const int ledPin2 = 4;
```

```
const int ledPin3 = 6;
```

```
const int ledPin4 = 7;
```

```
const int ledPin5 = 9;
```

```
const int ledPin6 = 10;
```

```
const int ledPin7 = 12;
```

```
const int ledPin8 = 13;
```

```
// Create a NewPing object
NewPing sonar1(trigPin1, trigPin1, 200);
NewPing sonar2(trigPin2, echoPin2, 200);
NewPing sonar3(trigPin3, echoPin3, 200);
NewPing sonar4(trigPin4, echoPin4, 200);
```

```
void setup() {
  // Set the LED pin as output
  pinMode(ledPin1, OUTPUT);
  pinMode(ledPin2, OUTPUT);
  pinMode(ledPin3, OUTPUT);
  pinMode(ledPin4, OUTPUT);
  pinMode(ledPin5, OUTPUT);
  pinMode(ledPin6, OUTPUT);
  pinMode(ledPin7, OUTPUT);
  pinMode(ledPin8, OUTPUT);
  Serial.begin(9600);
}
```

```
void loop() {
  // Get the distance to the object from the first sensor
  int distance1 = sonar1.ping_cm();
```

```
// If the distance is less than 10 cm, turn on first red LED
if (distance1 <= 10) {
    digitalWrite(ledPin1, HIGH);
} else {
    digitalWrite(ledPin1, LOW);
}
```

```
// If the distance is more than 10 cm, turn one first green
LED
if (distance1 > 10) {
    digitalWrite(ledPin2, HIGH);
} else {
    digitalWrite(ledPin2, LOW);
}
```

```
// Get the distance to the object from the second sensor
int distance2 = sonar2.ping_cm();
```

```
// If the distance is less than 10 cm, turn on second red
LED
if (distance2 <= 10) {
    digitalWrite(ledPin3, HIGH);
} else {
    digitalWrite(ledPin3, LOW);
}
```

```
// If the distance is more than 10 cm, turn on second  
green LED
```

```
if (distance2 > 10) {  
    digitalWrite(ledPin4, HIGH);  
} else {  
    digitalWrite(ledPin4, LOW);  
}
```

```
// Get the distance to the object from the third sensor  
int distance3 = sonar3.ping_cm();
```

```
// If the distance is less than 10 cm, turn on third red LED  
if (distance3 <= 10) {  
    digitalWrite(ledPin5, HIGH);  
} else {  
    digitalWrite(ledPin5, LOW);  
}
```

```
// If the distance is more than 10 cm, turn on third green  
LED
```

```
if (distance3 > 10) {  
    digitalWrite(ledPin6, HIGH);  
} else {  
    digitalWrite(ledPin6, LOW);
```

```

}

// Get the distance to the object from the fourth sensor
int distance4 = sonar4.ping_cm();

// If the distance is less than 10 cm, turn on fourth red
LED
if (distance4 <= 10) {
    digitalWrite(ledPin7, HIGH);
} else {
    digitalWrite(ledPin7, LOW);
}

// If the distance is more than 10 cm, turn one fourth
green LED
if (distance4 > 10) {
    digitalWrite(ledPin8, HIGH);
} else {
    digitalWrite(ledPin8, LOW);
}

// Wait for 1 second
delay(1000);
Serial.println("A1: " + String(distance1) + " cm");
Serial.println("A2: " + String(distance2) + " cm");

```



```
Serial.println("A3: " + String(distance3) + " cm");  
Serial.println("A4: " + String(distance4) + " cm");  
  
}
```

Code untuk NodeMCU menerima data dari Arduino.

```
#include <ESP8266WiFi.h>
```

```
const char* ssid = "ZeroTwo"; // Replace with your WiFi  
network SSID
```

```
const char* password = "zerotwo02"; // Replace with your  
WiFi network password
```

```
void setup() {
```

```
  Serial.begin(9600);
```

```
  delay(1000);
```

```
  // Connect to WiFi
```

```
  WiFi.begin(ssid, password);
```

```
  while (WiFi.status() != WL_CONNECTED) {
```

```
    delay(1000);
```

```
    Serial.print(".");
```

```
  }
```

```
  Serial.println("");
```

```

Serial.print("WiFi connected, IP address: ");
Serial.println(WiFi.localIP());

}

void loop() {
  // Check if there's data available on the serial connection
  from the Arduino
  while (Serial.available() > 0) {
    String data = Serial.readStringUntil('\n'); // Read the
    data until a newline character is received
    Serial.println("Received data: " + data);

    // If the received data starts with "Distance1:", parse and
    print it
    if (data.startsWith("Distance1: ")) {
      String distance1_str = data.substring(11); // Extract the
      distance value (assuming the distance is a positive integer)
      int distance1 = distance1_str.toInt();

    }

    // If the received data starts with "Distance2:", parse and
    print it
    else if (data.startsWith("Distance2: ")) {

```

```

    String distance2_str = data.substring(11); // Extract the
distance value (assuming the distance is a positive integer)
    int distance2 = distance2_str.toInt();

}

// If the received data starts with "Distance3:", parse and
print it
else if (data.startsWith("Distance3: ")) {
    String distance3_str = data.substring(11); // Extract the
distance value (assuming the distance is a positive integer)
    int distance3 = distance3_str.toInt();

}

// If the received data starts with "Distance4:", parse and
print it
else if (data.startsWith("Distance4: ")) {
    String distance4_str = data.substring(11); // Extract the
distance value (assuming the distance is a positive integer)
    int distance4 = distance4_str.toInt();

}
}

```

```
// Add other code or functionality for your NodeMCU here

delay(1000); // Add a small delay to avoid excessive
looping
}
```

Code untuk NodeMCU mengirim data ke Firebase.

```
#include <ESP8266WiFi.h>
#include <FirebaseArduino.h> // Include the Firebase
library
```

```
const char* ssid = "#####"; // Replace with your WiFi
network SSID
```

```
const char* password = "#####"; // Replace with your
WiFi network password
```

```
const char* FIREBASE_HOST = "park-cb163-default-
rtdb.asia-southeast1.firebaseio.com"; // Replace
with your Firebase project URL
```

```
const char* FIREBASE_AUTH =
"QPixrHFbinPsrtAJVdl9DHkJgNaIYo9VpQufRdoo"; //
Replace with your Firebase authentication token
```

```
int distance1 = 0;
```

```
int distance2 = 0;
```

```

int distance3 = 0;
int distance4 = 0;

void setup() {
  Serial.begin(9600);
  delay(1000);

  // Initialize Firebase
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);

  // Connect to WiFi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }
  Serial.println("");
  Serial.print("WiFi connected, IP address: ");
  Serial.println(WiFi.localIP());
}

void loop() {
  // Check if there's data available on the serial connection
  from the Arduino
  while (Serial.available() > 0) {

```

```
String data = Serial.readStringUntil('\n'); // Read the
data until a newline character is received
Serial.println("Received data: " + data);
```

```
// If the received data starts with "A1:", parse and store
it
```

```
if (data.startsWith("A1: ")) {
    String distance1_str = data.substring(4); // Extract the
distance value (assuming the distance is a positive integer)
    distance1 = distance1_str.toInt();
}
```

```
// If the received data starts with "A2:", parse and store
it
```

```
else if (data.startsWith("A2: ")) {
    String distance2_str = data.substring(4); // Extract the
distance value (assuming the distance is a positive integer)
    distance2 = distance2_str.toInt();
}
```

```
// If the received data starts with "A3:", parse and store
it
```

```
else if (data.startsWith("A3: ")) {
    String distance3_str = data.substring(4); // Extract the
distance value (assuming the distance is a positive integer)
```

```
        distance3 = distance3_str.toInt();
    }

    // If the received data starts with "A4:", parse and store
it
    else if (data.startsWith("A4: ")) {
        String distance4_str = data.substring(4); // Extract the
distance value (assuming the distance is a positive integer)
        distance4 = distance4_str.toInt();
    }
}

// Update Firebase Realtime Database with the distance
values
Firebase.setInt("/Distances/A1", distance1);
Firebase.setInt("/Distances/A2", distance2);
Firebase.setInt("/Distances/A3", distance3);
Firebase.setInt("/Distances/A4", distance4);

// Add other code or functionality for your NodeMCU here
```

```
    delay(1000); // Add a small delay to avoid excessive
looping
}
```

Code untuk pembuatan tampilan antarmuka *web*.

```
// Import the functions you need from the SDKs you need
import firebase_app from "@src/firebase/config";
import {   getDatabase,   ref,   onValue   }   from
"firebase/database";
import { useEffect, useState } from "react";
// Get a reference to the database
const database = getDatabase(firebase_app);
function Index() {
  const [distances, setDistances] = useState({});
  const [loading, setLoading] = useState(true);
  useEffect(() => {
    // Create a reference to the "Distances" map
    const distancesRef = ref(database, "Distances");
    // Set up a listener for changes to the "Distances" map
    const unsubscribe = onValue(distancesRef, (snapshot)
=> {
      if (snapshot.exists()) {
        const distancesData = snapshot.val();
        setDistances(distancesData);
      } else {
```



```

    setDistances({});
  }
});
setLoading(false);
// Clean up the listener when the component unmounts
return () => {
  unsubscribe(); // Unsubscribe to prevent memory
leaks
};
}, []); // The empty dependency array means this effect
runs once on component mount
const getParkingColor = (distanceValue) => {
  // Check if the distance is less than 10
  if (distanceValue < 10) {
    return "bg-red-500"; // Green background for available
parking
  } else {
    return "bg-green-500"; // Red background for
unavailable parking
  }
}; // Count the number of parkings with status tru
const getAvailableParkings = () => {
  const availableParkings =
Object.entries(distances).filter(
  ([_, distanceValue]) => distanceValue > 10

```

```

    );
    return availableParkings.map(([distanceName]) =>
distanceName);
  };
  return (
    <>
    <div className="flex justify-center items-center min-
h-screen flex-col">
      {loading ? ( // Render skeleton loading if data is not
ready
        <div className="animate-pulse">
          <div className="bg-gray-300 h-6 w-40 mb-4" />
          <table>
            <tbody>
              <tr>
                <td className="border-4 px-4 py-6 bg-gray-
300">Loading...</td>
              </tr>
            </tbody>
          </table>
        </div>
      ) : (
        <>
          <table>
            <tbody>

```

```

    <tr>
      <td className="border-4 px-4 py-6">
        Parkir
        Tersedia:
        {getAvailableParkings().length}
      </td>
    </tr>
  </tbody>
</table>
<div className="my-4" />
<table>
  <tbody>
    <tr>
      {Object.entries(distances).map(
        ([distanceName, distanceValue]) => (
          <td
            key={distanceName}
            className={`border-4      px-8      py-24
        ${getParkingColor(
          distanceValue
        )}}`
          >
            {distanceName}
          </td>
        )
      )}
    </tbody>
  </table>

```

```

        </tr>
    </tbody>
</table>
</>
    }}
</div>
);
</>
);
}

```

export default Index;

Code untuk fungsi web membaca data dari Firebase.

```
// Import the functions you need from the SDKs you need
```

```
import { initializeApp, getApps } from "firebase/app";
```

```
const firebaseConfig = {
```

```
  apiKey: process.env.NEXT_PUBLIC_FIREBASE_API_KEY,
```

```
  authDomain:
```

```
process.env.NEXT_PUBLIC_FIREBASE_AUTH_DOMAIN,
```

```
  projectId:
```

```
process.env.NEXT_PUBLIC_FIREBASE_PROJECT_ID,
```

```
  storageBucket:
```

```
process.env.NEXT_PUBLIC_FIREBASE_STORAGE_BUCKET,
```

```
    databaseURL:
process.env.NEXT_PUBLIC_FIREBASE_DATABASE_URL,
    messagingSenderId:
process.env.NEXT_PUBLIC_FIREBASE_MESSAGING_SEND
ER_ID,
    appId: process.env.NEXT_PUBLIC_FIREBASE_APP_ID,
};

// Initialize Firebase
let firebase_app =
  getApps().length === 0 ? initializeApp(firebaseConfig) :
  getApps()[0];

export default firebase_app;
```