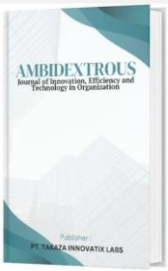
Takaza
INNOVATIX LABS

Ambidextrous: Journal of Innovation, Efficiency and Technology in Organization

https://journal.takaza.id/index.php/ambidextrous

Vol. 3 , No. 1 , 2025, pp. 134-146
E-ISSN: 3031-7002

E-mail:ambidex@takaza.id


AMBIDEXTROUS
Journal of Innovation, Efficiency and Technology in Organization
PT. TAKAZA INNOVATIX LABS

Development of a Web-Based Motorcycle Rental Application with Real-Time Status and Digital Financial Recording

Anggiah Salim[1], Sakina Sudin[2], Nurafni Muhammad[3], Fitriani Duwila[4], Frenti Polulu[5]

[12345]Program Studi Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Maluku Utara
Corresponding Author e-mail: anggiahsalim1@gmail.com

Article History:
Received: 01-01-2025
Revised: 07-01-2025
Accepted: 10-01-2025

Keywords: *information system; motorcycle rental; Supabase; Vercel; MSME*

Abstract: *Motorcycle rental businesses at the Micro, Small, and Medium Enterprises (MSME) level are still largely managed manually, which may lead to transaction recording errors and delays in vehicle status information. This study aims to design and implement a web-based motorcycle rental information system developed using HTML, CSS, JavaScript, and Bootstrap as the user interface. The system is managed through a GitHub repository, utilizes Supabase as a real-time database, and is deployed on the Vercel platform to provide public domain access. The research method follows system development stages consisting of requirement analysis, system design, implementation, and testing. The system is able to display real-time motorcycle availability, support admin-managed rental transactions, and generate daily income reports. To protect the confidentiality of MSME financial data, the income report is limited to a three-day period. The results indicate that the system operates properly and is feasible to be used as an operational support tool for web-based motorcycle rental services.*

How to Cite: Anggiah Salim, Sakina Sudin, Nurafni Muhammad, Fitriani Duwila, Frenti Polulu. (2025). *Development of a Web-Based Motorcycle Rental Application with Real-Time Status and Digital Financial Recording*. 3(1). Pp.133-146 https://doi.org/10.61536/ambidextrous.v3i1.422

https://doi.org/10.61536/ambidextrous.v3i1.422

Introduction

The development of web-based information technology has driven the digitalization of various business sectors, including motorcycle rental services. However, many motorcycle rental businesses still rely on manual or semi-digital record-keeping, which leads to delays in updating vehicle status, transaction recording errors, and difficulties in preparing financial reports. These issues ultimately reduce operational efficiency and service quality.

Web-based information systems represent an effective solution, consisting of interrelated components that interact to collect, process, store, and disseminate information to support decision-making and control within an organization (e.g., a rental business). Such systems can enhance the efficiency and effectiveness of service-oriented businesses, including motorcycle rental services (Alcianno G. Gani, 2025).

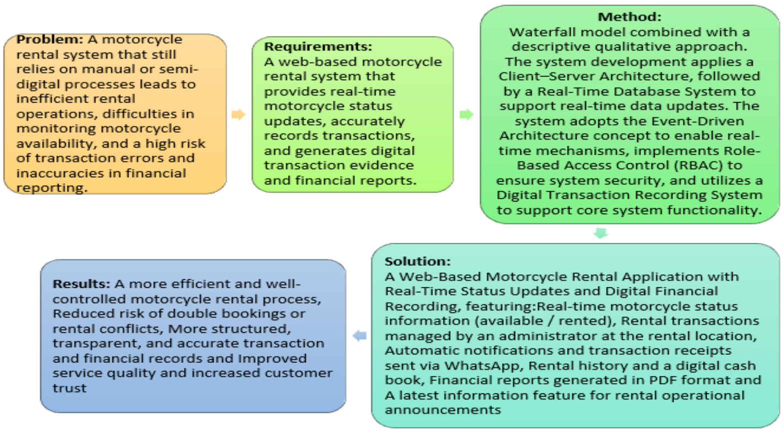
The novelty of this study lies in the design and implementation of a web-based motorcycle rental information system developed to address the core management needs of MSME (Micro, Small, and Medium Enterprise) motorcycle rental businesses. The system is designed to display real-time vehicle status to the general public, enabling fast and accurate access to motorcycle availability information. In addition, the system implements structured digital transaction and financial recording and is capable of generating income reports in PDF format as administrative documentation. From a security perspective, the system applies Role-Based Access Control (RBAC) to differentiate functions and access privileges between administrators and general users. The system is integrated with a Supabase database and WhatsApp notifications and is deployed via the Vercel platform to enable public access. To protect the confidentiality of MSME business financial data, income reports are limited to displaying data from a three-day period. The primary focus of this research is on system functionality and performance rather than long-term financial analysis (Anggiah Salim, 2025).

Based on the above discussion, this study is expected to contribute to the implementation of web-based information systems in MSME motorcycle rental businesses and to serve as a reference for the development of similar systems that emphasize operational efficiency, transparency of service status information, and the protection of business data confidentiality.The formula is written separately not in the sentence and equipped with numbering on the right. The formula is written using Microsoft equation.

Research Methods

Research Conceptual Framework

The following is the research conceptual framework that illustrates the flow from the identified problems to the proposed solutions and the outcomes achieved in the development of a web-based motorcycle rental application with real-time status updates and digital financial recording in PDF format.


Problem: A motorcycle rental system that still relies on manual or semi-digital processes leads to inefficient rental operations, difficulties in monitoring motorcycle availability, and a high risk of transaction errors and inaccuracies in financial reporting.
Requirements: A web-based motorcycle rental system that provides real-time motorcycle status updates, accurately records transactions, and generates digital transaction evidence and financial reports.
Method: Waterfall model combined with a descriptive qualitative approach. The system development applies a Client–Server Architecture, followed by a Real-Time Database System to support real-time data updates. The system adopts the Event-Driven Architecture concept to enable real-time mechanisms, implements Role-Based Access Control (RBAC) to ensure system security, and utilizes a Digital Transaction Recording System to support core system functionality.
Solution: A Web-Based Motorcycle Rental Application with Real-Time Status Updates and Digital Financial Recording, featuring:Real-time motorcycle status information (available / rented), Rental transactions managed by an administrator at the rental location, Automatic notifications and transaction receipts sent via WhatsApp, Rental history and a digital cash book, Financial reports generated in PDF format and A latest information feature for rental operational announcements
Results: A more efficient and well-controlled motorcycle rental process, Reduced risk of double bookings or rental conflicts, More structured, transparent, and accurate transaction and financial records and Improved service quality and increased customer trust

Research Location and Duration

This research was conducted at Merah Putih Motorcycle Rental, located in Sasa Subdistrict, Ternate City. System development and testing were carried out during the Field Work Practice (PKL/PKM/Pengabdian Kepada Masyarakat) period for approximately four months. Transactional and operational data were collected directly from ongoing motorcycle rental activities.
The data collection techniques used in this study include:

1. Interviews
   Interviews were conducted with the administrator or owner of the motorcycle rental business to obtain information regarding the rental workflow, limitations of the manual system, and application feature requirements.
2. Observation

https://doi.org/10.61536/ambidextrous.v3i1.422

Direct observation was conducted on the ongoing rental processes, including motorcycle data recording, availability status, and motorcycle return procedures.

3. Literature Review
   Data were collected from books, journals, and scientific references related to rental information systems, website development, and data management.
4. Tools and Materials
   The tools and materials used in this research include:
   - HTML, CSS, and JavaScript as web programming languages
   - Bootstrap for responsive user interface design
   - Supabase as the backend and database system
   - GitHub Pages and Vercel as hosting services
   - WhatsApp API for sending rental notifications
   - jsPDF for generating rental reports in PDF format
5. Open-Ended Questionnaire
   An open-ended questionnaire was conducted through direct interviews with motorcycle rental service managers to obtain feedback and suggestions for system development. The questions focused on application usability, clarity of motorcycle status information, and the expectations of both managers and users regarding future application feature development.

System Method or Algorithm

The following system methods are applied in the development of the website:

1. Client–Server Architecture (System Architecture)

The motorcycle rental application implements a client–server architecture, where the client consists of a web browser used by both users and administrators, while the server is responsible for handling application logic, authentication, and database management through the Supabase service.

The client sends requests such as retrieving motorcycle data, storing rental transaction records, and administrator login requests. The server then processes these requests and returns appropriate responses to the client.

Example of implementation in the code:

```
const SUPABASE_URL = "https://ifafji...supabase.co";
const SUPABASE_ANON = "e….OiJI..iIsIn.....";
const db = supabase.createClient(SUPABASE_URL, SUPABASE_ANON);
```

is connected to the Supabase server as the central component for data processing and database storage.

The client–server concept was introduced and extensively developed by Andrew S. Tanenbaum (1985) in the field of computer networks and distributed systems.

2. Real-Time Database System (Real-Time System Concept)

The system utilizes a real-time database, allowing any changes made to the bikes, rentals, and public_info tables to be immediately reflected in the application interface without requiring manual page refreshes.

This concept is essential for displaying up-to-date motorcycle status information, enabling users to instantly determine vehicle availability. The real-time system concept was introduced in studies on Real-Time Systems by Donald Gillies (1974) and further developed in modern reactive systems.

Example of real-time database implementation:

```
db.channel("bikes-change")
.on("postgres_changes", { event: "*", schema: "public", table: "bikes" }, safeRefresh)
.subscribe();
```

Any change in the bikes table will automatically trigger an update to the motorcycle list display.

3. Event-Driven Architecture (Real-Time Mechanism)

The application implements an event-driven architecture, in which the system reacts to specific events, such as a motorcycle being rented, returned, or when data is deleted or updated by the administrator.

Each event triggers subsequent processes, including updating the motorcycle status, sending WhatsApp notifications, and refreshing the user interface.

Example of an event mechanism in the rental process:

https://doi.org/10.61536/ambidextrous.v3i1.422

```
await db.from("rentals").insert({
  name,
  phone,
  bike_id: bikeId,
  duration: hours,
  total,
  start_time: start.toISOString(),
  end_time: end.toISOString(),
  status: "ongoing"});
await db.from("bikes").update({ status: "rented" }).eq("id", bikeId);
```

When a motorcycle rental event occurs, the system automatically records the transaction and updates the motorcycle status to rented. This concept was popularized by David C. Luckham (2002) through his research on Complex Event Processing.

4. Role-Based Access Control (RBAC) – System Security

To ensure data security, the system implements Role-Based Access Control (RBAC), which differentiates access privileges between administrators and general users.

Administrators have full access to manage motorcycle data, rental transactions, PDF reports, and public information, while general users are limited to viewing data without the ability to modify it.

Example of administrator access control implementation:

```
async function checkIsAdmin() {
  const { data } = await db.auth.getSession();
  if (!data.session) return false;
  const { data: admin } = await db
    .from("admins")
    .select("user_id")
    .eq("user_id", data.session.user.id)
    .single();
  return !!admin;}
```

With this mechanism, only accounts registered as administrators are authorized to access the system administration panel. Role-Based Access Control (RBAC) is a system security model developed by David Ferraiolo and Richard Kuhn (1992) at the National Institute of Standards and Technology (NIST).

5. Digital Transaction Recording System with CRUD (System Functionality)

The motorcycle rental application implements a Digital Transaction Recording System based on CRUD (Create, Read, Update, Delete) operations to digitally and systematically record all rental activities.

Implementation of CRUD within the system:

- Create: records a new rental transaction

```
await db.from("rentals").insert({ name, phone, bike_id, duration, total });
```

- Read: displays motorcycle data and rental history

```
const { data } = await db.from("bikes").select("*");
```

- Update: updates motorcycle status and transaction data

```
await db.from("bikes").update({ status: "available" }).eq("id", bikeId);
```

- Delete: deletes rental transaction data or motorcycle data

```
await db.from("rentals").delete().eq("id", rentalId);
```

All transactions are stored digitally and can be exported into PDF reports as documentation for rental management.

Conclusion of the System Method

By implementing a client–server architecture, a real-time database, event-driven architecture, RBAC-based security, and digital transaction recording using CRUD operations, this motorcycle rental application is able to improve the effectiveness of rental management, minimize recording errors, and provide fast, accurate, and secure information. The CRUD (Create, Read, Update, Delete) concept originates from the Relational Database Model theory introduced by Edgar F. Codd (1970).

https://doi.org/10.61536/ambidextrous.v3i1.422

System Performance Analysis and Measurement Method

The Waterfall method is a software development model that follows a structured and systematic process, starting from the analysis, design, implementation, testing, and maintenance stages. According to Pressman (2010) in Software Engineering: A Practitioner's Approach, the Waterfall model consists of a sequence of phases carried out linearly from the beginning to the end of the development process.

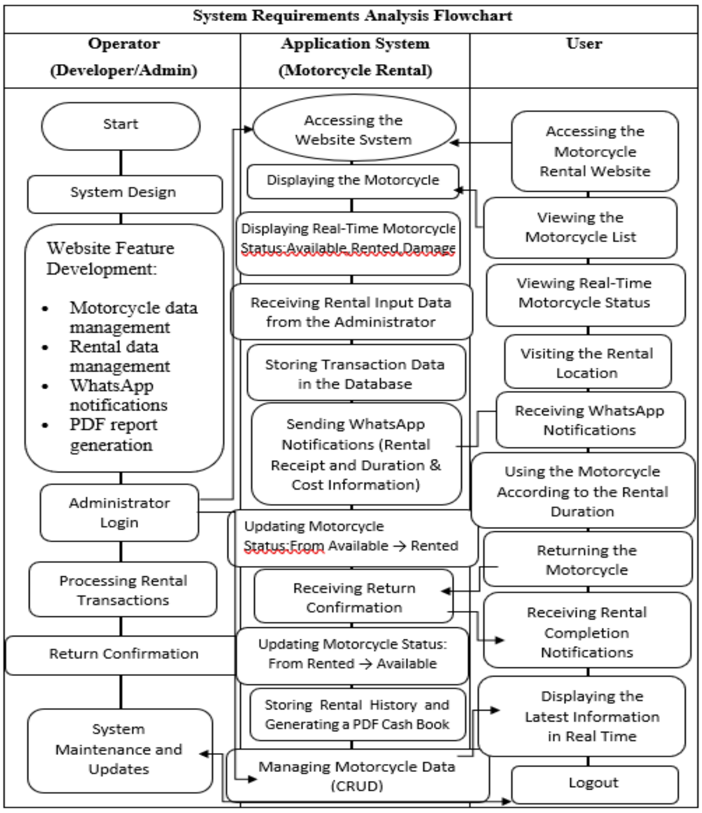
System development in this study employs the Waterfall method with the following stages (Pressman, 2010):

1. Requirement Analysis

   Functional requirements:
   - Motorcycle data management (CRUD).
   - Rental and customer data management.
   - Real-time motorcycle status information (available/rented).
   - Rental and return processes managed by the administrator.
   - Delivery of transaction receipts and status notifications via WhatsApp.
   - Generation of financial reports and digital cash books in PDF format.

   Non-functional requirements:
   - Web-based access.
   - A simple and user-friendly interface.
   - Secure access using Role-Based Access Control (RBAC).
   - System reliability and fast access performance.
2. Design
   - Design of system workflows using flowcharts.
   - Design of data structures using Entity Relationship Diagrams (ERD).
   - Design of user interfaces (UI/UX) for both public pages and administrator pages.
3. Implementation (Coding)
   - Front-end development using HTML, CSS, and JavaScript.
   - Data management using JSON local storage or a database system.
   - Integration of WhatsApp notifications for sending transaction receipts.
   - Generation of transaction reports and cash books in PDF format.
4. Testing
   - Black Box Testing to ensure that all system functions operate according to the specified requirements.
   - System trials conducted by rental administrators to evaluate usability and data accuracy.
5. Maintenance
   - Bug fixing and error correction.
   - System data and feature updates.
   - System adjustments based on the operational needs of the motorcycle rental business.

https://doi.org/10.61536/ambidextrous.v3i1.422

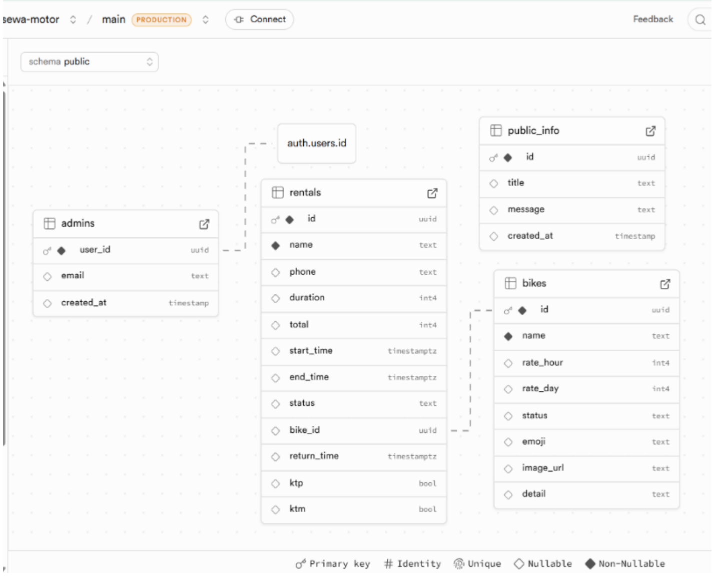

System Requirements Analysis Flowchart
Operator
(Developer/Admin)
Application System
(Motorcycle Rental)
User
Start
System Design
Website Feature Development:
• Motorcycle data management
• Rental data management
• WhatsApp notifications
• PDF report generation
Administrator Login
Processing Rental Transactions
Return Confirmation
System Maintenance and Updates
Accessing the Website Svstem
Displaying the Motorcycle
Displaying Real-Time Motorcycle Status:Available, Rented, Damage
Receiving Rental Input Data from the Administrator
Storing Transaction Data in the Database
Sending WhatsApp Notifications (Rental Receipt and Duration & Cost Information)
Updating Motorcycle Status:From Available → Rented
Receiving Return Confirmation
Updating Motorcycle Status: From Rented → Available
Storing Rental History and Generating a PDF Cash Book
Managing Motorcycle Data (CRUD)
Accessing the Motorcycle Rental Website
Viewing the Motorcycle List
Viewing Real-Time Motorcycle Status
Visiting the Rental Location
Receiving WhatsApp Notifications
Using the Motorcycle According to the Rental Duration
Returning the Motorcycle
Receiving Rental Completion Notifications
Displaying the Latest Information in Real Time
Logout

Results and Discussion

System Implementation

System implementation is the stage in which the design results are applied to create an executable system. At this stage, all designs developed in the previous chapters are implemented into a web-based motorcycle rental application using web technologies and the Supabase database.

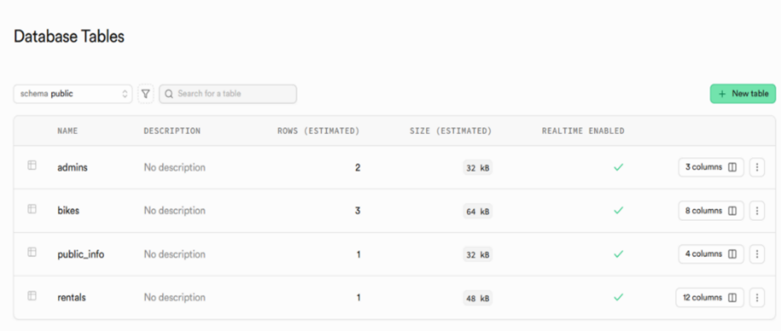
Entity Relationship Diagram (ERD) of the Motorcycle Rental Application (Entities and Attributes)



Database Implementation

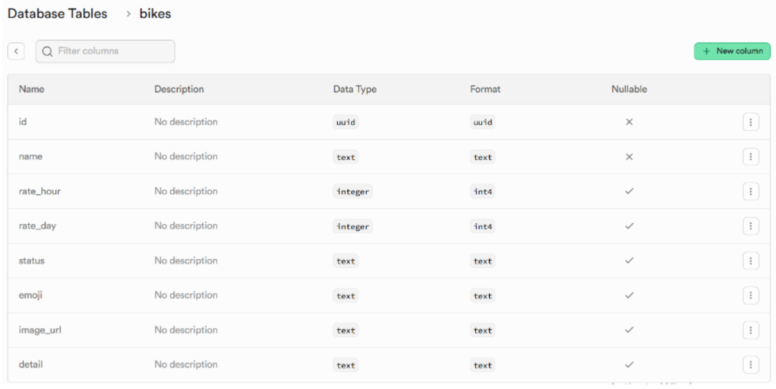
The database is implemented using Supabase with several main tables, including:

https://doi.org/10.61536/ambidextrous.v3i1.422



Database Tables

NAME	DESCRIPTION	ROWS (ESTIMATED)	SIZE (ESTIMATED)	REALTIME ENABLED	
admins	No description	2	32 kB	✓	3 columns
bikes	No description	3	64 kB	✓	8 columns
public_info	No description	1	32 kB	✓	4 columns
rentals	No description	1	48 kB	✓	12 columns



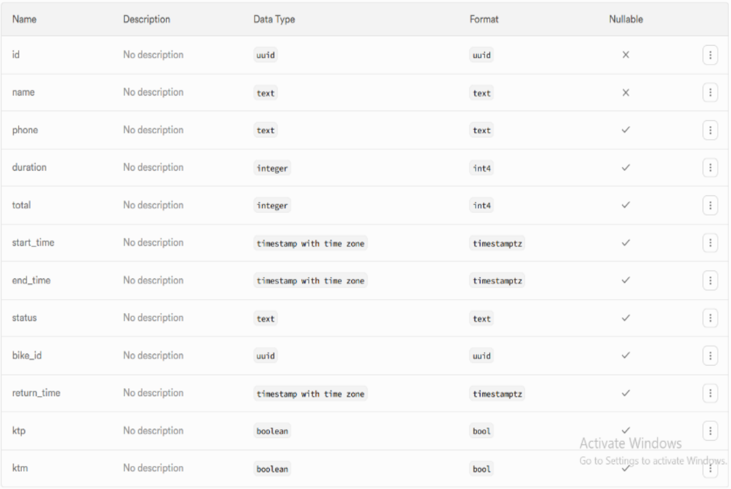
Database Tables › bikes

Name	Description	Data Type	Format	Nullable
id	No description	uuid	uuid	×
name	No description	text	text	×
rate_hour	No description	integer	int4	✓
rate_day	No description	integer	int4	✓
status	No description	text	text	✓
emoji	No description	text	text	✓
image_url	No description	text	text	✓
detail	No description	text	text	✓

1. Bikes Table

- Stores motorcycle data
- Attributes: id, name, rate_hour, rate_day, status, image_url, detail

2. Rentals Table

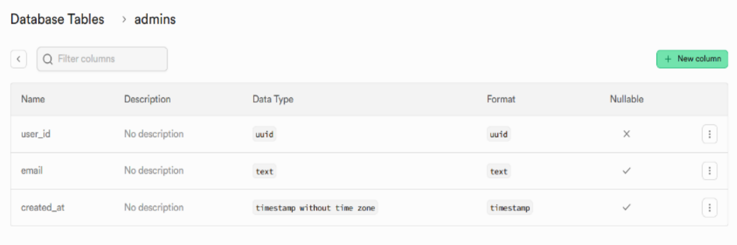


Name	Description	Data Type	Format	Nullable
id	No description	uuid	uuid	×
name	No description	text	text	×
phone	No description	text	text	✓
duration	No description	integer	int4	✓
total	No description	integer	int4	✓
start_time	No description	timestamp with time zone	timestamptz	✓
end_time	No description	timestamp with time zone	timestamptz	✓
status	No description	text	text	✓
bike_id	No description	uuid	uuid	✓
return_time	No description	timestamp with time zone	timestamptz	✓
ktp	No description	boolean	bool	✓
ktm	No description	boolean	bool	✓

- Stores rental transaction data
- Attributes: id, name, phone, duration, total, start_time, end_time, status, bike_id, return_time, ktp, ktm

3. Admins Table

https://doi.org/10.61536/ambidextrous.v3i1.422



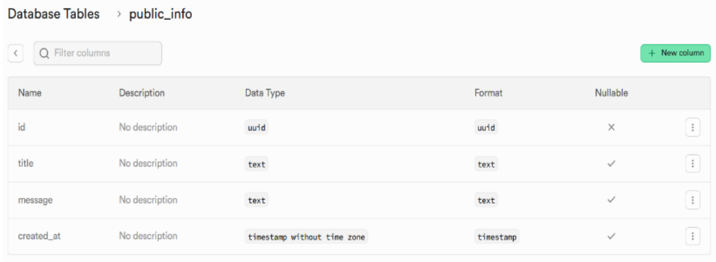
Database Tables › admins

Filter columns
+ New column

Name	Description	Data Type	Format	Nullable
user_id	No description	uuid	uuid	×
email	No description	text	text	✓
created_at	No description	timestamp without time zone	timestamp	✓

- Stores administrator data
- Attributes: user_id, email, created_at

4. Public_Info Table



Database Tables › public_info

Filter columns
+ New column

Name	Description	Data Type	Format	Nullable
id	No description	uuid	uuid	×
title	No description	text	text	✓
message	No description	text	text	✓
created_at	No description	timestamp without time zone	timestamp	✓

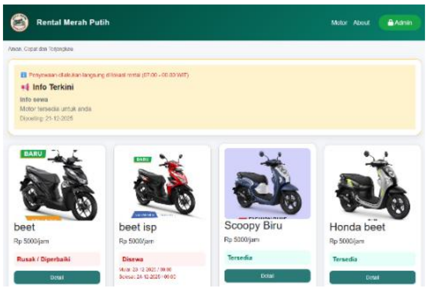
- Stores system information or announcements
- Attributes: id, title, message, created_at

The relationships among these tables enable the system to display motorcycle status in real time and to manage rental and return histories effectively.

User Interface Design

The motorcycle rental system interface is designed to be simple, responsive, and easy to use for both renters and administrators. The main interface components include:

- Main Page: Displays a list of motorcycles in a card-based layout, motorcycle status (available/rented), real-time latest information, and a button to rent a motorcycle. Users can directly view motorcycle details, rental rates, and rental duration information.
- Admin Page: A dedicated administrator panel for managing motorcycle data, rental history, and public information. Administrators can perform CRUD (Create, Read, Update, Delete) operations on motorcycle data and rental records, publish the latest information, and add new motorcycles.



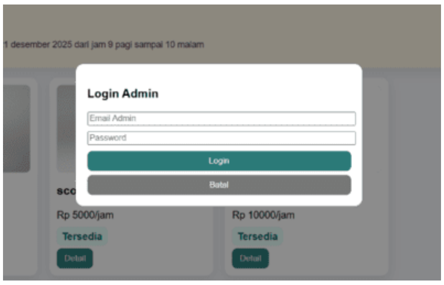
Info Terkini
Info sewa
Motor tersedia untuk anda
Diposting: 21-12-2025

beet
Rp 5000/jam
Rusak / Diperbaiki

beet isp
Rp 5000/jam
Disewa

Scoopy Biru
Rp 5000/jam
Tersedia

Honda beet
Rp 5000/jam
Tersedia



1 desember 2025 dari jam 9 pagi sampai 10 malam

Login Admin

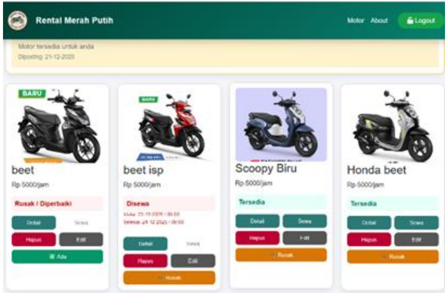
Email Admin
Password

Rp 5000/jam
Tersedia

Rp 10000/jam
Tersedia

Figure 1 Main Page

Figure 2 Admin Login Page



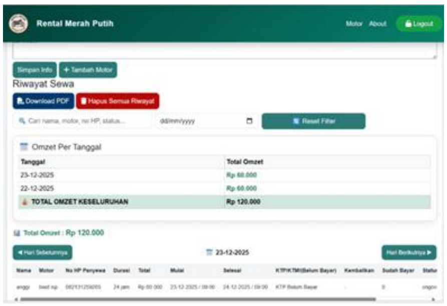
Motor tersedia untuk anda
Diposting: 21-12-2025

beet
Rp 5000/jam
Rusak / Diperbaiki

beet isp
Rp 5000/jam
Disewa

Scoopy Biru
Rp 5000/jam
Tersedia

Honda beet
Rp 5000/jam
Tersedia



Riwayat Sewa

Omzet Per Tanggal

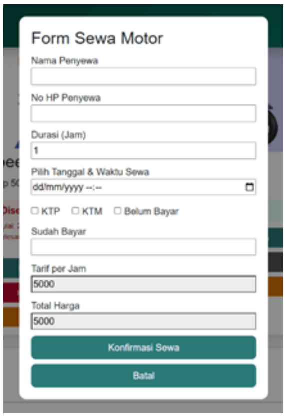
Tanggal	Total Omzet
23-12-2025	Rp 60.000
22-12-2025	Rp 60.000
TOTAL OMZET KESELURUHAN	Rp 120.000

Total Omzet : Rp 120.000

Nama	Motor	No HP Penyewa	Durasi	Total	Mulai	Selesai	KTP/KTM(Belum Bayar)	Kembalikan	Sudah Bayar	Status
anggi	beet isp	[illegible]	24 jam	Rp 60.000	23-12-2025 / 09:00	24-12-2025 / 09:00	KTP Belum Bayar	-	0	[illegible]

https://doi.org/10.61536/ambidextrous.v3i1.422

Figure 3 Admin Dashboard



Form Sewa Motor

Nama Penyewa

No HP Penyewa

Durasi (Jam): 1

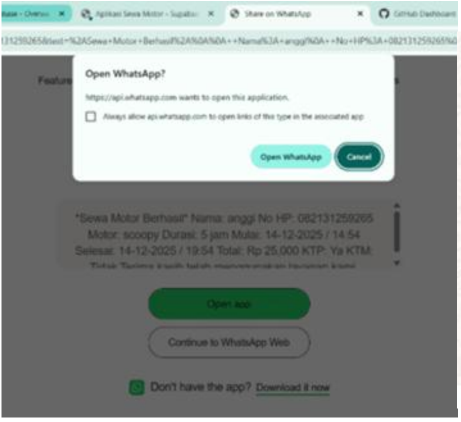
Pilih Tanggal & Waktu Sewa: dd/mm/yyyy --:--

□ KTP □ KTM □ Belum Bayar

Sudah Bayar

Tarif per Jam: 5000

Total Harga: 5000

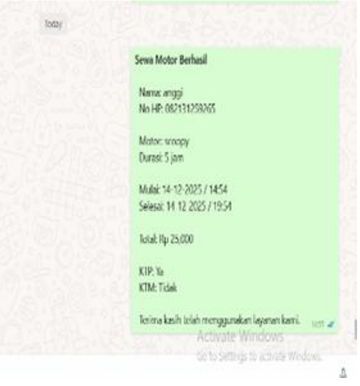


Open WhatsApp?

https://api.whatsapp.com wants to open this application.

□ Always allow api.whatsapp.com to open links of this type in the associated app

Sewa Motor Berhasil Nama: anggi No HP: 082131259265 Motor: scoopy Durasi: 5 jam Mulai: 14-12-2025 / 14:54 Selesai: 14-12-2025 / 19:54 Total: Rp 25.000 KTP: Ya KTM:



Sewa Motor Berhasil

Nama: anggi
No HP: 082131259265

Motor: snopy
Durasi: 5 jam

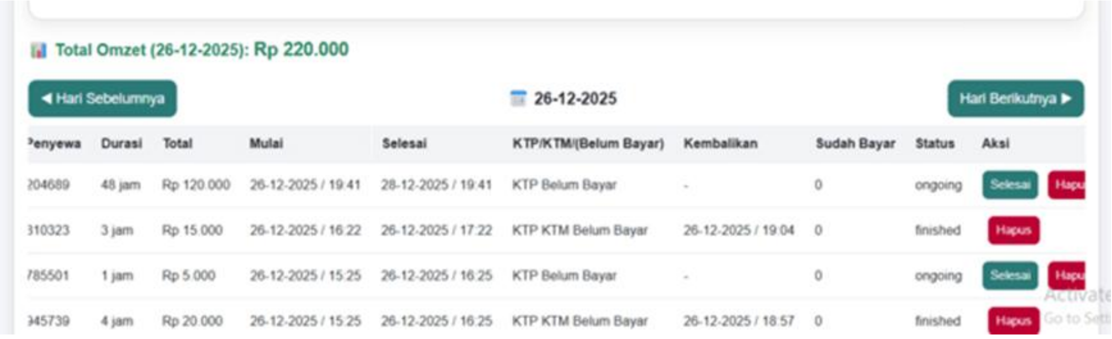
Mulai: 14-12-2025 / 14:54
Selesai: 14 12 2025 / 19:54

Total: Rp 25,000

KTP: Ya
KTM: Tidak

Terima kasih telah menggunakan layanan kami.

Figure 4 Rental Page Figure 5 Sending Rental Confirmation via WhatsApp to the Renter

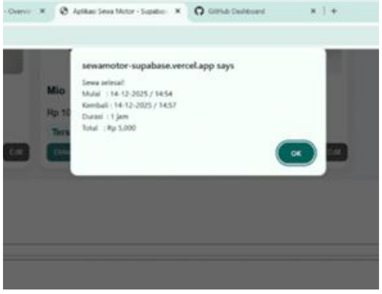


Total Omzet (26-12-2025): Rp 220.000

◀ Hari Sebelumnya | 26-12-2025 | Hari Berikutnya ▶

Penyewa	Durasi	Total	Mulai	Selesai	KTP/KTM/(Belum Bayar)	Kembalikan	Sudah Bayar	Status	Aksi
204689	48 jam	Rp 120.000	26-12-2025 / 19:41	28-12-2025 / 19:41	KTP Belum Bayar	-	0	ongoing	Selesai Hapu
310323	3 jam	Rp 15.000	26-12-2025 / 16:22	26-12-2025 / 17:22	KTP KTM Belum Bayar	26-12-2025 / 19:04	0	finished	Hapus
785501	1 jam	Rp 5.000	26-12-2025 / 15:25	26-12-2025 / 16:25	KTP Belum Bayar	-	0	ongoing	Selesai Hapu
345739	4 jam	Rp 20.000	26-12-2025 / 15:25	26-12-2025 / 16:25	KTP KTM Belum Bayar	26-12-2025 / 18:57	0	finished	Hapus

Figure 6 Automatic Input into Rental History



sewamotor-supabase.vercel.app says

Sewa selesai!
Mulai : 14-12-2025 / 14:54
Kembali : 14-12-2025 / 14:57
Durasi : 1 jam
Total : Rp 5,000

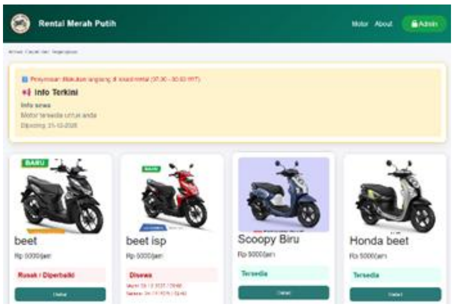
Terima kasih telah menggunakan layanan kami.

Sewa Motor Berhasil

Sewa Motor Telah Selesai

Terima kasih telah menggunakan layanan kami.

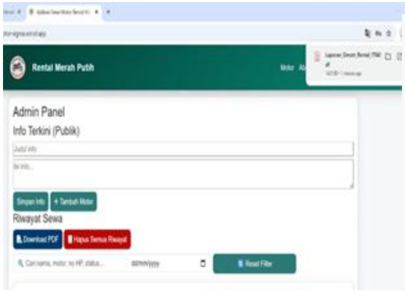
Sewa Motor Berhasil



Rental Merah Putih

Info Terkini

Figure 7 WhatsApp Notification for Rental Completion Confirmation Figure 8 Motorcycle Status Updated to Available

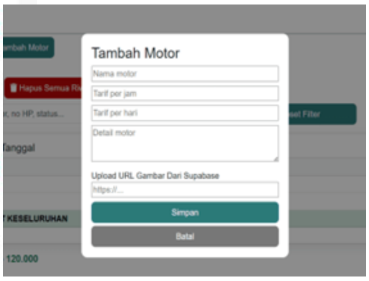


Rental Merah Putih

Admin Panel

Info Terkini (Publik)

Riwayat Sewa



Tambah Motor

Nama motor

Tarif per jam

Tarif per hari

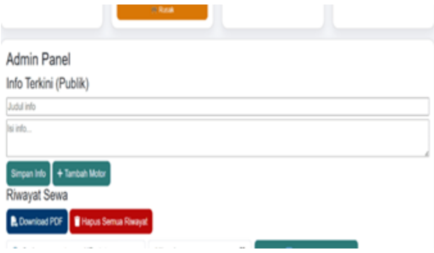
Detail motor

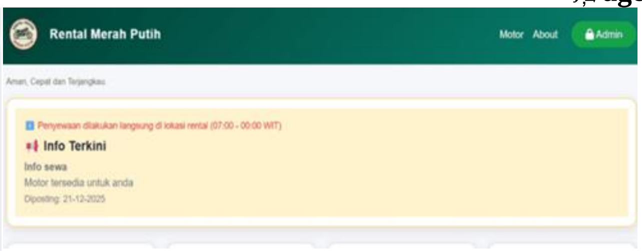
Upload URL Gambar Dari Supabase

https://...

Figure 9 Downloading the PDF Cash Book, Figure 10 PDF Cash Book, Figure 11 Rental and Motorcycle CRUD System (Add Motorcycle)

https://doi.org/10.61536/ambidextrous.v3i1.422





Aman, Cepat dan Terjangkau

Penyewaan dilakukan langsung di lokasi rental (07:00 - 00:00 WIT)

Info Terkini

Info sewa

Motor tersedia untuk anda

Diposting: 21-12-2025

Figure 12 Create Latest Information

Figure 13 Real-Time Latest Information

System Testing

System testing is conducted to ensure that the motorcycle rental system operates according to user requirements (both renters and administrators) and is free from major errors. The testing process includes functional testing (Black Box Testing) and user testing (usability testing).

Functional Testing (Black Box Testing)

Testing is performed on all main system features to verify that the inputs and outputs function as expected. The results of the testing are presented in the following table:

No	Tested Feature	Test Description	Expected Result	Actual Result	Status
1	Login & Logout	Administrator login and logout	Administrator can successfully log in and log out of the panel	As expected	OK
2	View Motorcycle Cards & Details	Users view the list of motorcycles and their details	Motorcycle status, rental rates, and details are displayed accurately	As expected	OK
3	Motorcycle Card CRUD	Administrator adds, edits, and deletes motorcycles	Motorcycle data is successfully added, updated, or deleted	As expected	OK
4	Rental History CRUD	Administrator deletes or updates rental history	Rental history is updated according to the input	As expected	OK
5	Motorcycle Rental Feature	User visits the rental location and the administrator inputs the rental data	Rental data is saved, motorcycle status changes, and total price is calculated correctly	As expected	OK
6	Add New Motorcycle Card	Administrator adds a new motorcycle	The new motorcycle is displayed on the main page	As expected	OK
7	Real-Time Latest Information	Administrator adds information that is displayed in real time	Information appears on the main page without page reload	As expected	OK
8	WhatsApp Feature	Rental is successful → notification sent via WhatsApp	WhatsApp message is sent according to the defined format	As expected	OK
9	Completion Feature & WhatsApp Notification	Motorcycle is returned → notification sent	Rental completion message appears and total price is accurate	As expected	OK
10	Rental Revenue & PDF Report Feature	Administrator views total rental revenue and downloads the report	The system displays total rental revenue and generates a PDF revenue report	As expected	OK

Functional Testing Analysis

The testing results indicate that all system features operate according to the specified requirements. No major errors were identified. Some minor feedback was related to the loading speed of real-time information and WhatsApp notifications when network conditions were poor. Overall, the system can be effectively used by both renters and administrators.

User Testing (Usability Test)

User testing was conducted to evaluate the ease of use of the interface, the clarity of motorcycle information, and the usefulness of system features. The respondents consisted of regular motorcycle

https://doi.org/10.61536/ambidextrous.v3i1.422

renters and system administrators.

No	Respondent	Role	Tested Feature	Perceived Advantages	Weaknesses / Feedback	Developer Improvements
1	Renter	Student	Viewing real-time motorcycle cards (available, rented, or damaged status)	Easy to select a motorcycle; information is complete and clear	Renters cannot directly perform the rental process through the system	No direct rental feature was added to avoid data and scheduling conflicts. The rental process continues to be conducted directly at the rental location with administrator approval
2	Admin A	Business Owner	Motorcycle damage status	Facilitates vehicle control and monitoring	No indicator for damaged motorcycles	A damaged motorcycle status feature was added to the system
3	Admin B	Staff	Payment status (paid/unpaid)	Rental record management is more organized	Payment feature was not available	Paid and unpaid status features were added
4	Admin B	Staff	Total rental cost calculation	Rental costs are calculated automatically	Total rental cost was inaccurate for a 24-hour duration	The calculation logic was revised according to time duration: IDR 5,000 per hour, IDR 60,000 per 24 hours. For example, 26 hours results in a total of IDR 70,000. A 15-minute tolerance is also applied for late returns, as implemented in the logic: if (extraMinutes > 15) hours += 1;

Usability Test Analysis

The usability test results indicate that the majority of users found the system easy to use, particularly the rental process, WhatsApp notification feature, and real-time information display. Some minor feedback was provided regarding button sizes, data access speed, and the layout of the PDF reports. These inputs serve as valuable evaluation materials for future system development.

Discussion

This discussion aims to analyze the results of the implementation of the web-based motorcycle rental system, address the formulated research problems, and achieve the established research objectives.

- Real-Time Vehicle Status

One of the main issues identified in this study was how to design and develop a motorcycle rental application capable of displaying vehicle status in real time. Based on system testing, the real-time information feature and detailed motorcycle cards successfully display motorcycle status (available, currently rented, or returned) instantly without requiring page reloads. This indicates that the first research objective—developing a motorcycle rental application with a real-time vehicle status system—has been achieved. The system utilizes a cloud-based database integrated with the web interface, ensuring that any status changes made by administrators or users are immediately reflected on the main page. As a result, users can quickly and accurately view motorcycle availability, facilitating the rental process.

- Transaction Recording and PDF Reporting

The second research problem concerns the implementation of a structured digital transaction and financial recording system. In this system, every rental transaction is automatically recorded in the rental history and can be accessed by administrators. The PDF feature allows administrators or users to export the entire rental and transaction history into a format that is easy to print or store. Based on functional testing, all recorded transaction data accurately reflects user input, including motorcycle details, rental duration, pricing, and total cost. The generated PDF reports are complete and well-organized, thereby facilitating financial documentation and administrative management. Thus, the second research

https://doi.org/10.61536/ambidextrous.v3i1.422

objective—implementing a digital transaction recording system with PDF reports—has been successfully achieved.

- Role-Based Security System (RBAC)

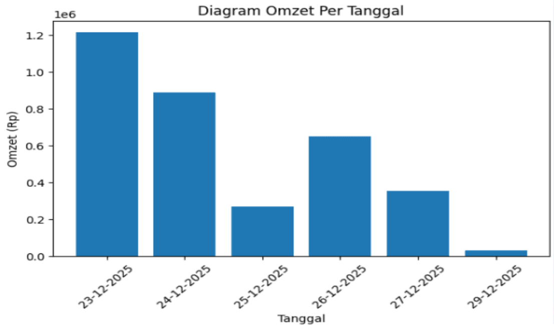
The third research problem addresses the implementation of role-based security to differentiate access rights between administrators and general users. This motorcycle rental system applies Role-Based Access Control (RBAC), in which:

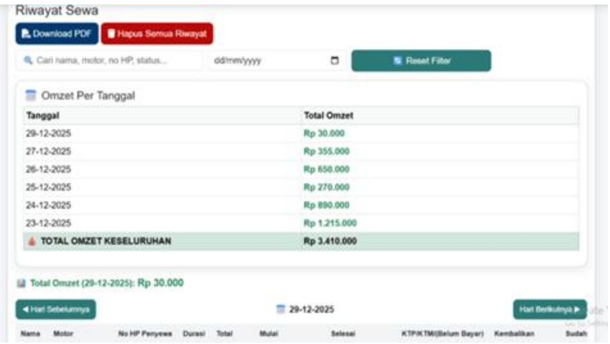
• Administrators have full access to perform CRUD operations on motorcycle data and rental history, add the latest information, and view PDF reports.

• General users are limited to viewing the motorcycle list, real-time status information (Available/Rented/Damaged), completing rentals at the rental location, and receiving WhatsApp notifications.

The results of functional and usability testing indicate that the RBAC mechanism operates as expected. Administrators can manage all system data without restrictions, while general users are unable to access administrative pages or modify motorcycle data. This successfully addresses the third research objective, which is to implement a data management and security system using RBAC, thereby ensuring that system data remains protected from unauthorized access.

Revenue Chart of Rental Merah Putih





Riwayat Sewa

Omzet Per Tanggal

Tanggal	Total Omzet
29-12-2025	Rp 30.000
27-12-2025	Rp 355.000
26-12-2025	Rp 650.000
25-12-2025	Rp 270.000
24-12-2025	Rp 890.000
23-12-2025	Rp 1.215.000
TOTAL OMZET KESELURUHAN	Rp 3.410.000

Total Omzet (29-12-2025): Rp 30.000

29-12-2025

Nama	Motor	No HP Penyewa	Durasi	Total	Mulai	Selesai	KTP/KTM(Belum Bayar)	Kembalikan	Sudah

The table in the Rental Merah Putih application displays a chart of daily motorcycle rental revenue during the observation period. It shows that the highest revenue occurred on December 23, 2025, while the lowest revenue was recorded on December 29, 2025. These fluctuations indicate that the motorcycle rental rate is dynamic and influenced by the renters' daily activities. To maintain the confidentiality of MSME (Micro, Small, and Medium Enterprises) business financial data, this study presents revenue data only within a limited time frame, with the primary focus on testing system functionality and the accuracy of transaction recording.

Conclusion

This research successfully developed a web-based motorcycle rental application featuring real-time vehicle status using Supabase, digital transaction recording with CRUD functions, WhatsApp notifications, and PDF financial reports limited to three days to protect MSME data confidentiality. Key findings show the system performs optimally through Black Box testing, with all features like automatic status updates, RBAC access control, and accurate cost calculations fully functional, thereby enhancing operational efficiency at Merah Putih rental and minimizing manual errors. Usability testing by renters and admins confirmed the Bootstrap-responsive interface's ease of use.

However, limitations include single-case development at one rental site, lack of long-term revenue analysis, and reliance on internet connectivity for real-time updates. Future research suggestions involve multi-location testing, online payment integration, and extended data analytics. Practically, this application offers MSME motorcycle rentals transparency in vehicle status, digital financial documentation, and professional management, supporting small business digital transformation in Indonesia.

https://doi.org/10.61536/ambidextrous.v3i1.422

References

Anggara, P. W., Wijaya, I. N. Y., & Putri Astawa, N. L. P. N. S. (2022). Sistem informasi rental motor berbasis web di Bali Scooters. Jurnal Teknologi Informasi & Komputer.

Bintari, W., Ghozali, M. I., & Sugiharto, W. H. (2024). Sistem informasi manajemen dan monitoring (Simamo) rental mobil berbasis IoT. JUMINTAL.

Bintoro, P., Farida, M., Zulkifli, Z., Ardhy, F., Andika, T. H., Yulia, A. F., & Andini, D. Y. A. (2025). Sistem informasi jadwal dokter berbasis web dengan fitur CRUD di RSU Az-Zahra Kalirejo (Jurnal Pengabdian Kepada Masyarakat Ungu). Jurnal Pengabdian Kepada Masyarakat Ungu (ABDI KE UNGU), 7(2), 131–137.

Dinata, W., Yusup, Y., & Prihati, Y. (2024). Pengembangan aplikasi rental motor berbasis web di Elnatan Garage. IPM2KPE Journal.

Gusriyanti, D. A. (2025). Analisis dan perancangan web sistem informasi rental (Jurnal Management and Information Systems). Universitas Dinamika Bangsa.

Guyen, H. J., & Prihanto, A. (2025). Real time notifikasi informasi produk e commerce menggunakan WhatsApp push message dan bot menu. Jurnal Sistematik, 6, 903–909.

Handayani, M., Utna, S., & Anggraini, P. (2023). Pengaruh UI/UX terhadap kepuasan pengguna aplikasi rental motor. Jurnal Sistem Informasi dan Informatika.

Mistang, S., & Aswandi, D. (2024). Aplikasi marketplace rental mobil di Kota Makassar berbasis web. Jurnal Teliska.

Nugroho, R. S., & Anton, A. (2024). Design of web based car rental information system using Extreme Programming at CV. Nugroho. Jurnal Techno Nusa Mandiri.

Salsabilah, S., Harsya, Mentari, K., Monicasari, S., & Wijaya, H. (2024). RENTALIN APPS: Development of a motorbike rental application using design thinking. Jurnal Bisnis & Komunikasi Digital

https://doi.org/10.61536/ambidextrous.v3i1.422