

# ParkinTime: Sistem Manajemen Parkir Berbasis IoT dengan Website Admin dan Penyempurnaan Fitur

Geovanno Ardy Wicaksana <sup>1\*</sup>, Noris Parompon <sup>2\*</sup>, Nofira Rahmadani <sup>3\*</sup>, Otri Sebastian <sup>4\*</sup>, Banu Filasuf, S.Tr. <sup>5\*\*</sup>

\* Teknologi Rekayasa Perangkat Lunak, Politeknik Negeri Batam

\*\* Teknologi Rekayasa Multimedia, Politeknik Negeri Batam

[geovannoardy@gmail.com](mailto:geovannoardy@gmail.com)<sup>1</sup>, [norisparompon004@gmail.com](mailto:norisparompon004@gmail.com)<sup>2</sup>, [nofirrahmadani0@gmail.com](mailto:nofirrahmadani0@gmail.com)<sup>3</sup>, [otri2008@gmail.com](mailto:otri2008@gmail.com)<sup>4</sup>,  
[banu@polibatam.ac.id](mailto:banu@polibatam.ac.id)<sup>5</sup>

---

## Article Info

### Article history:

Received ...

Revised ...

Accepted ...

---

### Keyword:

*IoT, Manajemen Parkir, Sensor Ultrasonik, Deteksi Objek, MySQL*

---

## ABSTRACT

The demand for a more effective, contemporary, and integrated parking management system has arisen due to the sharp increase in private vehicle ownership, particularly in urban areas. The inability of traditional parking systems to give real-time information on available parking spaces frequently leads to time loss, worsened traffic, and needless fuel usage. In order to solve this problem, this study presents ParkinTime, an Internet of Things (IoT)-based smart parking system that makes use of real-time MySQL databases, ultrasonic sensors, and CCTV-based object detection technology. A Laravel-based website monitoring tool is also included in the system to offer thorough oversight. The design process of ParkinTime uses a hybrid methodology and adheres to the CDIO (Conceive, Design, Implement, Operate) approach. According to an 86-respondent study, 83.72% of participants said they would be interested in utilizing ParkinTime in their everyday activities, and 94.19% thought it was a good way to deal with parking issues. These results demonstrate the community's high degree of acceptability and preparedness for IoT-based parking technology, which also provides a crucial basis for the system's future development and deployment.



This is an open access article under the [CC-BY-SA](#) license.

---

## I. PENDAHULUAN

### A. Latar Belakang

Jumlah kendaraan pribadi yang meningkat di kota-kota besar telah meningkatkan kebutuhan akan sistem parkir yang lengkap. Banyak area parkir masih menggunakan sistem konvensional, yang tidak dapat memberikan informasi tentang ketersediaan slot secara real-time, yang menyebabkan waktu yang terbuang, kemacetan di sekitar, dan pengeluaran bahan bakar yang tidak perlu. Sistem manajemen parkir cerdas mulai menggunakan teknologi Internet of Things (IoT) untuk mengatasi masalah ini.

Sistem parkir menggunakan sensor ultrasonik, kamera CCTV dengan deteksi objek, dan database realtime untuk memberikan informasi yang akurat kepada pengguna dan administrator. Teknologi ini meningkatkan efisiensi parkir dan memungkinkan pengelolaan data yang lebih baik dalam skala besar.

### B. Rumusan Masalah

1. Bagaimana merancang sistem parkir yang mampu memberikan informasi ketersediaan slot secara real-time?

2. Bagaimana integrasi sensor ultrasonik dan kamera CCTV dapat meningkatkan akurasi pendekripsi kendaraan?

3. Bagaimana sistem dapat dikelola dan dipantau melalui website admin secara efektif?

### C. Tujuan

Penelitian ini bertujuan untuk:

1. Mengembangkan sistem parkir berbasis IoT yang dapat mendekripsi slot kosong secara real-time.
2. Meningkatkan efisiensi dan akurasi sistem melalui integrasi sensor dan teknologi pengolahan citra.
3. Menyediakan platform website admin untuk monitoring dan manajemen sistem secara menyeluruh.

### D. Tinjauan Pustaka

Studi telah menunjukkan bahwa sistem parkir cerdas menggunakan sensor dan AI. Dalam deteksi parkir real-time, Paidi et al. (2018) dan Barriga et al. (2019) menekankan kinerja sensor ultrasonik dan desain Internet of Things yang hemat energi. Sebaliknya, penelitian oleh Chen et al. (2023), Wang et al. (2022), dan Neha et al. (2022) menunjukkan bahwa penggabungan algoritma deep learning seperti YOLO dan CNN dengan sistem CCTV meningkatkan akurasi deteksi kendaraan. Sensor ultrasonik telah ditunjukkan untuk mendekripsi objek statis dan dinamis (Mocanu et al., 2016;

Kim, 2016), dan sangat membantu meningkatkan keselamatan dan efisiensi area parkir.

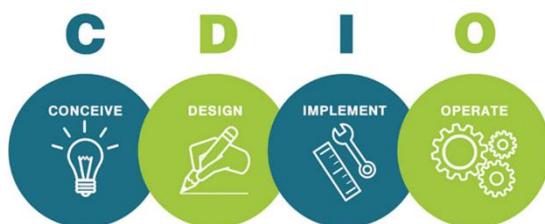
Pengelolaan data yang cepat dan sinkron juga sangat penting untuk sistem ini. Sebagai salah satu pilihan database realtime, MySQL telah menunjukkan kemampuan untuk memantau dan melaporkan data parkir (Marisa, 2015; Nurhanif et al., 2021; Kusdiwadiji & Mathar, 2023). Kombinasi teknologi ini menjadi dasar perancangan ParkinTime, yang merupakan sistem manajemen parkir berbasis Internet of Things yang memiliki website admin dan fitur yang telah disempurnakan.

## II. METODE

### A. Desain dan Pendekatan Penelitian

Studi ini menggunakan logika deduktif dan dimulai dengan teori yang sudah ada tentang seberapa baik sistem manajemen parkir berbasis Internet of Things bekerja. Untuk memastikan bahwa produk dikembangkan secara sistematis dan berpusat pada pengguna, proses pengembangan sistem disusun berdasarkan kerangka CDIO (*Conceive, Design, Implement, and Operate*). Selain itu, penelitian ini menggunakan metode campuran dengan model Desain Parallel Convergent; metode kuantitatif dan kualitatif dikumpulkan secara bersamaan tetapi dianalisis secara terpisah untuk memperoleh pemahaman yang lebih baik tentang pengalaman pengguna dan efektivitas sistem.

### B. Tahapan Pengembangan Berbasis CDIO



Gambar 1 Metode CDIO

1. **Conceive** : Pada tahap ini, tim mengidentifikasi masalah perparkiran melalui studi pasar dan analisis kebutuhan pengguna. Tujuannya adalah untuk membangun sistem manajemen parkir pintar yang dapat membantu pengendara dan pemilik lahan parkir dengan memberikan informasi real-time dan fitur reservasi yang efisien.

2. **Desain**: Prinsip desain berpusat pada pengguna atau user-centered design digunakan saat membangun sistem dan antarmuka. Prototipe awal untuk web admin dan aplikasi pengguna dibuat untuk menguji kinerja dan kemudahan penggunaan. Sensor ultrasonik, kamera CCTV dengan deteksi objek, mikrokontroler ESP32, dan database MySQL adalah komponen sistem yang dirancang untuk diintegrasikan ke dalam sistem.

3. **Implement**: Tahap implementasi dilakukan dengan mengembangkan komponen perangkat lunak: platform web

admin yang dibangun menggunakan Laravel, aplikasi mobile untuk pengguna, kamera CCTV dengan deteksi objek untuk area outdoor (menggunakan OpenCV), dan sensor ultrasonik untuk mendeteksi kendaraan di area parkir indoor. Data slot parkir secara real-time disimpan dan disinkronkan dengan antarmuka pengguna oleh MySQL untuk pengelolaan data backend.

4. **Operate** : Sistem diuji untuk melacak kinerja dan perilaku pengguna dalam lingkungan simulasi. Alat manajemen digunakan untuk memantau status slot parkir. Komentar pengguna awal dikumpulkan untuk memperbaiki versi berikutnya. Pola penggunaan dan masukan pengguna menentukan pemeriksaan data dan pembaruan sistem.

### C. Metode Pengumpulan Data

#### 1. Data Kuantitatif:

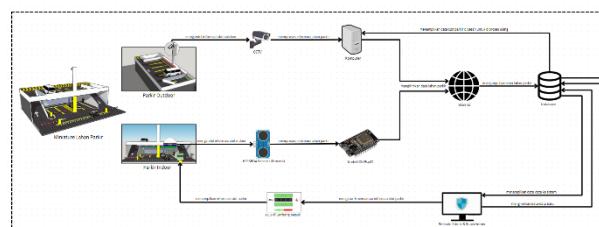
- Kuesioner tertutup menggunakan skala Likert dibagikan kepada pengguna untuk mengukur tingkat kepuasan, kemudahan penggunaan, serta kecepatan pencarian tempat parkir dengan sistem.
- Analisis kinerja sistem dilakukan dengan mengukur akurasi deteksi sensor (ultrasonik dan object detection) serta waktu rata-rata pencarian tempat parkir sebelum dan sesudah menggunakan sistem ParkinTime.

#### 2. Data Kualitatif:

- Wawancara mendalam dilakukan kepada pengguna awal dan pemilik lahan parkir untuk menggali opini terkait desain antarmuka, keandalan sistem, serta dampaknya terhadap operasional.
- Observasi langsung dilakukan untuk melihat perilaku dan interaksi pengguna saat menggunakan sistem pada pengujian lapangan.

## III. HASIL DAN PEMBAHASAN

### A. Gambaran Umum Sistem



Gambar 2 Gambaran Umum Sistem

Dirancang dengan menggunakan Internet of Things (IoT), sistem parkir cerdas (Smart Parking) kecil ini menggunakan platform pemantauan untuk memantau ketersediaan tempat

parkir secara real-time di dua lokasi parkir, indoor dan outdoor.

Dua alur utama membentuk alur input data. Sensor ultrasonik HC-SR04 yang terpasang di setiap slot mendeteksi kendaraan di alur parkir dalam ruangan. Selanjutnya, mikrokontroler NodeMCU Esp32 memproses data jarak dari sensor secara langsung. Mikrokontroler ini memiliki dua output dan juga memproses data. Sementara alur parkir luar menggunakan kamera CCTV untuk mengumpulkan data visual, informasi tentang slot parkir ditampilkan di LCD di Gerbang Masuk secara lokal dan dikirim ke backend melalui Internet. Selanjutnya, gambar atau rekaman CCTV dikirim ke komputer yang bertugas mengumpulkan data tentang lahan parkir. Selanjutnya, komputer ini dapat mengidentifikasi keberadaan kendaraan dengan menggunakan pengolahan gambar. Kemudian, seperti alur di dalam rumah, data parkir dikirim ke backend melalui Internet.

Database akan berfungsi sebagai repositori utama sistem dan mengumpulkan semua data yang dikirim dari NodeMCU dan kedua alur komputer. Selanjutnya, admin dan superadmin website menampilkan data yang diambil dari database dalam dashboard pengawasan dan kontrol. Administrator dapat menampilkan data ke sistem dan mengirimkan data baru ke database dengan menggunakan situs web ini, seperti pembaruan manual atau reservasi. Selain itu, administrator memiliki wewenang untuk mengatur dan memantau informasi slot parkir yang ditampilkan pada LCD Gerbang Masuk. Ini memberikan lapisan kontrol terpusat di atas tampilan data real-time NodeMCU.

## B. Kebutuhan Fungsional dan Non-Fungsional

Ada dua kategori spesifik sistem ParkinTime: fungsional dan non-fungsional

### 1. Kebutuhan Fungsional

Kebutuhan fungsional merupakan hal-hal yang mendasari perancangan aplikasi ini, hal ini dapat dirujuk pada Tabel 1

Tabel 1. Kebutuhan Fungsional

No	Kebutuhan Fungsional
F01	Pengguna dapat login ke sistem.
F02	Sistem mengambil dan menampilkan informasi ketersediaan slot parkir secara otomatis.
F03	Sistem menampilkan data statistik lahan parkir dan data statistik sistem.
F04	Admin dapat menambahkan, mengubah, dan menghapus data pengguna.
F05	Admin dapat menambahkan, mengubah, dan menghapus data lahan parkir.
F06	Admin dapat mengelola data transaksi parkir.

F07	Pengelola lahan parkir dapat mengubah, dan menghapus data lahan parkir mereka.
-----	--

### 2. Kebutuhan Non-Fungsional

Kebutuhan ini dibuat untuk menentukan bagaimana sistem harus berperilaku. hal ini dapat dirujuk pada Tabel 2.

Tabel 2. Kebutuhan Non-Fungsional

No	Kebutuhan Non-Fungsional
NF01	Sistem mampu berjalan selama 24 jam non-stop, kecuali apabila ada perawatan sistem atau pembaharuan sistem.
NF02	Ketahanan terhadap kegagalan minimal 95% uptime.
NF03	Sistem harus mudah digunakan oleh pengguna non-teknis atau user friendly.
NF04	Dapat diakses dari berbagai perangkat dan sistem operasi berbasis web.
NF05	Minimum 8GB RAM.
NF06	Tidak lebih dari 3 detik untuk setiap proses.
NF07	Browser harus menerima sertifikat SSL dari sistem untuk keamanan data.
NF08	Hanya pengguna yang terdaftar dapat mengakses sistem.
NF09	Menggunakan bahasa Indonesia dan bahasa Inggris.

Kebutuhan fungsional, juga dikenal sebagai "kebutuhan fungsional," berbicara tentang kemampuan inti yang harus dimiliki sistem. Ini adalah kebutuhan berikut: 1) Kemampuan sistem untuk mengumpulkan dan menampilkan informasi ketersediaan slot parkir secara real-time; 2) Fungsi autentikasi untuk memvalidasi pengguna (Admin/Super Admin); 3) Kemampuan Administrator untuk mengelola data master, yang mencakup data pengguna, lahan parkir, dan transaksi; dan 4) Kemampuan sistem untuk menampilkan laporan data statistik tentang operasional parkir. Kebutuhan non-fungsional dibuat untuk menjamin kinerja, kualitas, dan keandalan sistem. Salah satu fitur utama yang disebutkan adalah aksesibilitas, yang membutuhkan sistem untuk beroperasi setiap saat, tanpa henti. Respon waktu, yang memiliki batas toleransi 3 detik untuk setiap proses; keamanan, yang membatasi akses sistem hanya untuk pengguna yang telah melakukan autentikasi; dan portabilitas, yang memastikan sistem dapat diakses melalui berbagai web browser pada berbagai perangkat.

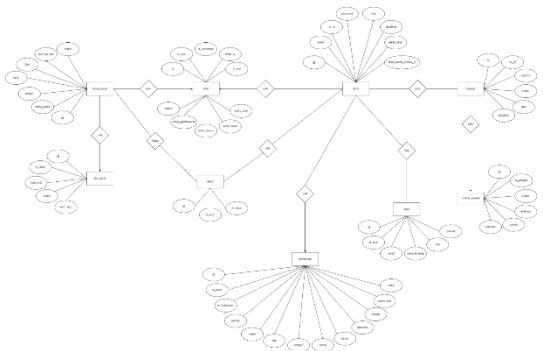
## C. Use Case Diagram



Gambar 3 Use Case Diagram

Diagram Use Case sistem ParkinTime menunjukkan interaksi fungsional antara aktor dan sistem, serta ruang lingkup fungsionalitas situs web admin. Admin dan super admin adalah dua aktor utama dalam arsitektur sistem, dan mereka berbagi kemampuan penting, seperti mengizinkan log in dan memiliki akses ke dashboard sistem dan lahan parkir. Aktor super admin memiliki wewenang utama, mencakup seluruh akses admin dan kontrol penuh atas manajemen data inti, yang mencakup data pengguna, loket parkir, slot parkir, dan tiket parkir.

#### D. ER Diagram



Gambar 4 ER Diagram

Berikut adalah penjelasan dari ER Diagram dari ParkinTime:

##### 1. akun

Entitas ini menyimpan data utama pengguna.

- a. id (Primary Key): Identifier unik untuk setiap akun.
- b. email: Alamat email unik untuk login dan komunikasi.

- c. no\_telp: Nomor telepon pengguna.
- d. password: Kata sandi terenkripsi untuk otentifikasi.
- e. role: Peran pengguna dalam sistem (misal:super admin, admin, user).
- f. datetime: Tanggal akun dibuat
- g. reset\_token: Token untuk proses reset password.
- h. reset\_token\_expires\_at: Waktu kedaluwarsa token reset.

##### 2. profil

Entitas ini menyimpan detail data diri pengguna.

- a. id (Primary Key): Identifier unik untuk profil.
- b. id\_akun (Foreign Key): Menghubungkan ke id pada entitas akun.
- c. email: Email yang terverifikasi (mungkin berbeda atau sama dengan akun).
- d. nama\_lengkap: Nama lengkap resmi pengguna.
- e. foto: Foto profil pengguna.
- f. alamat: Alamat pengguna.

##### 3. Kendaraan

Entitas ini menyimpan data kendaraan yang didaftarkan oleh akun.

- a. id (Primary Key): Identifier unik untuk setiap kendaraan.
- b. id\_akun (Foreign Key): Menghubungkan ke id pada entitas akun.
- c. no\_kendaraan: Nomor polisi (plat) kendaraan.
- d. pemilik: Nama pemilik STNK.
- e. merek: Merek kendaraan (misal: Toyota, Honda).
- f. tipe: Tipe/model kendaraan (misal: Avanza, Vario).
- g. kategori: Kategori kendaraan (misal: Mobil, Motor).

- h. warna: Warna kendaraan.

- i. tahun: Tahun kendaraan.

- j. kapasitas: Kapasitas Kendaraan.

- k. energi: Jenis bahan bakar (misal: Bensin, Listrik).

- l. warna\_plat: Warna plat nomor kendaraan.

- m. status: Status kendaraan (misal: terverifikasi, dalam proses).

##### 4. dompet

Entitas ini berfungsi sebagai dompet digital (e-wallet) internal pengguna

- a. id (Primary Key): Identifier unik untuk dompet.
- b. no\_hp: Nomor HP yang tertaut dengan dompet.
- c. balance: Sisa saldo dalam dompet.
- d. status: Status dompet (misal: aktif, diblokir).
- e. tipe: Tipe dompet digital.
- f. datetime: Waktu aktivitas dompet.

##### 5. riwayat\_update

Entitas ini mencatat semua mutasi (masuk/keluar) pada dompet.

- a. id (Primary Key): Identifier unik untuk setiap transaksi.
  - b. id\_dompet (Foreign Key): Menghubungkan ke id pada entitas dompet.
  - c. status: Status update dompet.
  - d. id\_riwayat: Kode unik untuk riwayat/referensi.
  - e. deskripsi: Keterangan transaksi.
  - f. jumlah: Jumlah uang (bisa positif untuk top-up, negatif untuk bayar).
  - g. datetime: Waktu transaksi digital.
6. lahan\_parkir
- Entitas ini mendefinisikan properti dari sebuah lokasi parkir.
- a. id (Primary Key): Identifier unik untuk lokasi parkir.
  - b. nama\_lokasi: Nama lokasi parkir (misal: "Mall A - Lantai 1").
  - c. alamat: Alamat fisik lokasi parkir.
  - d. jenis: Jenis lahan parkir.
  - e. foto: Foto lahan parkir.
  - f. tarif\_per\_jam: Biaya parkir per jam di lokasi ini.
  - g. status: Status operasional (misal: Buka, Tutup, Penuh).
7. slot\_parkir
- Entitas ini merepresentasikan setiap slot parkir individual (Inti IoT).
- a. id (Primary Key): Identifier unik untuk setiap slot.
  - b. id\_lahan (Foreign Key): Menghubungkan ke id pada entitas lahan\_parkir.
  - c. kode\_slot: Nama/kode slot.
  - d. status: Status ketersediaan slot.
  - e. jenis\_slot: Jenis slot.
8. tiket
- Entitas ini adalah catatan transaksional untuk setiap sesi parkir.
- a. id (Primary Key): Identifier unik untuk tiket.
  - b. id\_akun (Foreign Key): Menghubungkan akun dan tiket.
  - c. id\_kendaraan: Menghubungkan entitas kendaraan dan tiket.
  - d. order\_id: ID pesanan atau referensi transaksi.
  - e. id\_slot: Menghubungkan entitas slot parkir dan tiket.
  - f. status: Status tiket (misal: 'aktif', 'selesai').
  - g. status\_pembayaran: Status pembayaran (misal: 'lunas', 'belum\_lunas').
  - h. waktu\_masuk: Waktu dan tanggal saat kendaraan masuk.
  - i. waktu\_keluar: Waktu dan tanggal saat kendaraan keluar.
  - j. biaya\_total: Biaya akhir setelah parkir selesai.
9. tenant

Entitas ini digunakan untuk mencatat data terkait tenant.

- a. id (Primary Key): Identifier unik untuk laporan.
- b. id\_akun: Menyimpan foreign key dari entitas akun.
- c. id\_lahan: Menyimpan foreign key dari entitas lahan\_parkir.

## E. Hasil Implementasi

```

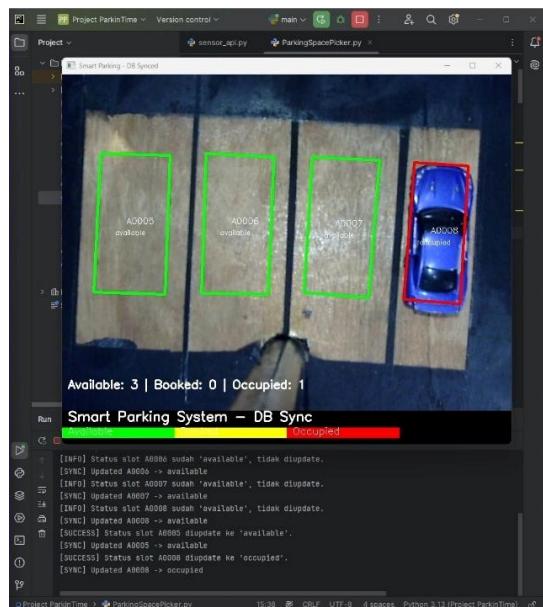
parktime_project.ino
File Edit Sketch Tools Help
ESP32 Dev Module
parktime_project.ino README.md cipron
91 digitalWrite(trigPin, LOW);
92 delayMicroseconds(2);
93 digitalWrite(trigPin, HIGH);
94 delayMicroseconds(10);
95 digitalWrite(trigPin, LOW);
96
97 long duration = pulseIn( echoPin, HIGH, 30000); // Timeout 30ms
98 if (duration == 0) return 1;
99 return duration * 0.034 / 2;
100 }

101 /**
102 * KIRIM DATA KE SERVER
103 */
104 void kirimStatus(String kode_slot, String status) {
105   if (WiFi.status() != WL_CONNECTED) {
106     Serial.println("WiFi tidak terkoneksi");
107     return false;
108   }
109   HTTPClient http;
110   http.begin(serverURL);
111
Output Serial Monitor X
Message (Enter to send message to ESP32 Dev Module on 'COM3')
New Line 115200 baud
Slot I0001: 10.23 cm - available
Slot I0002: 10.18 cm - available
Slot I0003: 9.64 cm - available
Slot I0004: 10.40 cm - available
[UPDATE SLOT]
Slot I0001: 10.23 cm - available
Slot I0002: 10.10 cm - available
Slot I0003: 9.64 cm - available
Slot I0004: 10.39 cm - available
[UPDATE SLOT]
Slot I0001: 10.23 cm - available
Slot I0002: 10.10 cm - available
Slot I0003: 9.64 cm - available
Slot I0004: 10.39 cm - available
Ln 101, Col 1 ESP32 Dev Module on COM3

```

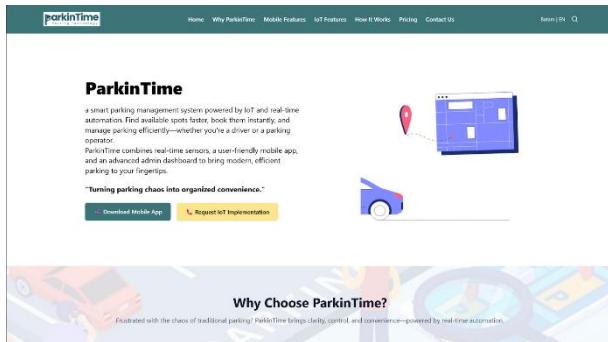
Gambar 5 Kode program IoT

Subsistem deteksi IoT dan platform administrasi berbasis web adalah dua skenario yang berbeda. Tujuan dari tahap implementasi adalah untuk menerapkan arsitektur sistem yang telah dirancang. Sensor ultrasonik HC-SR04 dan mikrokontroler ESP32 digunakan untuk parkir dalam ruangan. Seperti yang ditunjukkan pada gambar, alat pengukur jarak dapat dikodekan melalui IDE Arduino.



Gambar 6 Implementasi Object Detection

Ketika sensor menemukan jarak di bawah 9 cm, slot dikategorikan sebagai "Terisi". Untuk parkir di luar, Python dan library OpenCV digunakan untuk sistem deteksi objek berbasis komputer vision. Ini menggunakan umpan video kamera untuk mendeteksi keberadaan kendaraan secara real-time. Ini seperti yang ditunjukkan pada gambar.



Gambar 7 Halaman Home

Halaman web ini memperkenalkan ParkinTime, sebuah sistem manajemen parkir pintar berbasis Internet of Things yang bertujuan untuk mengatasi masalah parkir konvensional di kota-kota. Dengan menggabungkan sensor real-time, aplikasi seluler untuk pengemudi, dan dasbor admin untuk operator, solusi ini bertujuan untuk mengubah kekacauan parkir menjadi tempat yang lebih nyaman.



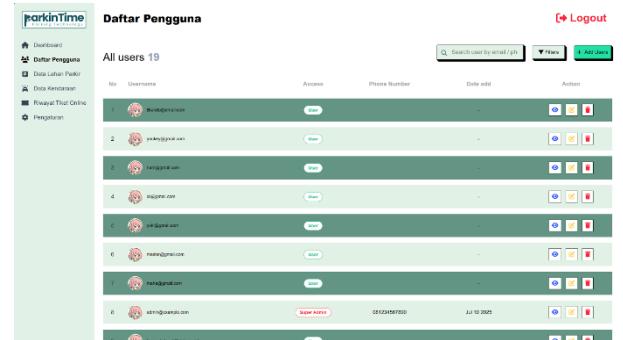
Gambar 8 Halaman Login

Halaman login ini berfungsi sebagai pintu masuk aman ke sistem ParkinTime untuk dua jenis pengguna yang berbeda Admin dan Tenant. Bergantung pada email dan kata sandi yang mereka masukkan, pengguna akan diarahkan ke dasbor yang sesuai untuk Admin, yang bertanggung jawab atas manajemen sistem secara keseluruhan, dan Tenant, yang bertanggung jawab atas operasional atau data khusus terkait area parkir mereka.



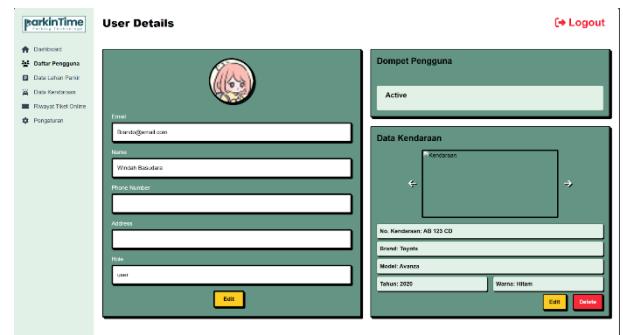
Gambar 9 Halaman Dashboard Admin

Dashboard ini berfungsi sebagai pusat pemantauan dan menyajikan ringkasan statistik penting, seperti jumlah pengguna, jumlah lahan parkir, dan statistik tiket, secara visual dengan menggunakan chart.js.



Gambar 10 Halaman Daftar Pengguna

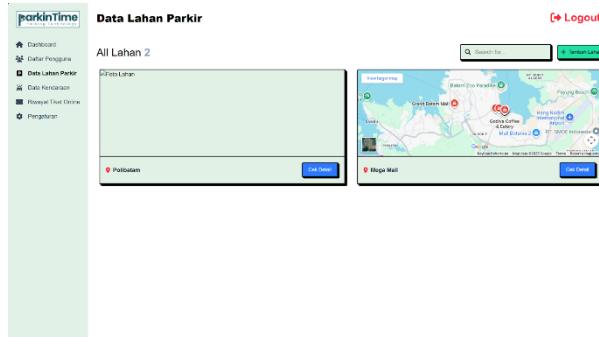
Halaman "Daftar Pengguna" adalah antarmuka admin utama yang digunakan untuk mengelola data pengguna. Halaman ini menampilkan daftar lengkap dari semua akun yang terdaftar dalam sistem, termasuk pembedaan hak akses untuk akun "User" dan "Super Admin". Halaman ini juga memungkinkan administrator untuk mencari, memfilter, menambah, atau menghapus data pengguna secara langsung ("Action").



Gambar 11 Halaman Detail Pengguna

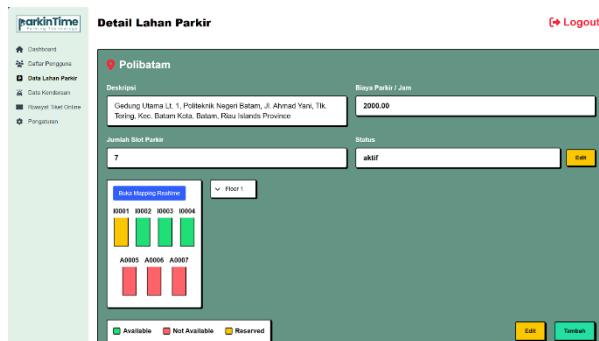
Halaman "Detail Pengguna" ini menampilkan profil utama dan tampilan admin lengkap untuk akun pengguna tertentu, di mana rincian data pengguna, informasi kontak,

dan hak akses ("Role) disertakan. Halaman ini juga menggabungkan data terkait pengguna, seperti status "Dompet Pengguna" dan daftar "Data Kendaraan" yang terdaftar, dengan opsi untuk admin untuk mengubah atau menghapus data tersebut.



Gambar 12 Halaman Data Lahan Parkir

Semua lokasi parkir yang terintegrasi dalam sistem ParkinTime disimpan di halaman "Data Lahan Parkir", yang menampilkan gambar dan peta (seperti "Polibatam" dan "Mega Mall"). Halaman ini memungkinkan manajemen seluruh lahan yang terdaftar; Anda dapat menambahkan tempat parkir baru dengan menekan tombol "Tambah Lahan", dan Anda dapat menggunakan "Cek Detail" untuk mengkonfigurasi atau memantau setiap lahan secara terpisah.



Gambar 13 Halaman Detail Lahan Parkir

Halaman "Detail Lahan Parkir" ini menampilkan GUI manajemen komprehensif untuk lokasi parkir tertentu, dalam hal ini "Polibatam". Ini memungkinkan petugas mengatur data operasional seperti deskripsi, biaya per jam, dan jumlah slot yang tersedia. Selain itu, halaman ini menyediakan fitur utama sistem Internet of Things, seperti memantau ketersediaan setiap slot parkir secara real-time dengan pemetaan visual yang dinamis yang menunjukkan apakah slot parkir tersebut tersedia, tidak tersedia, atau terreservasi.

Data Kendaraan						
No.	No. Kendaraan	Jenis	Merk	Tipe	Aksi	
1	SP123456	SPD.001 & S. BERPENGARUH	HONDA	APRIZUZI08 M/T	<input checked="" type="checkbox"/> <input type="checkbox"/>	<input type="button" value="Aksi"/>
2	SP123457	STTHON & ULIRGENNA	DANHISU	TOSHIBA P70000 TX-AF	<input checked="" type="checkbox"/> <input type="checkbox"/>	<input type="button" value="Aksi"/>
3	SP123458	STTHON & ULIRGENNA	DANHISU	TOSHIBA P70000 TX-AF	<input checked="" type="checkbox"/> <input type="checkbox"/>	<input type="button" value="Aksi"/>
4	SP123459	SPD.001 & S. BERPENGARUH	HONDA	E2V AT	<input checked="" type="checkbox"/> <input type="checkbox"/>	<input type="button" value="Aksi"/>
5	SP123460	SPD.001 & S. BERPENGARUH	HONDA	APRIZUZI08 M/T	<input checked="" type="checkbox"/> <input type="checkbox"/>	<input type="button" value="Aksi"/>
6	SP123461	STTHON & ULIRGENNA	DANHISU	TOSHIBA P70000 TX-AF	<input checked="" type="checkbox"/> <input type="checkbox"/>	<input type="button" value="Aksi"/>
7	SP123462	STTHON & ULIRGENNA	DANHISU	TOSHIBA P70000 TX-AF	<input checked="" type="checkbox"/> <input type="checkbox"/>	<input type="button" value="Aksi"/>

Gambar 14 Halaman Data Kendaraan

Halaman "Data Kendaraan" berfungsi sebagai database utama yang menyimpan semua data kendaraan yang telah didaftarkan oleh pengguna dalam sistem ParkinTime. Halaman ini menampilkan informasi penting seperti nomor polisi, jenis, merek, dan tipe kendaraan, dan memungkinkan administrator untuk mengatur dan melihat data kendaraan dengan cara yang berbeda. Selain itu, ada kemampuan untuk mencari, memfilter, mengedit, atau menghapus data spesifik kendaraan ("Aksi").

Riwayat Tiket Online						
Date		Tickets Booked	Ticket Price	Location		
31-10-2025		15	Rp373	One Batam Mall		
No.	Ma. No. Pelajar	ID User	Waktu Parkir	Lokasi	Saat	Aksi
1	PARKINTIME-170308E07	norwulanpradi@gmail.com	14:10	Polibatam	AM07	<input checked="" type="checkbox"/> <input type="checkbox"/>
2	PARKINTIME-170308T02	carina@maxxata.com	20:00	Polibatam	AM07	<input checked="" type="checkbox"/> <input type="checkbox"/>
3	PARKINTIME-170308T010	carina@maxxata.com	12:00	Polibatam	AM07	<input checked="" type="checkbox"/> <input type="checkbox"/>
4	PARKINTIME-170308E00	carina@maxxata.com	12:00	Polibatam	AM07	<input checked="" type="checkbox"/> <input type="checkbox"/>
5	PARKINTIME-170308T022	carina@maxxata.com	15:00	Polibatam	AM08	<input checked="" type="checkbox"/> <input type="checkbox"/>
6	PARKINTIME-170308E045	carina@maxxata.com	13:00	Polibatam	AM07	<input checked="" type="checkbox"/> <input type="checkbox"/>
7	PARKINTIME-170308T010	carina@maxxata.com	14:00	Polibatam	AM08	<input checked="" type="checkbox"/> <input type="checkbox"/>
8	PARKINTIME-170308T005	carina@maxxata.com	14:00	Polibatam	AM07	<input checked="" type="checkbox"/> <input type="checkbox"/>
9	PARKINTIME-170308T006	carina@maxxata.com	14:00	Polibatam	AM08	<input checked="" type="checkbox"/> <input type="checkbox"/>
10	PARKINTIME-170308T00	carina@maxxata.com	14:00	Polibatam	AM07	<input checked="" type="checkbox"/> <input type="checkbox"/>

Gambar 15 Halaman Riwayat Tiket Online

Halaman "Riwayat Tiket Online" berfungsi sebagai dasbor untuk melaporkan dan mengaudit semua transaksi parkir yang terjadi dalam sistem. Anda dapat memfilternya berdasarkan tanggal dan lokasi tertentu, seperti "Mall One Batam". Halaman ini tidak hanya menampilkan ringkasan metrik penting, seperti jumlah tiket yang dipesan dan total pendapatan, tetapi juga menyediakan log detail dari setiap transaksi, yang mencakup nomor tiket, ID pengguna, waktu, lokasi, dan lokasi. Administrator memiliki kemampuan untuk melihat detail atau menghapus data ("Aksi").



Gambar 16 Halaman Dashboard Tenant

Untuk Tenant, halaman "Dashboard" ini berfungsi sebagai pusat pemantauan visual khusus. Ini menampilkan metrik seperti "Jumlah Spot Parkir", "Jumlah Spot VIP", dan "Jumlah Tiket." Tenant dapat melihat kinerja harian dan tren penggunaan di tempat parkir mereka dengan melihat data seperti "Statistik Parkir Rata-Rata" (grafik batang) dan "Statistik Parkir Mingguan" (grafik garis).



Gambar 17 Halaman Data Spot Parkir

Untuk Tenant, seperti "Mega Mall", halaman "Data Spot Parkir" ini berfungsi sebagai platform utama untuk mengelola dan memantau slot parkir khusus di lokasi mereka. Halaman ini memiliki pemetaan visual real-time yang menunjukkan status ketersediaan setiap slot (Available, Not Available, Reserved). Selain itu, memungkinkan tenant untuk mengkonfigurasi data slot (Edit, Tambah) dan beralih antar lantai.

Riwayat Tiket Online						
Daftar Tiket yang Dipesan						
No.	No. Tiket #	ID User	No. Kendaraan	Tanggal	Bulan	Aksi
1	0-103	Mindra	BP 0001 AC	17.02	02	<a href="#">[Edit]</a> <a href="#">[Hapus]</a>
2	0-104	Lala	BP 0002 AC	18.02	02	<a href="#">[Edit]</a> <a href="#">[Hapus]</a>
Riwayat Parkir						
No.	No. Tiket #	ID User	In	Out	Spot	Aksi
1	0-103	Mindra	15.02	17.02	16	<a href="#">[Edit]</a> <a href="#">[Hapus]</a>
2	0-104	Lala	16.02	18.02	17	<a href="#">[Edit]</a> <a href="#">[Hapus]</a>

Gambar 18 Halaman Riwayat Tiket Online Tenant

Untuk membantu Tenant mengelola aktivitas parkir di lokasi mereka, halaman "Riwayat Tiket Online" ini terdiri dari dua bagian utama: "Daftar Tiket yang Dipesan" menampilkan pemesanan saat ini atau yang akan datang, dan "Riwayat Parkir" menampilkan log audit untuk semua transaksi parkir yang telah selesai, lengkap dengan waktu masuk dan keluar.

#### F. Hasil Pengujian

Metode black box testing digunakan untuk menguji fungsiionalitas dan kesesuaian alur kerja aplikasi web ParkinTime. Pengujian ini fokus pada verifikasi sistem dari sudut pandang admin dan tenant, tanpa memperhatikan struktur kode internal. Seluruh fitur utama yang telah dirancang dimasukkan dalam skenario pengujian. Ini termasuk proses autentikasi (Login), manajemen data (seperti Daftar Pengguna, Data Lahan Parkir, dan Data Kendaraan), pemantauan operasional dalam waktu nyata (seperti Dashboard, Detail Lahan Parkir), dan fungsi pelaporan. Tujuan dari pengujian ini adalah untuk memastikan bahwa setiap fungsi berjalan sesuai dengan spesifikasi yang diperlukan, dan bahwa setiap input dari antarmuka pengguna (UI) menghasilkan output yang akurat dan diharapkan.

Tabel 3 Kasus Uji 1

Identifikasi	KU-001	
Nama Kasus Uji	Melakukan Pengujian dalam pengambilan data parkir indoor	
Deskripsi	Menguji apakah sensor ultrasonik indoor pada ESP32 dapat membaca keberadaan kendaraan di atas slot dengan akurat	
Kondisi Awal	Slot indoor kosong, sensor aktif, ESP32 terhubung ke power	
Tanggal Pengujian	11 Juli 2025	
Penguji	Noris Parompon	
Skenario	<ol style="list-style-type: none"> <li>Nyalakan ESP32 dengan kode .ino yang terhubung ke sensor jarak.</li> <li>Letakkan kendaraan di atas salah satu slot indoor.</li> <li>Baca output dari Serial Monitor.</li> <li>Bandingkan pembacaan jarak dengan batas 9 cm dan lihat perubahan status slot.</li> </ol>	
Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan
Slot terbaca <b>occupied</b> saat kendaraan diletakkan	Sensor membaca < 9 cm dan status berubah menjadi <b>occupied</b>	Hasil pengamatan sesuai
Slot terbaca <b>available</b> saat tidak ada kendaraan	Sensor membaca > 9 cm dan status menjadi <b>available</b>	

Tabel 4 Kasus Uji 2

Identifikasi	KU-002	
Nama Kasus Uji	Melakukan Pengujian dalam pengambilan data parkir outdoor	
Deskripsi	Menguji apakah sistem dapat mendeteksi keberadaan kendaraan pada slot parkir outdoor menggunakan kamera webcam dan analisis citra	
Kondisi Awal	Kamera aktif, file layout (CarParkPost) & metadata sudah dimuat	
Tanggal Pengujian	11 Juli 2025	
Penguji	Noris Parompon	
Skenario	<ol style="list-style-type: none"> <li>1. Jalankan program main.py untuk sistem pengolahan citra webcam.</li> <li>2. Pastikan kamera mengarah ke area slot parkir outdoor yang sudah ditentukan.</li> <li>3. Letakkan kendaraan (atau benda penghalang) di atas slot.</li> <li>4. Amati perubahan status slot berdasarkan citra visual dan nilai ambang deteksi.</li> </ol>	
Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan
Slot terbaca <b>occupied</b> saat ada kendaraan secara visual	Kamera mendeteksi perubahan area slot dan sistem menampilkan status <b>occupied</b>	Hasil pengamatan sesuai
Slot terbaca <b>available</b> saat slot kosong	Tidak ada gangguan di area slot, status tetap <b>available</b>	

Tabel 5 Kasus Uji 3

Identifikasi	KU-003	
Nama Kasus Uji	Melakukan Pengujian dalam pengiriman data parkir indoor	
Deskripsi	Menguji apakah ESP32 indoor dapat mengirim status parkir secara otomatis ke server Flask API saat status berubah	
Kondisi Awal	ESP32 indoor aktif, WiFi terhubung, server API (Flask) berjalan	
Tanggal Pengujian	11 Juli 2025	
Penguji	Noris Parompon	
Skenario	<ol style="list-style-type: none"> <li>1. Jalankan Flask API (sensor_API.py) di laptop.</li> <li>2. Nyalakan ESP32 dengan kode .ino.</li> <li>3. Letakkan kendaraan di atas slot dan amati perubahan status di serial monitor.</li> <li>4. Cek apakah data berhasil terkirim dan status berubah di database (slot_parkir).</li> </ol>	
Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan
Status slot dikirim ke server saat berubah	ESP32 menampilkan “200 OK” di Serial Monitor	Hasil pengamatan sesuai
Database slot_parkir terupdate otomatis	Data status berhasil diubah sesuai yang dikirim oleh ESP32	

Tabel 6 Kasus Uji 4

Identifikasi	KU-004	
Nama Kasus Uji	Melakukan Pengujian dalam pengiriman data parkir outdoor	
Deskripsi	Menguji apakah sistem dengan kamera webcam outdoor dapat mengirim status slot secara otomatis ke database CPanel melalui fungsi sinkronisasi	
Kondisi Awal	Kamera aktif, koneksi internet tersedia, database hosting siap	
Tanggal Pengujian	11 Juli 2025	
Penguji	Noris Parompon	
Skenario	<ol style="list-style-type: none"> <li>1. Jalankan main.py di laptop yang terhubung ke internet.</li> <li>2. Pastikan server CPanel bisa diakses dan koneksi database berhasil.</li> <li>3. Letakkan kendaraan di slot outdoor dan amati perubahan status visual.</li> </ol>	

4. Buka database CPanel → cek apakah status slot berubah sesuai hasil webcam.		
Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan
Sistem kirim status terbaru ke database	Status occupied / available berubah di tabel slot_parkir	
Data terkirim hanya saat status berubah	Tidak ada spam/pengulangan kiriman jika status tetap sama	Hasil pengamatan sesuai

#### IV. KESIMPULAN

ParkinTime merupakan inovasi berbasis Internet of things (IoT) yang dikembangkan untuk menjawab berbagai tantangan pada sistem parkir konvensional, seperti kesulitan menemukan tempat parkir, keterbatasan dalam pemantauan, serta rendahnya efisiensi waktu. Sistem ini menggabungkan penggunaan sensor ultrasonic, teknologi deteksi objek melalui kamera CCTV, serta platform administrasi berbasis web menggunakan Laravel yang terhubung dengan database MySQL secara real-time. Berdasarkan survei yang dilakukan terhadap 86 responden, lebih dari 94% menilai Parkintime sebagai Solusi yang tepat, dan lebih dari 83% menyatakan kesedianya untuk menggunakannya. Temuan ini mencerminkan Tingkat penerimaan yang tinggi dari Masyarakat terhadap konsep sistem parkir cerdas. Meskipun sistem ini masih berada pada tahap awal dan belum diimplementasikan secara penuh dalam scenario nyata, hasil tersebut memberikan dasar yang kuat untuk melanjutkan ke tahap pengembangan berikutnya, termasuk uji coba prototipe dan evaluasi menyeluruh terhadap integrasi sistem.

Penting untuk digarisbawahi bahwa penelitian ini masih bersifat berkelanjutan dan sistem yang diusulkan sedang dalam tahap pengembangan aktif. Meskipun sistem ini masih berada pada tahap awal dan belum diimplementasikan secara penuh dalam skenario nyata, hasil positif dari survei tersebut memberikan dasar yang kuat untuk melanjutkan ke tahap pengembangan berikutnya, termasuk uji coba prototipe yang lebih komprehensif dan evaluasi menyeluruh terhadap integrasi sistem di lapangan.

#### UCAPAN TERIMA KASIH

Ucapan terima kasih kepada Politeknik Negeri Batam, khususnya Program Studi Teknologi Rekayasa Perangkat Lunak atas penyediaan fasilitas serta bimbingan yang telah diberikan selama berlangsungnya proses penelitian dan pengembangan sistem ParkinTime. Rasa terima kasih juga

ditunjukkan kepada para dosen pembimbing yang telah memberikan panduan serta masukan yang bernilai dan membangun. Selain itu, penghargaan diberikan kepada seluruh responden yang telah bersedia meluangkan waktu untuk berpartisipasi dalam pengisian kuesioner dan proses uji coba system. Seluruh bentuk dukungan dan keterlibatan yang diberikan sangat berkontribusi dalam keberhasilan penyelesaian penelitian ini.

#### DAFTAR PUSTAKA

- [1] V. Paidi, H. Fleyeh, J. Håkansson, and R. G. Nyberg, “Smart parking sensors, technologies and applications for open parking lots: a review,” IET Intelligent Transport Systems, vol. 12, no. 8, pp. 735–741, 2018. <https://doi.org/10.1049/iet-its.2017.0406>
- [2] J. J. Barriga et al., “Smart Parking: A Literature Review from the Technological Perspective,” Applied Sciences, vol. 9, no. 21, p. 4569, 2019. <https://doi.org/10.3390/app9214569>.
- [3] X. Chen, W. Yang, S. Zeng, L. Geng, and Y. Jiao, “YOLOAL: Focusing on the Object Location for Detection on Drone Imagery,” IEEE Access, 2023. <https://doi.org/10.1109/ACCESS.2023.3332815>.
- [4] Y. Neha, V. Saritha, N. Samyuktha, B. Gayathri, and A. Charith, “Smart Parking System Using Object Detection,” Proceedings of the 2nd Indian International Conference on Industrial Engineering and Operations Management, 2022.
- [5] X. Wang et al., “Real-Time and Efficient Multi-Scale Traffic Sign Detection Method for Driverless Cars,” Sensors, vol. 22, no. 18, p. 6930, 2022. <https://doi.org/10.3390/s22186930>.
- [6] B. Mocanu, R. Tapu, and T. Zaharia, “When Ultrasonic Sensors and Computer Vision Join Forces for Efficient Obstacle Detection and Recognition,” Sensors, vol. 16, no. 11, p. 1807, 2016. <https://doi.org/10.3390/s16111807>
- [7] E. Y. Kim, “Wheelchair Navigation System for Disabled and Elderly People,” Sensors, vol. 16, no. 11, p. 1806, 2016. <https://doi.org/10.3390/s16111806>
- [8] F. Marisa, “Model dan Implementasi Teknik Query Realtime Database untuk Mengolah Data Finansial pada Aplikasi Server Pulsa Reload Berbasis .NET,” Conference Paper, Universitas WidyaGama Malang, 2015.
- [9] N. Nurhanif, Y. Away, and M. S. Surbakti, “Analisis kinerja sinkronisasi database pada DBMS MySQL dan Oracle dengan menggunakan event-driven dan time-driven untuk pemantauan data cuaca,” J. Aceh Phys. Soc., vol. 10, no. 4, pp. 107–112, 2021. <https://doi.org/10.24815/jacps.v10i4.20084>