PROP-AI

PROJECT DOCUMENTATION

SOFTWARE ENGINEERING COURSE

CREATED BY GROUP 12

Ganendra Garda Pratama	**2306250642**
Raka Arrayan Muttaqien	**2306161800**
Muhammad Raihan Mustofa	**2306161946**

FAKULTAS TEKNIK

PROGRAM STUDI TEKNIK KOMPUTER

UNIVERSITAS INDONESIA

DEPOK

TABLE OF CONTENTS

CHAPTER I

INTRODUCTION

1.1 BACKGROUND

The development of information technology and the internet has brought significant changes to various sectors, including the property industry. Buying and selling activities that were once conducted conventionally are now shifting to digital platforms that offer greater convenience, speed, and accessibility. However, there are still many challenges to be faced, such as a lack of price transparency, difficulties in determining fair market value, and limited smart features in existing property buying and selling platforms. These conditions have driven the need for innovation in the form of a more modern, intuitive, and artificial intelligence-based system.

To address these issues, a web-based property trading platform equipped with smart price recommendation features has been designed. This platform features a modern, responsive, and easy-to-use interface, providing an optimal experience for both sellers and buyers. Sellers can easily market their properties, while buyers can efficiently search for and find their dream properties through a well-organized system.

In addition to offering ease of use, this system also integrates a machine learning model that can analyze various property attributes, such as location, building area, number of rooms, and available facilities. This analysis produces accurate and objective price estimates, which not only help buyers assess the fairness of a property's price, but also support sellers in setting realistic prices. Thus, the technology applied is able to provide added value in the decision-making process.

The optimal price recommendation feature available on this platform is expected to increase transparency and efficiency in the property market. Sellers receive competitive pricing advice without losing profits, while buyers gain greater confidence through clear data and transparent price comparisons. Overall, this platform is expected to be an innovative solution in supporting fairer, more efficient, and modern technology-based property transactions.

1.2 PROJECT OBJECTIVES

The main objective of this project is to design, develop, and implement a web-based property trading platform integrated with a machine learning-based smart price recommendation system. This platform is intended to improve efficiency, transparency, and accuracy in property transaction decision-making for sellers and buyers.

To achieve this goal, several specific objectives have been set:

1. To build a full-stack system architecture with a back-end using Node.js to manage APIs and business logic, as well as a front-end based on Single Page Application (SPA) using React.js or Vue.js to deliver a responsive and interactive user interface.

2. To develop an accurate property price prediction model using machine learning algorithms such as regression or random forest with Python libraries (Scikit-learn, Pandas) to produce objective and reliable price estimates.

3. To implement an optimal price recommendation feature that helps sellers set competitive prices and buyers assess the fairness of property values based on data analysis.

4. To design and manage a reliable database system using PostgreSQL or MongoDB to store listing, user, and transaction data efficiently and scalably.

5. To improve transparency in the property market by providing a data-driven platform that supports more informed and fair decision-making for all parties involved

1.3 SIMILAR COMPETITORS

Here are some existing event/crowd management platforms/event applications, and how they compare to PROP-AI.

Competitor	What they do well	Their gaps vs PROP-AI
Rumah123	It has an extensive property database, a user-friendly interface, support from official real estate agents, and detailed search filter features.	Does not yet have AI-based smart price recommendation features. Price estimates still depend on the general market or agents, so price transparency is not optimal for buyers who want objective analysis.
OLX Properti	It has a huge user base, integration with the OLX platform in general, and an easy and fast property listing upload process.	The main focus is only on advertising and listings. There is no AI support for price analysis, competitive price recommendations, or market price transparency.
Lamudi	International reputation, a strong network of real estate agents, and the availability of articles and education related to real estate.	It has not yet integrated machine learning to provide automatic price recommendations, making it less helpful for buyers in objectively assessing price fairness.

1.4 LITERATURE REVIEW

The research conducted by Fauzi et al. (2025) focused on developing a property price prediction model in Indonesia using the Random Forest algorithm. In their study, various factors such as land area, building area, number of rooms, and distance to the city center were analyzed to build an accurate predictive model. The results showed that distance to the city center was the most influential variable, followed by building area and land area. With an evaluation metric of Mean Absolute Error

(MAE) of Rp2.48 billion and Root Mean Squared Error (RMSE) of Rp2.89 billion, this study successfully proved that Random Forest is an effective algorithm capable of handling non-linear relationships in property price data, providing better results than conventional methods. This work provides a strong foundation for this project, as it directly validates the selection of the Random Forest algorithm for price prediction tasks and identifies key features relevant to the property market in Indonesia. This project aims to implement a similar model, not only as an academic study, but also to integrate it into a functional transactional web platform.

We also found a paper by Syafarina and Zaenuddin (2023) titled "Implementation of the Streamlit Framework for Predicting House Prices Using Linear Regression." This study developed a web-based application to predict house prices in the city of Banjarmasin. Their system architecture uses a linear regression model to make predictions and the Streamlit framework to build an interactive user interface. This study successfully produced a functional application with a model accuracy rate of 67.8%. Although their approach successfully validated the concept of combining predictive models with web interfaces, this project aims to build a more comprehensive and scalable solution. Our project will use more complex algorithms such as Random Forest, which is capable of capturing more complex data patterns than Linear Regression. Additionally, the proposed full-stack architecture with Node.js on the back-end and React.js/Vue.js on the front-end is designed for a more mature trading platform, surpassing the capabilities of Streamlit, which is generally more suitable for data visualization and rapid prototyping.

1.5 FUNCTION AND MODULE

This system is designed with a modular architecture to ensure scalability and ease of management. Core functions are divided into the following modules:

1. Front-End Application Module: This is the component that directly interacts with users, including buyers, sellers, and administrators. Developed as a modern web application, this module provides an intuitive interface for searching and filtering properties, viewing listing details, registering/logging

in, as well as a dedicated dashboard for users to manage properties and for administrators to moderate content and monitor platform activity.

2. Back-End Service Module: This module functions as the system hub that manages all business logic. It includes a centralized API that acts as a communication bridge between the front-end application and the artificial intelligence module, handles user authentication, processes data, and ensures real-time data synchronization so that all listing information is always accurate and up to date.

3. Artificial Intelligence (AI) Module: This module is the core of the platform's intelligent features. Using machine learning models (such as Regression and Random Forest), this module is responsible for analyzing various property attributes to predict and provide optimal price recommendations for sellers, as well as presenting market price data visualizations to help buyers make decisions.

4. Database Module: Utilizing PostgreSQL or MongoDB, this module is responsible for persistent data storage. It covers all crucial information such as property listing data, user profiles, and transaction history, designed to support the platform's performance, security, and scalability.

1.6 TOOLS

The development of this platform will utilize the following technologies and software:

- Front-End Development (User & Admin Applications):
    - Languages & Frameworks: JavaScript, React.js or Vue.js
    - Markup & Styling: HTML, Tailwind CSS
- Back-End Development (Server & Database):
    - Runtime Environment: Node.js

- API: REST API for communication between services
- Database: PostgreSQL or MongoDB

- Artificial Intelligence (AI & Machine Learning):
  - Programming Language: Python
  - Library: Scikit-learn (for Regression & Random Forest models), Pandas (for data pre-processing)
  - Data Visualization: Matplotlib, Seaborn

- Project Management & Version Control:
  - Version Control: Git / GitHub
  - Task Management: Trello

1.7 RISK ANALYSIS

Proactive analysis of potential risks has been conducted to ensure timely and successful project completion.

1. Technical Risks
   - Integration between web platforms and machine learning models can encounter obstacles, particularly in terms of performance and compatibility.
   - The system may experience response delays or inaccurate price prediction results.
2. Data Quality Risk
   - The property data collected may be incomplete, inconsistent, or biased.
   - The AI model provides incorrect or misleading price recommendations.

3. Security and Privacy Risks
    - Potential leakage of user data (identity, transactions, or property listings).
    - Loss of user trust, financial losses, and even legal problems.
4. System Scalability Risks
    - An increase in the number of users and property data can overload servers and databases.
    - Decreased application performance or system downtime.
5. Market Acceptance Risk
    - Users (sellers/buyers) are reluctant to use a new platform because they are already accustomed to major competitors.
    - The number of users is low, so the platform ecosystem is not yet established.

CHAPTER II

PROJECT MANAGEMENT

2.1. ROLES AND RESPONSIBILITIES

2.1.1 Ganendra Garda Pratama (Project Lead / System Architect & Database Collection)

Main Objectives : Oversee the entire project lifecycle, managing assignments, making sure the collaboration runs smoothly, designs the workflow and the structure of the whole software enviroment. Scrapes the data needed for the project.

- Lead project planning, defining the structure of the team, assigns roles and responsibilities. Supervise all development stages.
- Act as System design, ensuring stable, consistent, efficient, and secure system design.
- Create and maintain UML diagrams as a guide for other members.
- Tracks progress using Trello and ensures accountability. Making sure the milestones are completed according to schedule.
- Defines and monitors KPIs for system performance, audit, and evaluation.
- Identify, evaluate, and mitigate project risks.
- Manages communications and coordination.
- Develops a scraper to collect data for the AI Models.

2.1.2 Muhammad Raihan Mustofa (Client App Developer & UI/UX Developer)

Main Objectives : Creates the client web app. Designs how the user interacts and sees the app to ensure a pleasant, seamless, responsive, and user-friendly experience.

2.1.2.1 Client App Developer

- Build the web app frontend using React and TailwindCSS
- Making sure the web app is fast-loading.

- Securing the web app from cross-site scripting or spoofing.
- Integrates the frontend with server API

2.1.2.2 UI/UX Developer

- Designs the UI with Figma
- Making a pleasant user experience based on the UML

2.1.3 Raka Arayyan Muttaqien (AI Engineer & Database Management)

Main Objectives : Trains the AI Models needed for the project. Manages the database collection.

2.1.3.1 AI Engineer

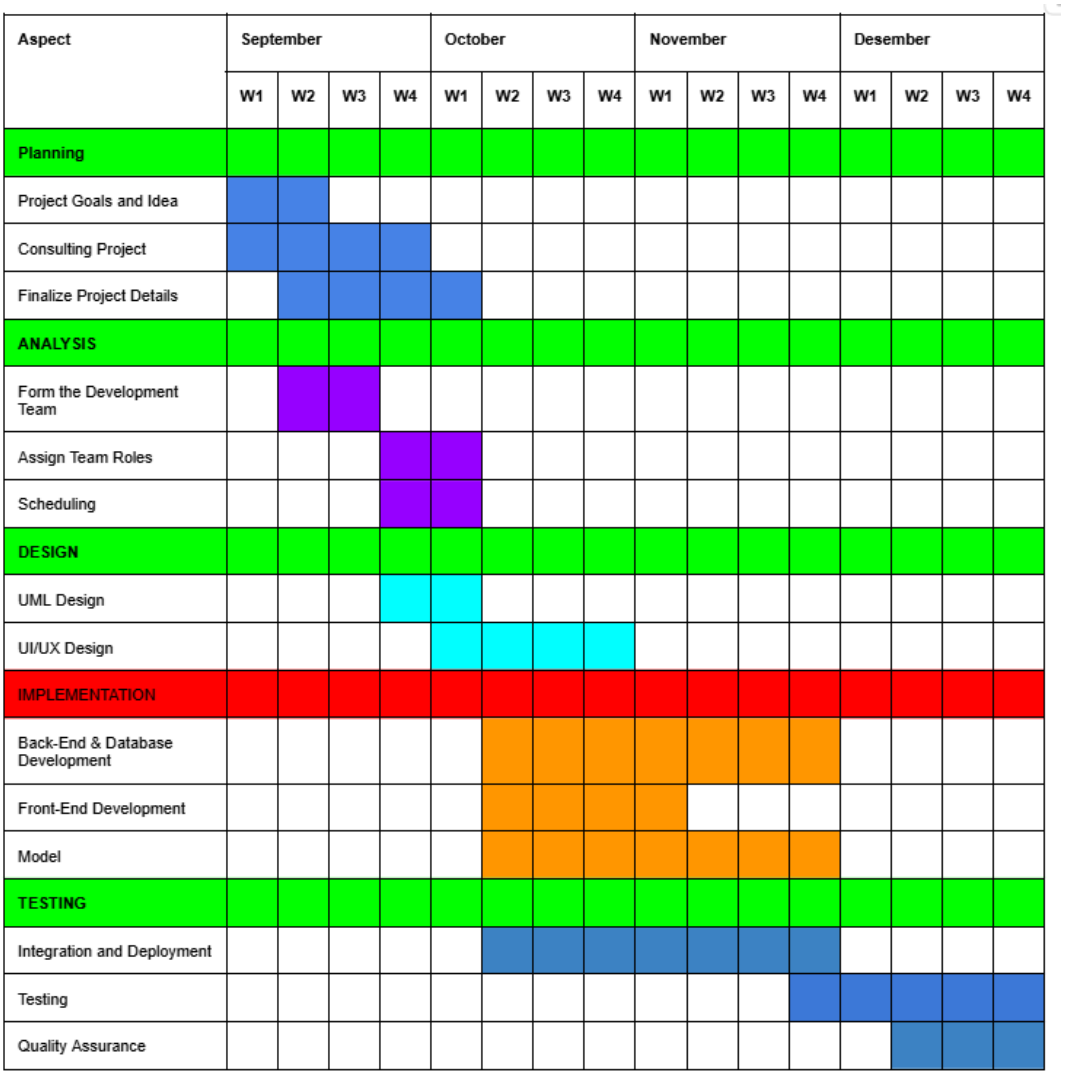
- Creates and trains the AI Models needed using scikit-learn.
- Develop a REST API using express.js and nodejs to connect between client, server, and database.
- Manages real-time synchronization and reliability.
- Creates functions to do automatic maintenance and update.

2.1.3.2 Database Management

- Manages how the training dataset is stored.
- Creates functions to query the database.
- Creates a system when storing so it is accessible from anywhere.

2.2. SCHEDULE AND TIMELINE

2.2.1 GENERAL TIMELINE



2.2.2 TIMELINE DETAILS AND SOFTWARE DEVELOPMENT CYCLE

The development methodology used in this project is the Waterfall Model (Sequential Model), in which each stage of development is carried out sequentially and systematically. The following are details of each phase to be implemented in the PROP-AI project.

- **Planning**

At this early stage, the main focus is to analyze project requirements and define the scope of the Minimum Viable Product (MVP). The team will identify key features such as a property search system, user account management, and a price prediction model. In addition, the technology stack will be selected using Node.js for the back-end, React.js for the front-end, and Python (Scikit-learn & Pandas) for the AI model.

- **Design & Prototyping**

  This stage focuses on system architecture design. The team will create UML diagrams to illustrate data flow and interactions between components. On the interface side, UI/UX design will be carried out to ensure an intuitive and responsive user experience before the development process begins.

- **Development**

  This stage is the core of the project, where the designs that have been created are converted into functional code.

  - Back-End Developer: implements APIs, user authentication systems, and database connections.
  - Front-End Developer: builds dynamic web interfaces using React.js
  - AI Developer: Develop and train price prediction models using property datasets and the Scikit-learn library.

- **Integration**

  At this stage, all system components (front-end, back-end, and AI model) will be integrated. The goal is to ensure smooth data communication between the user interface, server, and artificial intelligence module.

- **Testing**

  The testing phase is conducted to ensure that the system works according to the specified requirements. Testing includes unit testing, integration testing, and user acceptance testing (UAT) to ensure the stability and accuracy of price predictions.

- **Deployment**

Once the system has been declared satisfactory, the application will be implemented in the production environment (cloud server) so that it can be accessed by the public.

- **Maintenance**

The final stage focuses on monitoring system performance after launch. The team will handle bugs after release, perform regular updates, and add additional features based on user feedback.

2.2.3 TIMELINE EVALUATION

Timeline and project evaluation is a very important process to ensure that projects remain within the specified scope and schedule.To support this, our team will implement three structured evaluation methods on a regular basis, as follows

- **Weekly Progress Meetings**

  Every Wednesday, the team will hold a brief meeting to report on the progress of the project, discuss any obstacles or barriers that have arisen, and plan the tasks to be carried out in the following week. The purpose of this meeting is to maintain communication between team members and ensure that each part of the project is running according to plan.

- **Task Management Board**

  The team will utilize project management tools such as Trello to monitor the status of each task, where all team members will be able to see the progress of the project, including tasks that are in progress, completed, or pending.

- **Milestone Checking**

  Once a part of a project is completed, the team will conduct a review to ensure that all targets and results for that part have been achieved before proceeding to the next stage. This step helps to ensure the

quality of the results at each stage and prevent errors from impacting the next stage.

2.3. PROJECT PERFORMANCE INDICATOR

To ensure that PROP-AI is working properly and maximize user satisfaction, we have developed indicators and metrics for us to reach. These indicators are divided into two primary categories, System Performance Metrics, which measures the quality of the software, and Project Management Performance Metrics, which measures the effectiveness of the development process.

- System Performance Metrics
  The metrics measured are;
  - System Uptime, our target is at least 90% uptime during operational hours.
  - Request-to-Response time, our target is less than 15 seconds
  - AI Model Accuracy, our target is between 80% and 99%
- Project Management Metrics
  - Milestone Completion Rate, Target: 100% of project milestones completed within the planned timeline
  - User Satisfaction, Target: ≥ 80% satisfaction rate from both user groups.

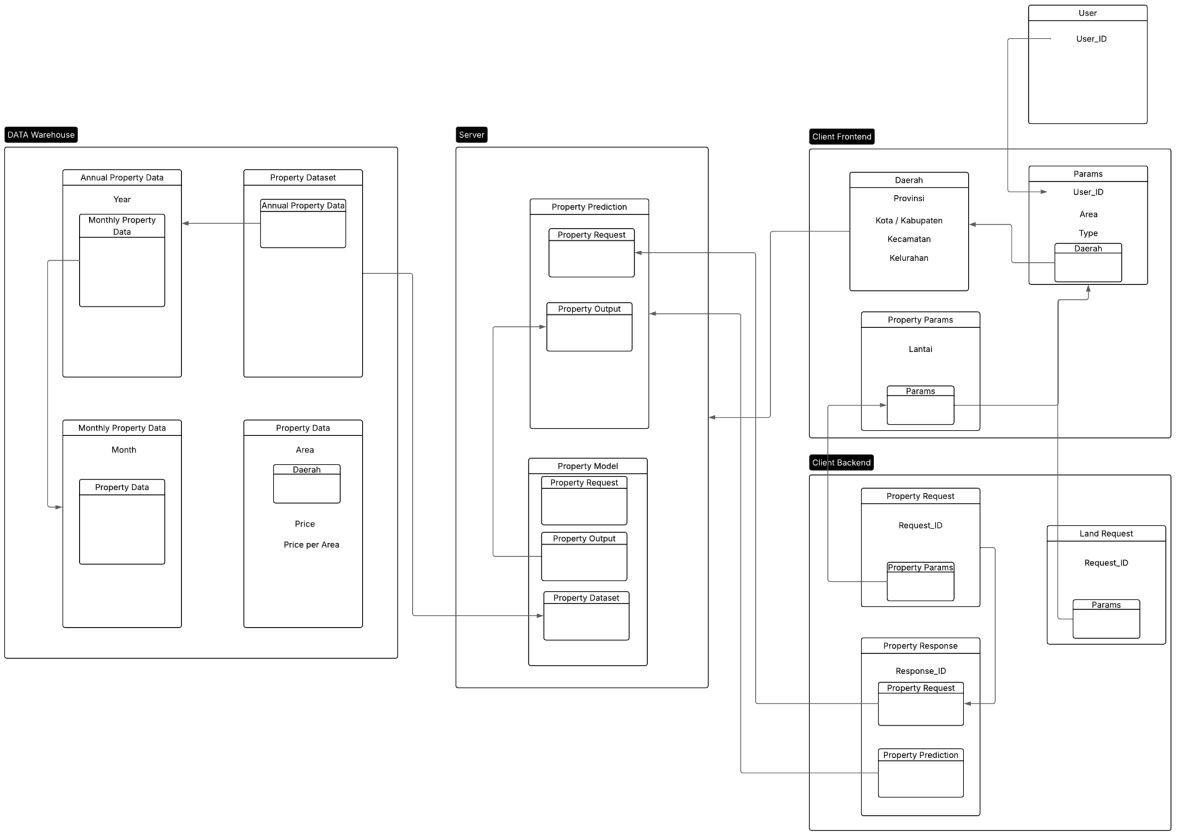
Once the targets are reached, it is used as indicators of success rate of this projects from the technical and management sides.

CHAPTER III

UML DIAGRAMS AND DESIGN

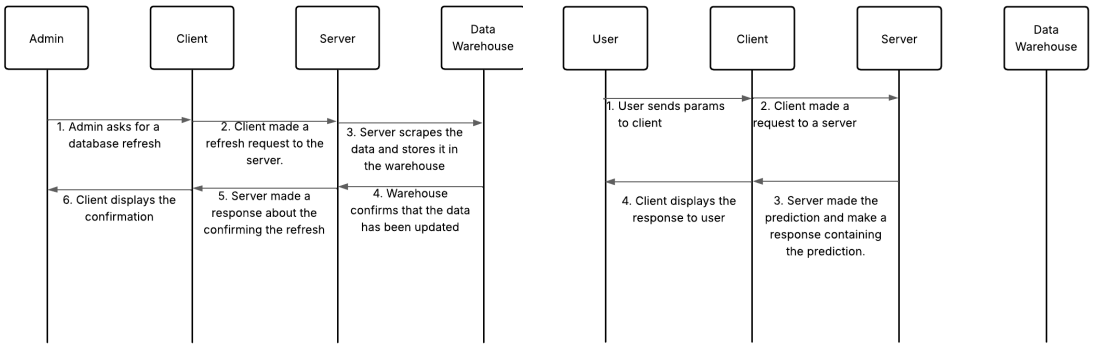
3.1. UNIFIED MODELING LANGUAGE

3.1.1 Class Diagram



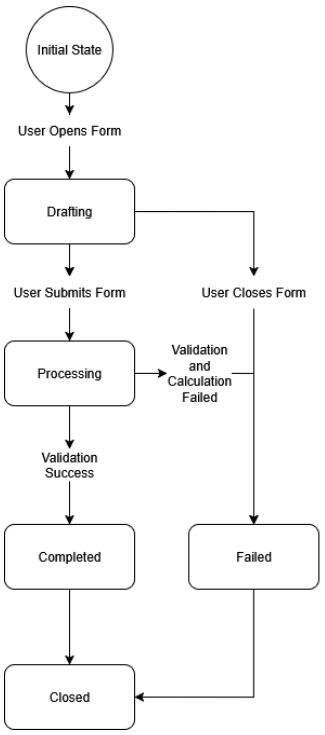
This diagram illustrates the architecture of a property price prediction system where a user interacts with the Client Frontend to input property details like location and area. This information is passed to the Client Backend, which formats it into a request and sends it to the Server. The Server, containing the core machine learning Property Model, processes this request to generate a price prediction. The model's accuracy is derived from a comprehensive Property Dataset which is built from historical data stored in the Data Warehouse. Finally, the server sends the prediction result back through the Client Backend to be displayed to the user on the frontend interface.

3.1.2 Sequence Diagram



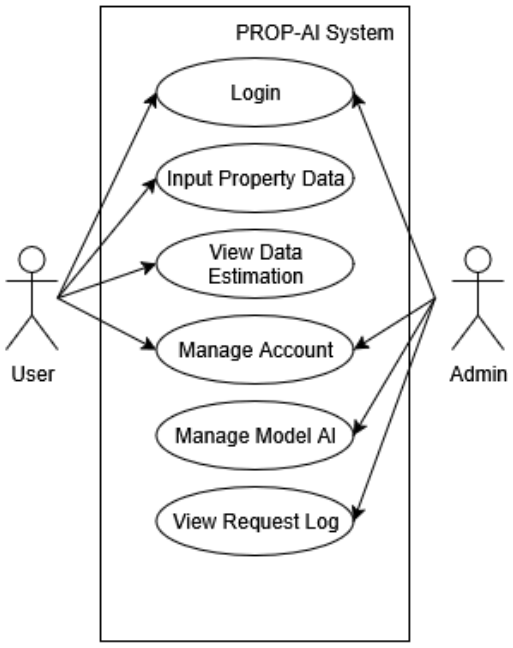
The first diagram shows the user prediction process, where a User sends parameters to the Client, which in turn requests a prediction from the Server. The Server then processes the request, generates a prediction, and returns it to the Client to be displayed to the User. The second diagram details the data refresh workflow, which is initiated by an Admin through the Client. The Client sends this refresh request to the Server, which proceeds to scrape new data and store it in the Data Warehouse. After the warehouse confirms the update, the Server sends a success message back through the Client to notify the Admin that the process is complete.

3.1.3 State Diagram



This diagram illustrates the various states of a form submission process. The process begins at the Initial State when a user opens the form, which transitions to the Drafting state. From here, the user can either submit the form (User Submits Form), which moves the state to Processing, or close the form (User Closes Form), which leads directly to the Failed state. During the Processing state, the system performs validation and calculation. A Validation Success transitions the state to Completed, whereas a Validation and Calculation Failed event moves it to the Failed state. Ultimately, from both the Completed and Failed states, the process concludes in the final Closed state.

3.1.4 Use Case Diagram



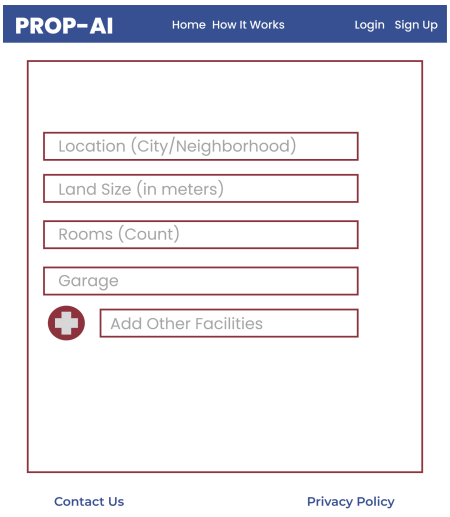
The diagram on the right is a Use Case Diagram that describes the functionality of the "PROP-AI" system and the interactions between the system and its users (actors). There are two actors: a User and an Admin. A User can perform actions such as Login, Input Property Data, View Data Estimation, and Manage Account. The Admin, on the other hand, has more extensive privileges. In addition to being able to Login, an Admin can also Manage Model AI and View Request Log, which are key functions for system maintenance and monitoring.

3.2 Design Implementation

3.2.1 Color Pallete

3.2.2 Logo

3.2.3 User Interface Plan



CHAPTER IV

PERJANJIAN KERJA SAMA PROP-AI

Kami yang bertanda tangan di bawah ini :

Nama : - Muhammad Raihan Mustofa
- Ganendra Garda Pratama
- Raka Arrayan Muttaqien

Selanjutnya disebut sebagai **pihak 1 (pihak pertama)**

Nama : Amalia Syahadatin

Selanjutnya disebut sebagai **pihak 2 (pihak kedua)**

Dengan ini menerangkan bahwa semua pihak setuju dan sepakat untuk mengadakan perjanjian kerjasama dengan memakai ketentuan ketentuan dan syarat-syarat sebagai berikut :

LINGKUP KERJASAMA

Pasal 1

Semua pihak telah sepakat dan setuju untuk mengadakan suatu perjanjian kerjasama dalam pelaksanaan pekerjaan pengadaan sebuah perangkat lunak sebagai berikut :

a. Perangkat lunak yang dikerjakan adalah Perancangan dan Pengembangan Platform PROP-AI (Platform Jual Beli Properti berbasis Web dengan Fitur Rekomendasi Harga Cerdas).
b. Pengguna perangkat lunak ini adalah **PIHAK KEDUA**.
c. Detail scope atau fitur akan selalu didiskusikan dan disepakati bersama (**PIHAK PERTAMA** dan **PIHAK KEDUA**)
d. **PIHAK KEDUA** berhak untuk Hak Atas Kekayaan Intelektual penuh atas perangkat lunak yang dikerjakan.

TUJUAN

Pasal 2

Tujuan perjanjian kerjasama ini adalah bahwa PIHAK PERTAMA melaksanakan dan menyelesaikan pekerjaan perangkat lunak PROP-AI dengan cakupan tujuan berikut

a. Merancang dan mengimplementasikan platform jual beli properti berbasis web yang intuitif, responsif, dan mudah digunakan.
b. Mengembangkan model prediksi harga akurat menggunakan machine learning.
c. Menyediakan rekomendasi harga optimal kepada penjual agar kompetitif dan realistis.
d. Meningkatkan transparansi pasar properti melalui data perbandingan harga yang objektif.

PENDANAAN

Pasal 3

Kedua belah pihak akan bertanggung jawab dari segi pendanaan yang diatur pada beberapa aturan di bawah ini :

a. Semua pengeluaran dana untuk keperluan pengadaan dan pengerjaan dilakukan melalui pengajuan dari **PIHAK PERTAMA** dan harus disetujui oleh **PIHAK KEDUA** yang akan dituangkan dalam anggaran biaya dengan waktu yang akan ditentukan sesuai dengan kesepakatan kedua belah pihak.

HAK DAN KEWAJIBAN

Pasal 4

Kedua belah pihak akan melaksanakan tugas dan tanggung jawabnya berkaitan dengan pelaksanaan pengadaan perangkat lunak, sebagai berikut :

4.1. **PIHAK PERTAMA** berkewajiban untuk :

a. Melaksanakan seluruh jadwal pekerjaan (event) sesuai dengan Sprint Goal yang telah ditentukan.
b. Mentaati semua ketentuan pengerjaan yang diberikan oleh PIHAK KEDUA.
c. Menjaga Rahasia dan Hak Intelektual dari perangkat lunak PROP-AI.

4.2. **PIHAK KEDUA** berkewajiban untuk :

a. Menyiapkan dan Meminjamkan :
    1. Dokumentasi kebutuhan bisnis dan analisis properti.
    2. Dataset properti (lokasi, harga, luas bangunan, klasifikasi ruang, fasilitas, dll.) untuk kebutuhan pengembangan model machine learning.
b. Mengawasi dan menyetujui seluruh proses pengerjaan apabila sudah sesuai dengan ketentuan yang telah disepakati bersama.

JANGKA WAKTU

Pasal 5

Kedua belah pihak sepakat bahwa pengerjaan akan dilaksanakan pada [26 September 2025] dan selesai pada [11 Januari 2026].

PENGALIHAN PERJANJIAN

Pasal 6

Kedua belah pihak sepakat untuk tidak mengalihkan perjanjian kerjasama ini kepada pihak manapun, kecuali dengan persetujuan tertulis dari kedua belah pihak.

KETENTUAN PERJANJIAN

Pasal 7

a. Menggunakan Scrum sebagai manajemen kerangka kerja. Dan menjadikan Scrum Guide (www.scrumguides.org) menjadi acuan dan aturan utama yang disepakati kedua belah pihak.
b. Perwakilan PIHAK KEDUA bertindak mengisi role Product Owner
c. Perwakilan PIHAK PERTAMA bertindak mengisi role Scrum Master
d. PIHAK PERTAMA dan PIHAK KEDUA bertindak mengisi role Development Team.
e. Change for Free (PIHAK KEDUA sebagai Product Owner bebas untuk menentukan perubahan dalam suatu sprint)
f. Product Owner bebas menentukan Scope atau Sprint Goal tiap Sprint Planning.

TAMBAHAN PERJANJIAN

Pasal 8

Apabila ada beberapa pasal tambahan setelah ditandatangani perjanjian ini, maka perjanjian tambahan akan diberlakukan sebagai adendum (perjanjian tambahan) setelah disepakati oleh masing-masing pihak terkait.

PENYELESAIAN PERSELISIHAN

Pasal 9

Apabila terjadi perselisihan antara kedua belah pihak, maka kedua belah pihak sepakat untuk menyelesaikan secara musyawarah untuk mufakat. Apabila tidak tercapai penyelesaian secara musyawarah untuk mufakat, masing-masing pihak sepakat untuk menempuh jalur hukum yang berlaku.

FORCE MAJURE

Pasal 10

Apabila terjadi gempa, bencana alam, tanah longsor atau semua kejadian yang disebabkan oleh alam yang akan menyebabkan gagalnya / tertundanya perjanjian ini maka kedua belah pihak sepakat untuk meninjau kembali perjanjian yang telah dibuat.

PENUTUP

Pasal 11

Surat Perjanjian Kerjasama ini telah dibaca, dimengerti dan disetujui oleh kedua belah pihak pada hari ini dan tanggal tersebut pada surat perjanjian kerjasama ini, dibuat sebanyak rangkap 2 (dua) dan dibubuhi tanda-tangan sebagai tanda kesepakatan bersama dan mempunyai kekuatan hukum yang sama.

Pihak Pertama,	Pihak Kedua,
	
Muhammad Raihan Mustofa	Amalia Syahadatin

CHAPTER V

TEST PLAN & TESTING DOCUMENTATION

5.1 BACKGROUND

The purpose of this Test Plan is to define the testing approach for the PROP-AI backend API services and client application. Testing will be performed using :

- Manual testing
- Automated API testing using Postman
- Functional and non-functional testing
- User Acceptance Testing (UAT)

5.2 TEST OBJECTIVES

5.2.1 Verify Core System Functionality

- Validate authentication endpoints
- Validate property listing module
- Validate prediction module
- Validate user management module

5.2.2 Ensure API Stability

- Validate correctness of request and response formats using Postman
- Ensure all API endpoints return correct status codes

5.2.3 Ensure Frontend & Backend Integration

- Validate communication between frontend client app and backend API
- Validate machine learning model can be used properly

5.2.4 Validate Non-functional Requirements

- Security and token validation
- Performance for key endpoints
- Error handling and input validation

5.3 TEST SCOPE

This Test Plan outlines the scope, objectives, resources, test items, test design, and pass/fail criteria.

Included

- Backend REST API testing via Postman
- Integration: Frontend consuming API
- Authentication, Property Listings, Predictions
- Negative testing (invalid data, wrong token)

Excluded

- Load testing > 1000 requests/sec
- Penetration testing

5.4 TESTING METHODOLOGY

5.4.1 Manual API Testing (Postman)

Postman is used to verify

- Request body validation
- Response structure
- Status codes (200, 400, 401, 404, 409, etc.)
- Multipart upload (property images, user avatar upload)

5.4.2 Black-Box Testing

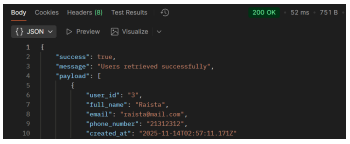
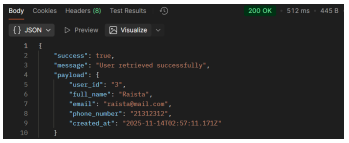
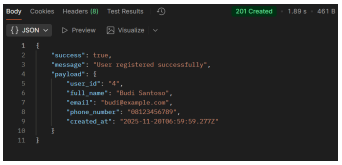
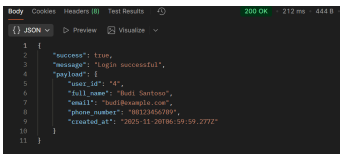
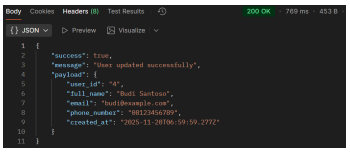
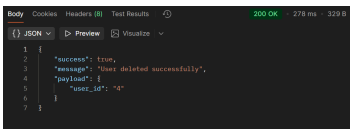
Focus only on inputs and outputs without looking at internal code.

5.4.3 Exploratory Testing

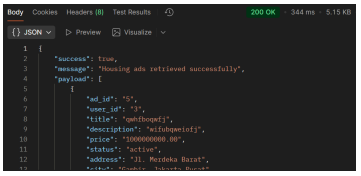
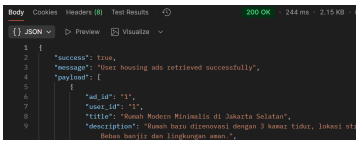
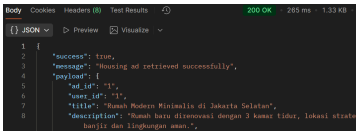
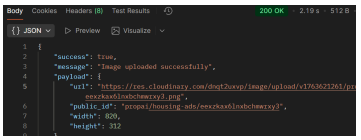
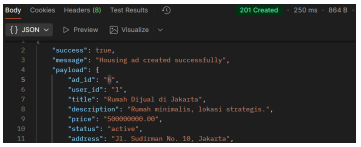
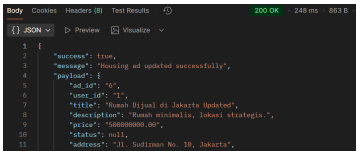
Unscripted testing to identify unexpected behavior.

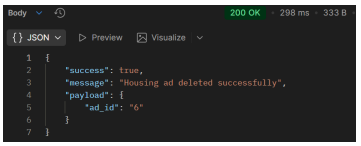
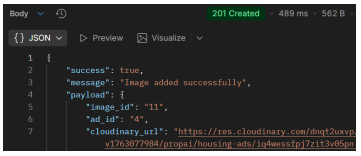
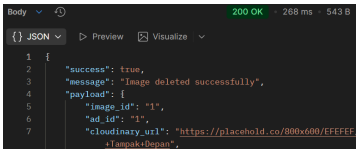
5.5 Testing Documentation

5.5.1 User Documentation

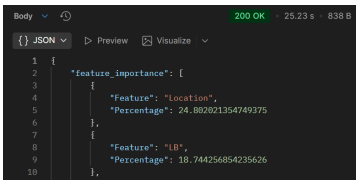
Test ID	Endpoint	Scenario	Prereq uisites	Input	Expected Output	Actual Result
USER-000	GET /api/users	Get all users	-	-	Status 200 OK, returns users object	
USER-001	GET /api/users/:id	Get user by ID	-	Path Variable : id	Status 200 OK, returns a user object	
USER-002	POST /api/users/register	Register new user	-	{ "full_name": "Budi Santoso", "email": "budi@example.com", "password": "yourpassword123", "phone_number": "08123456789" }	Status 201 Created, returns user object that get created	
USER-003	POST /api/users/login	Login user	The account is already exist	{ "email": "budi@example.com", "password": "yourpassword123" }	Status 200 OK, returns user object that get logged in	
USER-004	PUT /api/users/:id	Update user	The account is already exist	{ "full_name": "Budi Santoso", "phone_number": "08123456789" }	Status 200 OK, returns a user object	
USER-005	DELETE /api/users/:id	Delete user	The account is already exist	Path Variable : id	Status 200 OK, returns payload user_id that gets deleted	

5.5.2 Housing Ad Documentation

Test ID	Endpoint	Scenario	Prerequisites	Input	Expected Output	Actual Result
ADS-000	GET /api/housing-ads	Get all housing ads	-	-	Status 200 OK, returns housing ads object	
ADS-001	GET /api/housing-ads/user/:userId	Get ads by user ID	-	Path Variable : id	Status 200 OK, returns ads from user_id	
ADS-002	GET /api/housing-ads/:id	Get housing ad by ID	-	Path Variable : id	Status 200 OK, returns ads from its id	
ADS-003	POST /api/housing-ads/upload-image	Upload image to Cloudinary	-	req.file : image	Status 200 OK, returns url of the image	
ADS-004	POST /api/housing-ads	Create new housing ad	-	{ "user_id": 1, "title": "Rumah Dijual di Jakarta", "price": 500000000, "address": "Jl. Sudirman No. 10, Jakarta", "city": "Jakarta", "province": "DKI Jakarta", "postal_code": "10220", "land_size_sqm": 120, "building_size_sqm": 100, "bedrooms": 3, "bathrooms": 2, "garage_capacity": 1, "contact_phone": "08123456789" }	Status 201 Created, returns housing ads object	
ADS-005	PUT /api/housing-ads/:id	Update housing ad	The housing ad is already exist	Path Variable : id { "title": "Rumah Dijual di Jakarta Updated", "price": 500000000, "address": "Jl. Sudirman No. 10, Jakarta", "city": "Jakarta", "province": "DKI Jakarta", "postal_code": "10220", "latitude": -6.21462, "longitude": 106.84513, "land_size_sqm": 120, "building_size_sqm": 100, "bedrooms": 3, "bathrooms": 2,	Status 200 OK, return housing ads object	

				"garage_capacity": 1, "contact_phone": "08123456789" }		
ADS-006	DELETE /api/housing-ads/:id	Delete housing ad	The housing ad is already exist	Path Variable : id	Status 200 OK, return ad_id that gets deleted	
ADS-007	POST /api/housing-ads/:id/images	Add image to housing ad	The housing ad is already exist	Path Variable : id { "cloudinary_url" : "https://res.cloudinary.com/dnqt2uxvp/image/upload/v1763077984/propai/housing-ads/iq4wessfpj7zit3v05pn.jpg" }	Status 201 Created, returns image and its ad_id.	
ADS-008	DELETE /api/housing-ads/images/:imageId	Delete image	The housing ad image is already exist	Path Variable : id	Status 200 OK, returns image and its ad_id.	

5.5.3 Machine Learning Documentation

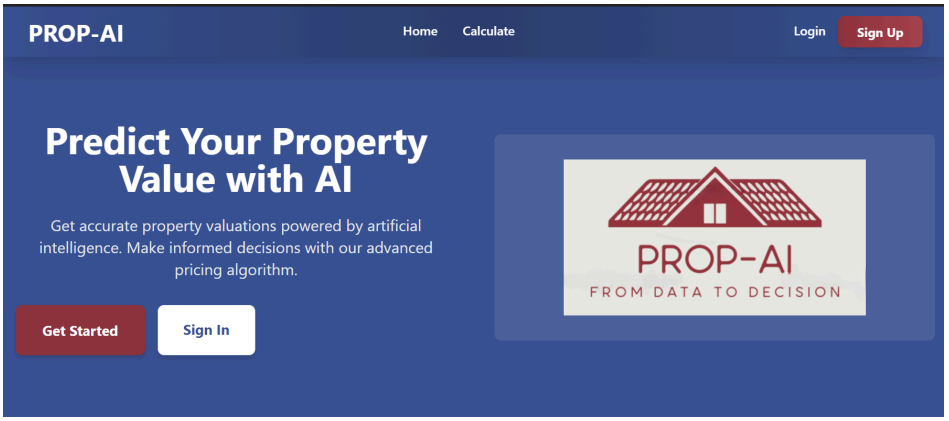
Test ID	**Endpoint**	**Scenario**	**Prerequisites**	**Input**	**Expected Output**	**Actual Result**
MLM-000	POST /predict	Predict house prices	-	{ "location": "Beji, Depok", "bedrooms": 3, "toilet": 2, "garage": 1, "LT": 100, "LB": 120 }	Status 200 OK returns predicted price and features prices	

CHAPTER VI

USER MANUAL

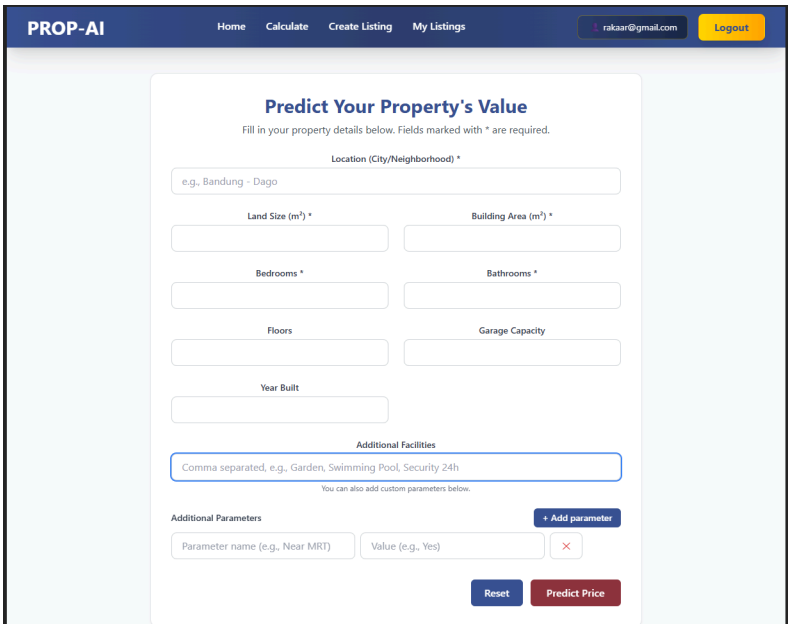
6.1 Create new Account and Login

Users are required to register or sign in to access personalized features.



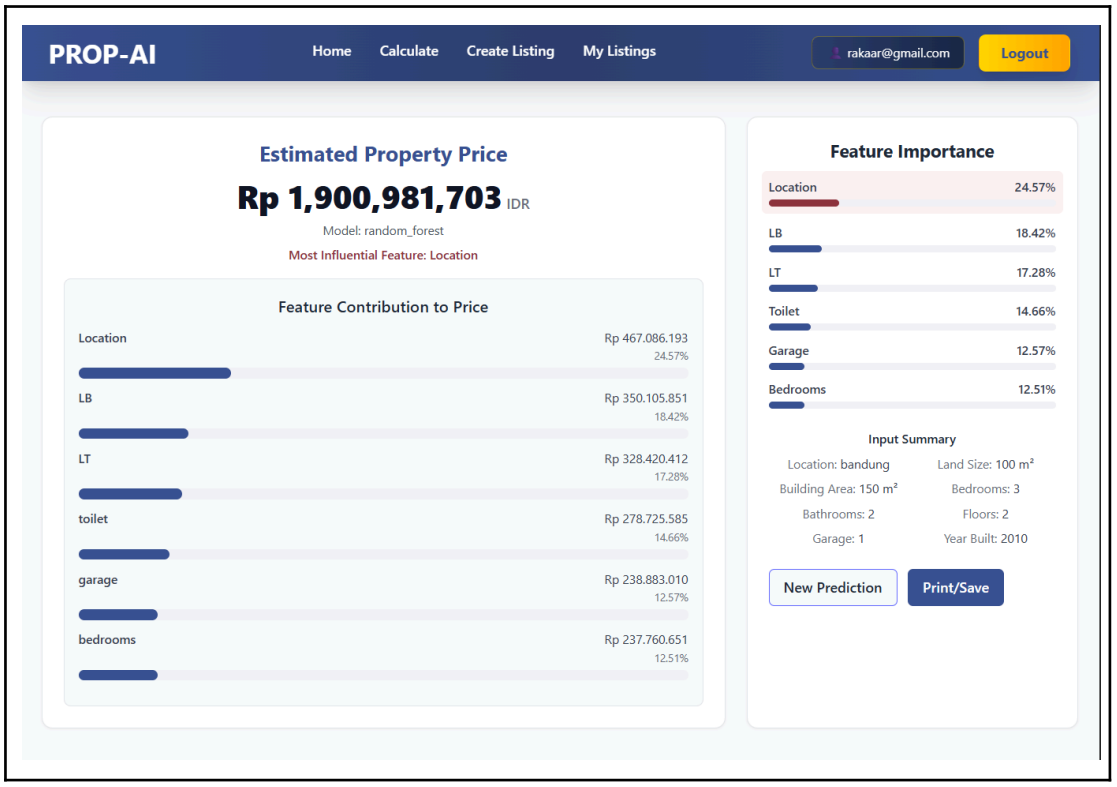
6.2 Calculate

Price Prediction allows users to input detailed information about a property to instantly receive an AI-powered valuation.



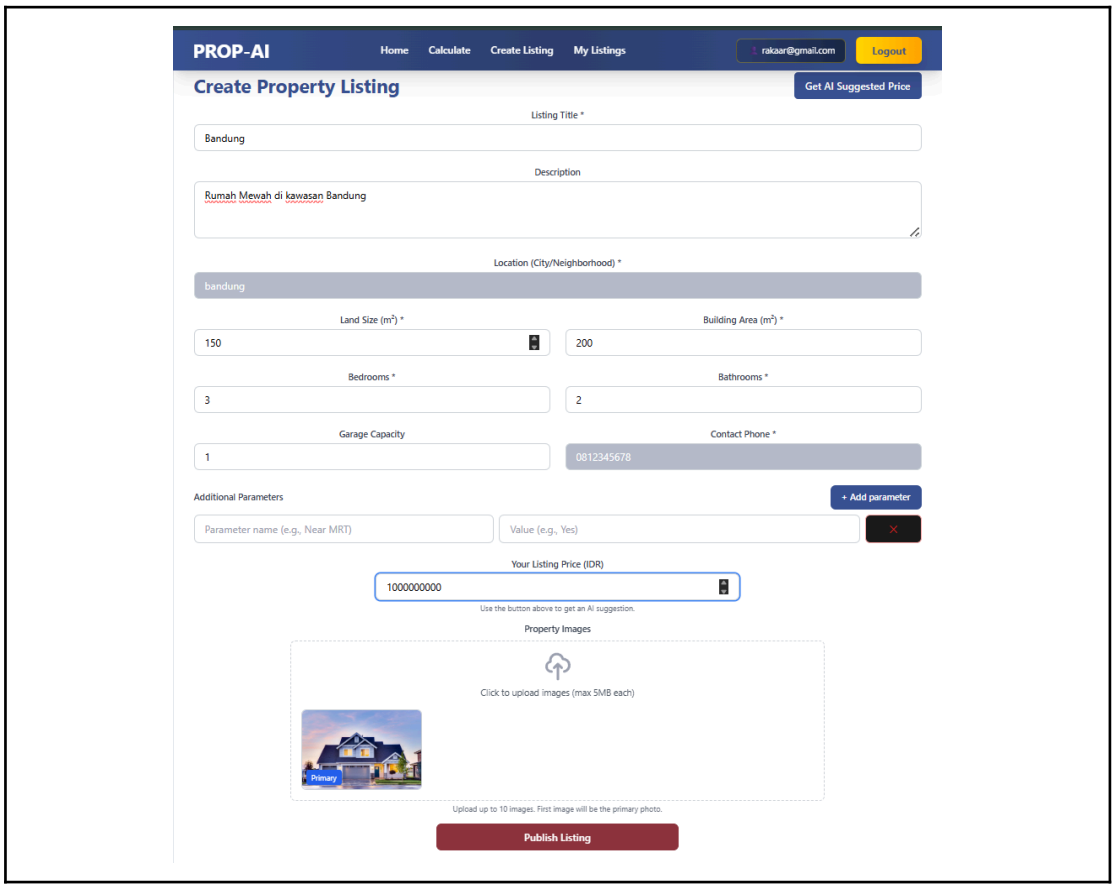
6.3 Estimated Property Price

The Estimated Property Price feature provides an AI-powered valuation that helps users understand the potential market value of their property with greater accuracy. In addition to showing the predicted price, the system also displays detailed feature importance and contribution charts, allowing users to see which factors such as location, building size, or number of rooms most strongly influence the valuation. Users can review a complete summary of their input data and, if needed, print or save the generated report for documentation



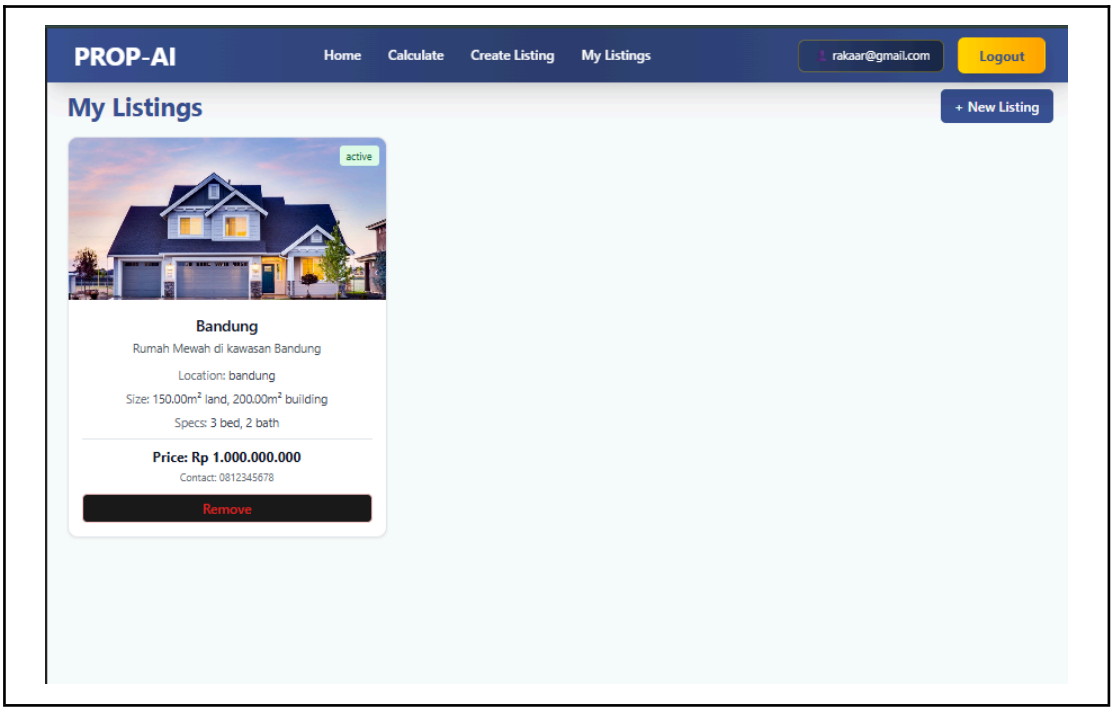
6.4 Create Property Listing

Create Property Listing allows users to independently create, publish, and manage their own property advertisements. Through this feature, users can enter complete property details and upload photos to showcase their listings effectively.



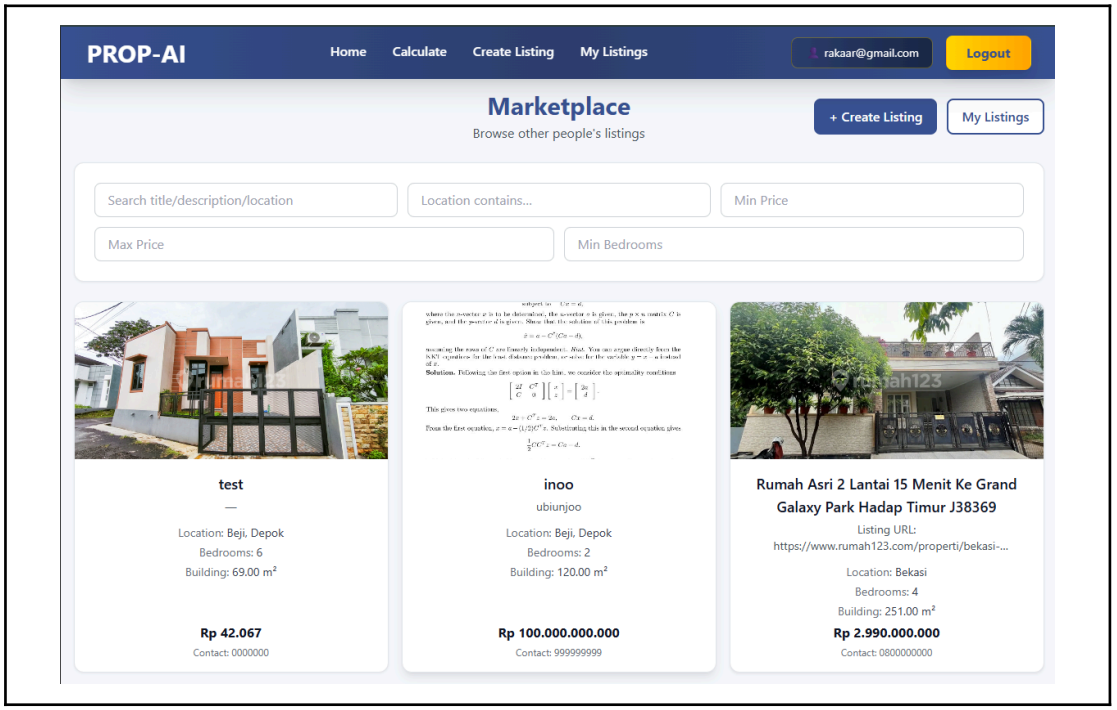
6.5 My Listing

My Listing provides users with a personalized space to view and manage only the properties they have created, complete with detailed information and posting history. Unlike the Marketplace, which displays listings from all users, My Listing focuses solely on the owner's advertisements, making it easier to update, monitor, and track each property's performance. Once a user posts a property through My Listing, it will automatically appear in the Marketplace so that all visitors of the platform can see the advertisement and potential buyers can discover the property.



6.6 Marketplace

Marketplace offers a comprehensive market overview that allows users to browse current property listings sourced from various websites. This feature provides broader market visibility, helping users compare prices, analyze trends, and gain valuable insights into available properties across different platforms.



REFERENCES

[1]F. Maloku and B. Maloku, “House Price Prediction Using Machine Learning and Artificial Intelligence,” Journal of Artificial Intelligence & Cloud Computing, vol. 3, no. 4, pp. 1–10, Aug. 2024, doi: https://doi.org/10.47363/jaicc/2024(3)357.

[2]S. Oh, S. P. Hang, and J. Thye, “Prediction of residential property prices using machine learning algorithms,” ITM Web of Conferences, vol. 01042, no. 67, pp. 01042–01042, Jan. 2024, doi: https://doi.org/10.1051/itmconf/20246701042.

[3]J. Yu, “Predicting New York Housing Prices: A Comparative Analysis of Machine Learning Models,” Proceedings of the 1st International Conference on Innovations in Applied Mathematics, Physics and Astronomy, vol. 1, no. 1, pp. 102–109, 2024, doi: https://doi.org/10.5220/0012999000004601.