Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th June 2025. Vol. 4. No. 3; e-ISSN: 2808-4519

Prediction of Red Chili Prices at Musi Market Using Android-Based Linear Regression Algorithm

Muhammad Syaiful Iskandar[1*], Rita Wahyuni Arifin[2]

[1]*Informatics engineering, Universitas Bina Insani, Indonesia.*
[2]*Informatics management ,Universitas Bina Insani, Indonesia.*
msyaifuliskandar@gmail.com[1*], ritawahyuni@binainsani.ac.id[2]

Abstract

The prices of nine essential goods (sembako) often experience fluctuations, which can affect people's purchasing power and economic stability. One commodity that frequently undergoes price changes is red bird's eye chili. One of the factors causing the instability of red chili pepper prices is extreme weather changes, such as heavy rainfall or droughts, which directly impact harvest yields and the availability of supplies in the market. This research aims to predict the price of red bird's eye chili in Musi Market using the Linear Regression algorithm. Furthermore, the study also develops an Android-based application to provide users with real-time and predictive price information for red bird's eye chili. This predictive information will be displayed through the Android-based application, making it easily accessible to users and helping them obtain price data quickly and accurately. This system integrates price data, weather, and seasonal events to predict price fluctuations. Evaluation results show that Linear Regression is the best model, with an MAE of 14,380.53 and an R^2 of 0.686, indicating the model's ability to explain 68.6% of the data variation, providing an efficient solution for price monitoring at Pasar Musi.

Keywords: *Android, Linear Regression, Prediction, Red Chili.*

1. Introduction

Bird's Eye Chili (Capsicum frutescens) is a high-value horticultural commodity in Indonesia [1]. Chili is a highly perishable and seasonal agricultural product, and farmers using the same cultivation technology can typically only produce it once per season. During harvest peaks, it is advisable to plant large quantities of chili. This creates a dilemma where chili prices plummet and the produce spoils easily if mishandled. One major issue with red chili production is that farmers are constantly worried about price fluctuations. When chili production surges at certain times, the market price often drops. This is due to abundant supply while demand tends to remain stable in the short term [2]. Demand for red bird's eye chili continues to increase each year, in line with population growth and the expansion of industrial sectors that use red bird's eye chili as a raw material [3].

Based on observations at Pasar Musi, located at Jl. Ps. Musi No.14, Abadijaya, Sukmajaya Subdistrict, Depok City, West Java, which is one of the traditional markets playing a significant role in staple food distribution, this study utilizes two types of data: primary and secondary data, to achieve comprehensive and accurate results. Primary data will be obtained through direct observation at the research location, including observation of trading activities, interactions between sellers and buyers, and price variations of commodities. Meanwhile, secondary data will be collected from reliable sources such as the official website https://dataonline.bmkg.go.id/data-harian for weather data, https://infopangan.jakarta.go.id/commodity for red bird's eye chili prices, and https://hargajateng.org/tabel-harga-kab for red bird's eye chili prices, which provide up-to-date and relevant national food price information.

To address the price fluctuations of red bird's eye chili, the application of machine learning technology using the linear regression algorithm becomes a promising innovative solution [4]. This algorithm enables the development of a price prediction model based on historical data and factors that influence price, such as weather conditions, harvest volume, market demand, and distribution costs [5]. Through linear regression analysis, the relationship between independent variables (such as rainfall, air temperature, and stock availability) and the dependent variable (red chili price) can be structurally identified to produce accurate predictions [6].

The regression method is used to model data and predict numerically continuous price values by understanding the characteristics or features of unknown objects through patterns in the dataset. The prediction process in machine learning involves training using training data, such as date, commodity, market, and price variables [7]. The resulting model from this training is then used to predict daily chili prices. A data mining approach is also applied in this study, where mining is performed on a data set to produce useful new information, namely daily chili price predictions [8]. Its implementation uses the Python programming language as the backend for chili price prediction [9].

This study aims to develop a red bird's eye chili price prediction model at Pasar Musi using a linear regression algorithm based on machine learning. This model is expected to provide accurate daily price predictions by utilizing historical data and relevant variables such as weather conditions, harvest volume, and other market factors. The prediction information will be displayed through an Android-based application, allowing easy access for users [10].

2. Research Methodology

2.1. Rapid Application Development (RAD)

The system development method used is Rapid Application Development (RAD). RAD is an adaptive software development method based on models or prototypes. This method is iterative, relying on feedback in each iteration. The RAD approach emphasizes the system development process through models or prototypes rather than detailed and extensive planning [11]. The RAD system development approach is illustrated in the figure below.

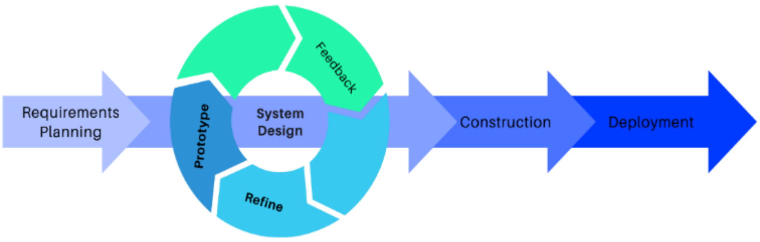

Fig. 1: Rapid Application Deveopment (RAD)

The stages of the RAD method are explained as follows:

1. Requirements Planning:
   Identification of system requirements based on interviews, consisting of:
   Main features: Red bird's eye chili price prediction and selling feature.
   Supporting features: Vegetable price list and vegetable commodity information articles.
2. System Design:
   Three design stages:
   Prototype: UI development using Flutter based on feature requirements.
   Feedback: Gathering input from users.
   Refine: Refining the application based on the feedback.
3. Construction:
   Intensive prototype development using:
   Technology: Flutter and Dart.
   Main features: Price prediction (using linear regression) and selling feature.
   Supporting features: Vegetable price list and informational articles.
   Testing: Using the black box method.
4. Deployment:
   The application is ready for use by users after all development stages are completed.

2.2. Cross Industry Standard Process for Data Mining (CRISP-DM)

CRISP-DM was chosen because it is a proven, flexible, and structured process model for data mining and data science projects. This model provides a clear framework from business understanding to deploying the model in a production environment. It is capable of being integrated with agile approaches and focuses on the implementation of analytical results. Below is an illustration of the CRISP-DM process along with its explanation.

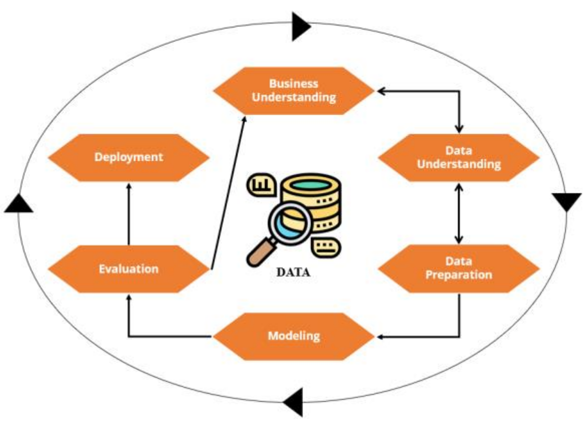

Fig. 2: CRISP-DM Method

The stages of the CRISP-DM method are explained as follows:

1. Business Understanding:
   Identify the problem and business objectives.
   Main objective: To predict the price of red bird's eye chili at Pasar Musi.
2. Data Understanding:
   Collect and explore primary and secondary data related to chili prices to understand the context of the problem.
3. Data Preparation:
   Clean and prepare the data for analysis, including:
   Handling missing data
   Format transformation
   Data processing using Python
4. Modeling:
   Develop a predictive model using training data to estimate the price of red bird's eye chili.
5. Evaluation:
   Assess the model's performance using metrics such as accuracy to ensure it meets the initial objectives.
6. Deployment:
   The model is implemented into the system to automatically predict chili prices at Pasar Musi.

3. Result and Discussion

Currently, the fluctuation of red bird's eye chili prices at Pasar Musi still relies on manual recording conducted by vendors and market managers, which means that price updates are not done in real time. This update process often experiences delays, as chili prices can change daily, but the available information is only updated during market operating hours [12]. This results in limited access to fast and accurate information. In addition, the methods used to predict red chili prices are still manual, lacking a technology-based analytical system that can estimate price movements more precisely. The absence of a price prediction system using analytical algorithms further limits the accuracy and speed of price information accessible to vendors and consumers. Therefore, the development of an Android-based application that can provide real-time price information, equipped with a price prediction feature using a linear regression algorithm, can greatly improve the efficiency and accuracy of red bird's eye chili price data [13]. This application will enable users to access faster and more accurate price information, as well as provide price predictions based on more accurate historical data [14].

3.1. Functional Requirements Analysis

Functional requirements analysis is conducted to identify the core functions that the system must have in order to operate according to its objectives. These requirements are related to the features and services provided by the system to ensure the developed application can be used properly and effectively. The following are the functional requirements of the system to be developed:

1. A home feature used to predict the price of red bird's eye chili.
2. A category feature that displays the prices and types of available vegetables.
3. A store feature that shows the available stores and allows users to contact them.
4. A news feature that displays the latest news related to economy, sports, national, international, and politics.

3.2. System Implementation Results

The red bird's eye chili price prediction application was developed to address delays and inaccuracies in price recording, which is still done manually at Pasar Musi. This Android-based application includes several core features designed to enhance the efficiency, accuracy, and speed of accessing and predicting chili price information [15].

Functional requirements analysis was conducted to identify the main functions that the system must have to operate effectively and meet its objectives. These requirements relate to the features and services provided by the system, ensuring that the application can be used properly and delivers value to its users. The implemented features of the system are as follows:

1. Home Feature: Enables price prediction for red bird's eye chili using a linear regression algorithm. This feature analyzes historical price data to forecast future price movements, allowing users to obtain price information more quickly and accurately.
2. Category Feature: Displays real-time prices and types of available vegetables in the market, providing users with comprehensive and up-to-date information.
3. Store Feature: Lists stores that sell red bird's eye chili and other vegetables, along with contact details such as phone numbers and store addresses, making it easier for users to make purchases or contact vendors directly.
4. News Feature: Presents the latest news related to the economy, sports, national, international, and politics, offering users additional insights into external factors that may influen.

3.3. Data Collection and Preprocessing

In this study, a total of 1,892 datasets were collected for chili price data and another 1,892 datasets for weather data, of which 30 datasets were obtained through direct observation at Pasar Musi. The remaining data was gathered from three sources that provide up-to-date information on red bird's eye chili prices and other commodities. First, data was sourced from https://hargajateng.org/tabel-harga-kab, which provides tables of agricultural commodity prices in the Central Java region, including regularly updated prices for red bird's eye chili. Second, price and commodity type information was obtained from https://infopangan.jakarta.go.id/commodity, a website that offers food price data from markets across DKI Jakarta, enabling monitoring of price movements in the capital region. Third, relevant meteorological data that influences red chili prices was gathered from https://dataonline.bmkg.go.id/data-harian, which provides daily weather data that plays a crucial role in agricultural yields and commodity price fluctuations.

The integration of these three data sources enables more in-depth and accurate analysis of the factors affecting red bird's eye chili prices in the market and provides a strong foundation for future price prediction [16]. After data collection, the next step is data preprocessing, which aims to reduce machine computation, speed up processing time, and improve the accuracy of model estimation. This involves data cleaning, including filling in missing values and correcting inconsistent data entries.

3.4. Algorithm Implementation

The implementation begins by loading two main datasets: chili price data and weather data, which are then processed and merged based on the date column to form a unified dataset. Subsequently, the weather and price data are combined and enriched with additional features, such as the previous month's price and indicators for special events—namely the fasting month and the Christmas–New Year period—which are expected to influence price fluctuations. Below is a screenshot of the code that has been created in Google Colab.

```
[ ]     # Fluktuasi harga selama event
        if row['puasa'] == 1:
            # Fluktuasi kenaikan harga 10% hingga 120% selama bulan puasa sampai lebaran
            price_fluctuation = np.random.uniform(1.1, 2.2) * (1 + price_adjustment + rain_adjustment)
        elif row['natal_tahun_baru'] == 1:
            # Fluktuasi kenaikan harga 10% hingga 60% selama Natal dan Tahun Baru
            price_fluctuation = np.random.uniform(1.1, 1.6) * (1 + price_adjustment + rain_adjustment)
        else:
            # Harga normal jika tidak ada event
            price_fluctuation = 1 * (1 + price_adjustment + rain_adjustment)

        # Menghitung harga yang disesuaikan berdasarkan semua faktor
        adjusted_price = row['Harga'] * price_fluctuation  # Harga akhir yang disesuaikan
        return adjusted_price

    # Menerapkan aturan pada dataset
    merged_df_v2["Harga_Adjusted"] = merged_df_v2.apply(apply_rules, axis=1)  # Terapkan fungsi untuk menyesuaikan harga

    # Menyiapkan data untuk training (fitur dan target)
    X_v2 = merged_df_v2[["harga_bulan_lalu", "Lamanya Penyinaran Matahari (jam)", "Curah Hujan (mm)", "puasa", "natal_tahun_baru"]]  # Fitur
    y_v2 = merged_df_v2["Harga_Adjusted"]  # Target (harga yang disesuaikan)

    # Membagi data menjadi training (70%) dan testing (30%)
    X_train_v2, X_test_v2, y_train_v2, y_test_v2 = train_test_split(X_v2, y_v2, test_size=0.3, random_state=72)  # Pembagian data

    # Mendefinisikan model-model yang akan digunakan untuk evaluasi
    models_v2 = {
        "Decision Tree": DecisionTreeRegressor(random_state=72),  # Model pohon keputusan
        "Random Forest": RandomForestRegressor(n_estimators=100, random_state=72),  # Model random forest dengan 100 pohon
        "Linear Regression": LinearRegression(),  # Model regresi linear
    }

    # Evaluasi model
    results_v2 = {}
```

Fig. 3: Code Implementation Part 1

The next step is the application of business rules based on domain knowledge. These rules associate chili prices with two main weather factors—sunlight duration and rainfall—which are logically assumed to influence prices either upward or downward depending on their intensity. In addition, seasonal price fluctuation scenarios are incorporated, based on the assumption that during festive months such as the fasting month and the Christmas–New Year period, chili prices tend to surge due to increased demand. After preprocessing and feature engineering are completed, the data is split into training data (70%) and testing data (30%).

```
[ ]     # Fluktuasi harga selama event
        if row['puasa'] == 1:
            # Fluktuasi kenaikan harga 10% hingga 120% selama bulan puasa sampai lebaran
            price_fluctuation = np.random.uniform(1.1, 2.2) * (1 + price_adjustment + rain_adjustment)
        elif row['natal_tahun_baru'] == 1:
            # Fluktuasi kenaikan harga 10% hingga 60% selama Natal dan Tahun Baru
            price_fluctuation = np.random.uniform(1.1, 1.6) * (1 + price_adjustment + rain_adjustment)
        else:
            # Harga normal jika tidak ada event
            price_fluctuation = 1 * (1 + price_adjustment + rain_adjustment)

        # Menghitung harga yang disesuaikan berdasarkan semua faktor
        adjusted_price = row['Harga'] * price_fluctuation  # Harga akhir yang disesuaikan
        return adjusted_price

    # Menerapkan aturan pada dataset
    merged_df_v2["Harga_Adjusted"] = merged_df_v2.apply(apply_rules, axis=1)  # Terapkan fungsi untuk menyesuaikan harga

    # Menyiapkan data untuk training (fitur dan target)
    X_v2 = merged_df_v2[["harga_bulan_lalu", "Lamanya Penyinaran Matahari (jam)", "Curah Hujan (mm)", "puasa", "natal_tahun_baru"]]  # Fitur
    y_v2 = merged_df_v2["Harga_Adjusted"]  # Target (harga yang disesuaikan)

    # Membagi data menjadi training (70%) dan testing (30%)
    X_train_v2, X_test_v2, y_train_v2, y_test_v2 = train_test_split(X_v2, y_v2, test_size=0.3, random_state=72)  # Pembagian data

    # Mendefinisikan model-model yang akan digunakan untuk evaluasi
    models_v2 = {
        "Decision Tree": DecisionTreeRegressor(random_state=72),  # Model pohon keputusan
        "Random Forest": RandomForestRegressor(n_estimators=100, random_state=72),  # Model random forest dengan 100 pohon
        "Linear Regression": LinearRegression(),  # Model regresi linear
    }

    # Evaluasi model
    results_v2 = {}
```

Fig. 4: Code Implementation Part 2

3.5. Algorithm Testing and Evaluation Results

In this study, regression modeling was conducted to predict the target variable using three different algorithms: Decision Tree Regressor, Random Forest Regressor, and Linear Regression. The dataset was split into two parts: 70% for training and 30% for testing, using the parameter random_state=72 to ensure reproducibility of the results. Each model was trained on the training data subset and tested on the testing data to evaluate its predictive performance. The evaluation was carried out using two primary metrics: Mean Absolute Error (MAE) and R-squared (R^2) score.

1. MAE measures the average absolute error between the predicted and actual values.
2. R^2 indicates how well the model explains the variability of the data.

Below is the formula used to calculate MAE and R^2:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} | \gamma_i - \hat{\gamma}_i |$$

Where:

γ_i is the actual value

$\hat{\gamma}_i$ is the predicted value

n is the number of data points

Table 1: Manual Calculation Table for 5 Samples

No	**Nilai Aktual (γ_i)**	**Nilai Prediksi $\hat{\gamma}_i$**	**Selisih Absolut**
1	48193	54,417.57	6,224.57
2	57107	59,981.53	2,874.53
3	66431	89,867.37	23,436.37
4	73970	94,573.08	20,603.08
5	86907	102,648.13	15,741.13

MAE = (6,224.57 + 2,874.53 + 23,436.37 + 20,603.08 + 15,741.13) / 5 = 68,879.68 / 5 = 13775.936

Simple Explanation:

1. Calculate the difference between the predicted value and the actual value for each data point.
2. Convert the difference to an absolute (positive) value.
3. Sum all the absolute differences.
4. Divide by the number of data points.

Based on the calculation example above, where we have 1,892 data points—70% used as training data (1,324 data points) and 30% as testing data (568 data points)—if the total absolute difference from the 568 test samples is 8,168,139.24, then:

MAE = 8,168,139.24 ÷ 568 = 14,380.53

The above explains the MAE formula and how it is calculated. Below is the formula and calculation method for R^2.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(\gamma_i - \hat{\gamma}_i)^2}{\sum_{i=1}^{n}(\gamma_i - \bar{\gamma}_i)^2}$$

Dimana:

γ_i is the actual value

$\hat{\gamma}_i$ is the predicted value

$\bar{\gamma}_i$ is the average of the actual values

Table 2: SSres Calculation Table for 5 Samples

No	Nilai Aktual (γ_i)	Nilai Prediksi $\hat{\gamma}_i$	$(\gamma_i - \hat{\gamma}_i)^2$ SSres
1	48193	54,417.57	(6,224.57)2 = 38,745,271.69
2	57107	59,981.53	(2,874.53)2 = 8,262,922.72
3	66431	89,867.37	(23,436.37)2 = 549,263,438.78
4	73970	94,573.08	(20,603.08)2 = 424,486,905.49
5	86907	102,648.13	(15,741.13)2 = 247,783,173.68

Total SSres = 38,745,271.69 + 8,262,922.72 + 549,263,438.78 + 424,486,905.49 + 247,783,173.68 = 1,268,541,712.36

Average ($\bar{y}$) = (48,193 + 57,107 + 66,431 + 73,970 + 86,907) / 5 = 332,608.00 / 5 = 66,521.6

Table 3: SStot Calculation Table for 5 Samples

No	Nilai Aktual (γ_i)	$\bar{y}$ = 66,521.6	$(\gamma_i - \bar{y})^2$ SStot
1	48193	18328.6	335,937,577.96
2	57107	9414.6	88,634,693.16
3	66431	90.6	8,208.36
4	73970	7448.4	55,478,662.56
5	86907	20385.4	415,564,533.16

Total SStot = 335,937,577.96 + 88,634,693.16 + 8,208.36 + 55,478,662.56 + 415,564,533.16 = 895,623,675.20

R^2 = 1 - (SSres / SStot) = 1 - (1,268,541,712.36 / 895,623,675.20) = 1 – 1.416 = -0.416

Simple Explanation:

1. Calculate the average of all actual values ($\bar{y}$).
2. Calculate the total residual sum of squares (SSres): the sum of the squared differences between actual and predicted values.
3. Calculate the total sum of squares (SStot): the sum of the squared differences between actual values and the average of actual values.
4. Calculate R^2 = 1 - (SSres / SStot)

Based on the calculation example above, where we have 1,892 data points—70% used as training data (1,324 points) and 30% as testing data (568 points)—if from the 568 test samples we obtain:

SSres = 2.3826×10^{12} = 2,382,600,000,000

SStot = 7.5895×10^{12} = 7,589,500,000,000

$$R^2 = 1 - \frac{SSres}{SStot} = 1 - \frac{2{,}382{,}600{,}000{,}000}{7{,}589{,}500{,}000{,}000} = 1 - 0.314 = 0.686$$

The evaluation results show that Linear Regression delivered the best performance, with a MAE of 14,380.53 and an R^2 of 0.686, indicating that the model is able to explain approximately 68.6% of the variation in the target data. Random Forest followed with a fairly competitive performance (MAE = 14,992.93, R^2 = 0.617), reflecting the strength of ensemble models in reducing overfitting. Meanwhile, Decision Tree showed lower performance, with a MAE = 20,641.79 and R^2 = 0.307, indicating that this model is less stable and tends to overfit the training data. Overall, Linear Regression proved to be the most effective model in the context of the dataset and preprocessing techniques used in this experiment. Below is a visual comparison of predicted and actual values.

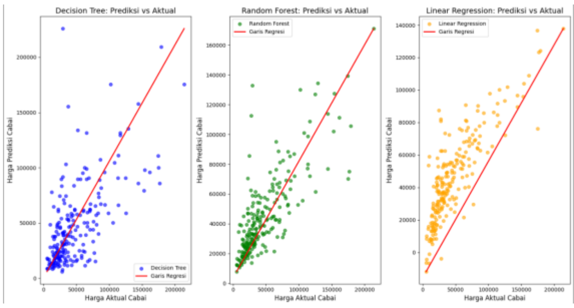

Fig. 5: Scatter Plot Comparison

The image above shows scatter plots comparing the predicted chili prices and the actual prices for three regression models: Decision Tree, Random Forest, and Linear Regression. The Decision Tree model shows widely scattered predictions from the ideal line, indicating overfitting and inaccuracy, especially at higher price values. The Random Forest model produces more accurate and stable results, with most points closer to the ideal line, although some deviations still occur at extreme values. The Linear Regression model demonstrates the most consistent predictions, closely aligned with the regression line, reflecting a strong linear relationship and the best performance based on both MAE and R^2.

3.6. Software Implementation

After completing the training process of the linear regression model using the training data, the trained model was saved into a file using the joblib module. The next step was to implement this model into a Flutter application, by first storing the trained model [17]. Below is the User Interface (UI) display of the developed application, which includes several features:

1. A home feature for predicting the price of red bird's eye chili,
2. A category feature to display different types of vegetables,
3. A store feature to show available stores,
4. A news feature that provides updates on economic, sports, national, international, and political news.

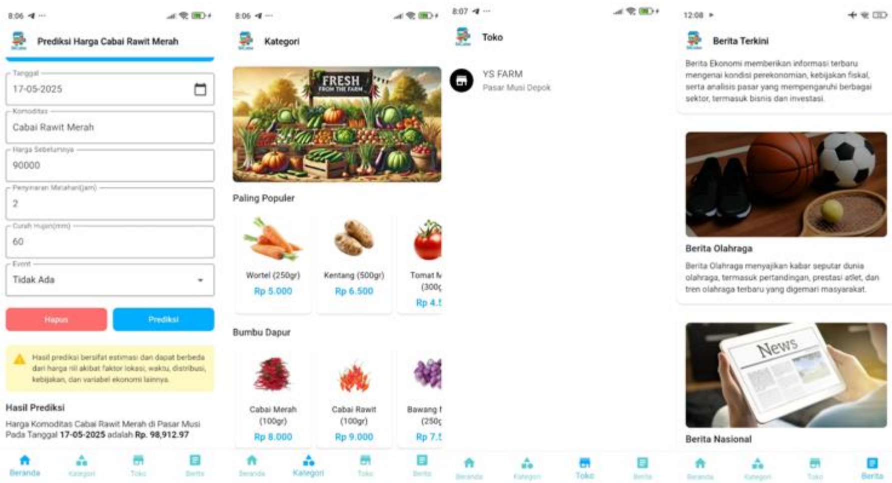

Fig. 6: Application Interface

3.7. Application Testing

In the alpha testing phase, a direct experiment was conducted to evaluate the functionality and performance of the Android-based system. The testing covered the accuracy of red bird's eye chili price predictions using linear regression, the display of vegetable prices and types, the store list along with contact features, and the news feature across various categories. The goal was to ensure that all features function as designed under real-world conditions.

Table 4: Software Testing

No	Software	Expected Outcome	Conclusion
1	Running the modeling code to generate MAE and R^2 Score in Google Colab.	Obtain MAE and R^2 Score values.	Successful
2	Running the code to display the scatter plot in Google Colab.	Obtain a scatter plot visualization.	Successful
3	Running the code to save the model as prediksi_harga_model.pkl in Google Colab.	Obtain the prediction model file prediksi_harga_model.pkl.	Successful
4	Running the API endpoint code to obtain a local server in Visual Studio Code.	Obtain a local API server that can be tested in Postman.	Successful

Table 5: Application Testing

No	Application Feature Testing	Expected Outcome	Conclusion
1	Running the SiCabe application to test red bird's eye chili price prediction by inputting previous price values, sunlight duration, rainfall, and selecting an event on the home feature.	Obtain the predicted price of red bird's eye chili.	Successful
2	Running the SiCabe application on the category feature to display vegetables.	Display the types of vegetables and their prices.	Successful
3	Placing a vegetable order or contacting a store via the vegetable detail or store detail page.	Redirected to WhatsApp to place an order or make contact.	Successful
4	Running the application on the store feature to display available stores.	Display available stores.	Successful
5	Running the application on the news feature to display available news.	Display available news.	Successful

4. Conclusion

This study utilized data from three main sources: agricultural commodity prices in Central Java, food prices in DKI Jakarta, and meteorological data such as rainfall and sunlight duration. This data was used to develop a red bird's eye chili price prediction model using three regression algorithms: Decision Tree, Random Forest, and Linear Regression. The evaluation results showed that Linear Regression delivered the best performance, with an MAE of 14,380.53 and an R^2 of 0.686, indicating that the model could explain approximately 68.6% of the price variation based on input factors such as the previous month's price, weather conditions, and seasonal events like Ramadan and the Christmas–New Year holidays.

From this analysis, it can be concluded that the two main factors causing red chili price fluctuations are weather conditions, especially rainfall and sunlight, which affect harvest outcomes, and seasonal events, such as Ramadan and year-end celebrations, which lead to spikes in demand. The testing results confirmed that all application features performed well and met expectations. Further development can include: adding other variables such as stock availability to improve prediction accuracy, expanding prediction coverage to other commodities, using more region-specific price data. Additionally, the vegetable types and store features can be further developed through the backend to enable more efficient automatic updates, and real-time data integration from various sources can be enhanced to provide faster and more accurate price information.

References

[1] L. Marianah, "Serangga Vektor Dan Intensitas Penyakit Virus Pada Tanaman Cabai Merah," *AgriHumanis: Journal of Agriculture and Human Resource Development Studies*, vol. 1, no. 2, pp. 127–134, 2020, doi: 10.46575/agrihumanis.v1i2.70.

[2] A. Fauzi *et al.*, "Pengaruh Meningkatnya Harga Cabai Terhadap Permintaan Dan Penawaran Di Indonesia," *Jurnal Akuntansi dan Manajemen Bisnis (JAMAN)*, vol. 3, no. 1, pp. 73–79, 2023.

[3] A. Puspitasari, R. Priyadi, and D. Sufyadi, "Struktur, Perilaku Dan Kinerja Pemasaran Cabai Rawit Merah Di Kecamatan Cigalontang," *AGIBUSSINES SYSTEM SCIENTIFIC JOURNAL*, vol. 1, no. 1, pp. 43–55, 2020.

[4] W. Andriani, G. Gunawan, and A. E. Prayoga, "Prediksi Nilai Emas Menggunakan Algoritma Regresi Linear," *Jurnal Ilmiah Informatika Komputer*, vol. 28, no. 1, pp. 27–35, 2023, doi: 10.35760/ik.2023.v28i1.8096.

[5] P. Pincheira-Brown, A. Bentancor, N. Hardy, and N. Jarsun, "Forecasting Fuel Prices With The Chilean Exchange Rate: Going Beyond The Commodity Currency Hypothesis," *Energy Econ*, vol. 106, pp. 1–16, Feb. 2022, doi: 10.1016/j.eneco.2021.105802.

[6] S. Makridakis, R. J. Hyndman, and F. Petropoulos, "Forecasting In Social Settings: The State Of The Art," *Int J Forecast*, vol. 36, no. 1, pp. 15–28, 2020, doi: 10.1016/j.ijforecast.2019.05.011.

[7] M. Pandia, P. Sihombing, P. Simamora, and R. Kaban, "Kajian Literatur Multimedia Retrieval : Machine Learning Untuk Pengenalan Wajah," *Jurnal Ilmu Komputer dan Sistem Informasi (JIKOMSI)*, vol. 7, no. 1, pp. 161–166, 2024.

[8] S. Supardi *et al.*, "Peran Data Mining dalam Memprediksi Tingkat Penjualan Sepatu Adidas Menggunakan Metode Algoritma Regresi Linear Sederhana," *Jurnal Ekonomi Manajemen Sistem Informasi (JEMSI)*, vol. 4, no. 5, pp. 883–890, 2023, doi: 10.31933/jemsi.v4i5.

[9] M. R. sirfatullah Alfarizi, M. Z. Al-farish, M. Taufiqurrahman, G. Ardiansah, and M. Elgar, "Penggunaan Python Sebagai Bahasa Pemrograman Untuk Machine Learning Dan Deep Learning," *Karimah Tauhid*, vol. 2, no. 1, p. 1, 2023.

[10] R. W. Arifin, A. Nurul Alfian, and Inayah Ainina Mawardi, "Implementasi Metode Mobile D pada E-Posyandu berbasis Android sebagai Alat Literasi dalam Mencegah Stunting Anak Usia Dini," *TEMATIK*, vol. 11, no. 1, pp. 110–115, Jun. 2024, doi: 10.38204/tematik.v11i1.1886.

[11] P. Anaking, M. N. P. Ma'ady, and 'Ainatul Fathiyah Abdul Rahim, "Implementation Of Rapid Application Development (RAD) Method In The Design Of Research Partner Recommendation System In Higher Education," *Asia Information System Journal (AISJ)*, vol. 2, no. 2, pp. 53–59, 2023, [Online]. Available: http://ejournal.radenintan.ac.id/index.php/AISJ/index://creativecommons.org/licenses/by-sa/4.0/

[12] K. Puteri and A. Silvanie, "Machine Learning Untuk Model Prediksi Harga Sembako Dengan Metode Regresi Linear Berganda," *Jurnal Nasional Informatika (JUNIF)*, vol. 1, no. 2, pp. 82–94, 2020, [Online]. Available: www.data.jakarta.go.id.

[13] N. Sofi and R. Dharmawan, "Perancangan Aplikasi Bengkel CSM Berbasis Android Menggunakan Framework Flutter (Bahasa Dart)," *Jurnal Teknik dan Science (JTS)*, vol. 1, no. 2, pp. 53–64, 2022.

[14] R. W. Arifin, R. Apriani, H. Wicaksono, A. Prameswara, K. Nabila SO, and S. Romlah, "Pengembangan Aplikasi E-UKM Berbasis Android Untuk Mendukung Era Digitalisasi Badan Eksekutif Mahasiswa Universitas Bina Insani," *TEMATIK*, vol. 10, no. 1, pp. 131–136, Jun. 2023, doi: 10.38204/tematik.v10i1.1344.

[15] R. Yussandi, "Analisis Dan Perancangan Sistem Informasi Simulasi Pengecatan Kendaraan Berbasis Android," *Jurnal Informatika dan Rekayasa Perangkat Lunak (JATIKA)*, vol. 2, no. 3, pp. 382–389, 2021.

[16] S. Asyuti and A. A. Setyawan, "Data Mining Dalam Penggunaan Presensi Karyawan Denga Cluster Means," *Jurnal Ilmiah Sains Teknologi Dan Informasi*, vol. 1, no. 1, pp. 1–10, 2023.

[17] A. Alfriansyah, I. Mayada, and M. Fauzi, "Perancangan Sistem Booking Jadwal Pernikahan Berbasis Mobile Apps Menggunkan Flutter Microservice," *Scientia Sacra: Jurnal Sains, Teknologi dan Masyarakat*, vol. 3, no. 2, pp. 190–200, 2023, [Online]. Available: http://pijarpemikiran.com/index.php/Scientia