

# PREDIKSI HARGA CABAI RAWIT MENGGUNAKAN LGBM, RANDOM FOREST, DAN XGBOOST

**Jason Permana <sup>1)</sup> Rakha Naufal Sujana <sup>2)</sup>Kane Pradipa Komala <sup>3)</sup> Benny Dwiyanto <sup>4)</sup>**

<sup>1,2,3,4)</sup> Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Tarumanagara, Indonesia

email : <sup>1)</sup>[jason.535220002@stu.untar.ac.id](mailto:jason.535220002@stu.untar.ac.id) <sup>2)</sup>[rakha.535220006@stu.untar.ac.id](mailto:rakha.535220006@stu.untar.ac.id) <sup>3)</sup>[kane.535220007@stu.untar.ac.id](mailto:kane.535220007@stu.untar.ac.id) <sup>4)</sup>[benny.535220012@stu.untar.ac.id](mailto:benny.535220012@stu.untar.ac.id)

## ABSTRAK

*Harga cabai rawit merah di Kota Tasikmalaya sering mengalami fluktuasi yang tajam dari waktu ke waktu. Ketidakstabilan harga ini menimbulkan ketidakpastian bagi petani, pedagang, dan konsumen, sehingga diperlukan model prediksi yang andal untuk mendukung pengambilan keputusan. Penelitian ini bertujuan membandingkan kinerja tiga algoritma machine learning berbasis ensemble, yaitu Light Gradient Boosting Machine (LGBM), Random Forest, dan Extreme Gradient Boosting (XGBoost), dalam memprediksi harga harian cabai rawit merah. Data yang digunakan berupa deret waktu univariat dari Pusat Informasi Harga Pangan Nasional (PIHPS) periode 1 Januari 2020 hingga 30 September 2025. Metode sliding window dengan jendela 30 hari digunakan untuk memprediksi harga pada hari berikutnya, dengan tuning hyperparameter menggunakan GridSearchCV dan validasi silang TimeSeriesSplit. Evaluasi dilakukan menggunakan MAE, RMSE, MAPE, dan  $R^2$  pada tiga skenario pembagian data (60:40, 70:30, dan 80:20). Hasil penelitian menunjukkan bahwa ketiga model memiliki nilai  $R^2$  di atas 0.97. LGBM unggul pada metrik MAPE (sekitar 2.6%) dan memiliki waktu pelatihan GridSearchCV tercepat, yaitu rata-rata 149.09 detik, dibandingkan XGBoost sebesar 766.16 detik dan Random Forest sebesar 1085.83 detik. Dengan mempertimbangkan akurasi dan efisiensi, LGBM dinilai sebagai model paling optimal untuk studi kasus ini.*

## Key words

*Cabai Rawit Merah, LGBM, Random Forest, Time Series, XGBoost*

## 1. Pendahuluan

Cabai rawit merah merupakan salah satu komoditas hortikultura yang berperan penting dalam konsumsi rumah tangga dan industri pangan di Indonesia [1]. Perubahan harga cabai ini sendiri sering menjadi perhatian publik karena lonjakannya dapat memengaruhi inflasi, khususnya pada pengeluaran bahan makanan [2]. Salah satu wilayah yang rentan terdampak inflasi karena

kenaikan harga tersebut adalah Kota Tasikmalaya. Faktor musiman, kondisi cuaca, distribusi, dan dinamika permintaan pasar menjadi penyebab utama ketidakstabilan harga tersebut [3], [4], [5].

Fluktuasi harga yang sulit diprediksi ini menimbulkan tantangan baik bagi petani, pedagang, maupun konsumen. Petani kesulitan menentukan strategi produksi dan pemasaran, sedangkan konsumen menghadapi ketidakpastian daya beli [2], [6], [7]. Dengan demikian, diperlukan upaya untuk membangun model prediksi harga cabai rawit merah yang dapat membantu pemangku kepentingan dalam membuat keputusan yang lebih tepat.

Berbagai penelitian sebelumnya telah menerapkan metode pembelajaran mesin seperti *Random Forest*, *Gradient Boosting*, maupun algoritma lain dalam memprediksi harga komoditas pertanian. Namun, sebagian besar penelitian tersebut masih berfokus pada komoditas secara umum atau di wilayah lain yang lebih luas. Kajian spesifik mengenai prediksi harga cabai rawit merah di Kota Tasikmalaya masih jarang dilakukan, padahal daerah ini merupakan salah satu wilayah yang rentan terhadap inflasi karena perubahan harga cabai tersebut [3], [4], [5].

Kebaruan penelitian ini terletak pada pemilihan lokasi studi, yaitu Kota Tasikmalaya, dengan memanfaatkan algoritma *Light Gradient Boosting Machine* (LGBM), *Random Forest* (RF), dan *Extreme Gradient Boosting* (XGBoost) sebagai pembanding yang berfokus pada komoditas cabai rawit merah. Dengan menitikberatkan pada konteks lokal, penelitian ini diharapkan dapat memberikan kontribusi pada literatur prediksi harga pangan serta menjadi bahan pertimbangan awal bagi kajian kebijakan di tingkat daerah.

Adapun beberapa penelitian sebelumnya terkait prediksi harga cabai telah dilakukan dengan menerapkan beragam metode, baik statistika maupun *machine learning*. Labolo et al. (2021) melakukan perbandingan antara *Support Vector Machine* (SVM), *K-Nearest Neighbor* (KNN), dan *Neural Network* (NN) untuk memprediksi harga cabai di Kota Gorontalo. Hasil penelitian mereka menunjukkan bahwa *Neural Network* menghasilkan model terbaik dengan nilai *Root Mean*

*Square Error* (RMSE) terkecil sebesar 0,206 dan akurasi rata-rata mencapai 99,25% [8].

Di sisi lain, pendekatan statistika klasik juga telah dieksplorasi. Susanti et al. (2022) menerapkan metode *Autoregressive Integrated Moving Average* (ARIMA) untuk memprediksi harga cabai rawit merah di dua pasar di Kota Pangkalpinang, yang berhasil mengidentifikasi model terbaik untuk masing-masing pasar [9]. Pendekatan lain yang juga spesifik untuk data temporal dilakukan oleh Komaria et al. (2023), yang menggunakan metode *Fuzzy Time Series* (FTS) untuk prediksi harga cabai di Jawa Timur, dan berhasil mencapai MAPE sebesar 13,09% [10].

Seiring dengan perkembangan teknik komputasi, model berbasis pohon keputusan (*tree-based*) dan *ensemble* semakin banyak diterapkan karena kemampuannya menangani pola data yang kompleks. Sebagai contoh, Riando dan Afiyati (2024) secara spesifik mengimplementasikan algoritma XGBoost untuk memprediksi harga cabai di DKI Jakarta dan berhasil mencapai nilai  $R^2$  sebesar 92% pada data uji, menunjukkan performa yang sangat baik [11]. Sementara itu, Wibowo et al. (2025) menggunakan algoritma *Decision Tree* C4.5, namun dengan pendekatan yang sedikit berbeda, yaitu untuk mengklasifikasikan tingkat volatilitas harga cabai di Kabupaten Bekasi, dengan akurasi mencapai 80,56% [12].

Selain itu, penelitian lain juga menunjukkan bahwa metode *Long Short-Term Memory* (LSTM) dan *Extreme Learning Machine* (ELM) dapat digunakan secara efektif untuk memprediksi harga pangan di Kota Tasikmalaya. Berdasarkan hasil evaluasi, metode ELM menghasilkan nilai *Mean Absolute Percentage Error* (MAPE) terendah sebesar 0,227% dan *Root Mean Square Error* (RMSE) sebesar 39,68, sedangkan metode LSTM menghasilkan MAPE sebesar 0,285% dan RMSE sebesar 43,91, yang menunjukkan bahwa kedua algoritma tersebut memiliki kemampuan prediksi yang sangat baik dalam mendekripsi pola musiman pada data harga pangan [13].

Meskipun penelitian-penelitian tersebut memberikan wawasan berharga, terdapat beberapa celah yang menjadi justifikasi bagi penelitian ini. Model seperti XGBoost dan LSTM telah terbukti efektif dalam memprediksi harga komoditas pertanian, namun sebagian besar penelitian masih berfokus pada penerapan tunggal tanpa melakukan perbandingan langsung dengan model *ensemble* canggih lainnya seperti LGBM dan *Random Forest*. Oleh karena itu, penelitian ini akan mengisi kekosongan tersebut dengan melakukan perbandingan komprehensif antara LGBM, *Random Forest*, dan XGBoost secara spesifik pada studi kasus prediksi harga cabai rawit merah di Kota Tasikmalaya.

## 2. Studi Literatur

Pada penelitian ini telah dilakukan beberapa studi literatur dan ditetapkan bahwa ini penelitian ini akan memanfaatkan 3 jenis algoritma, yaitu *Light Gradient*

*Boosting Machine* (LGBM), *Random Forest*, dan *Extreme Gradient Boosting* (XGBoost).

### 2.1. Light Gradient Boosting Machine (LGBM)

*Light Gradient Boosting Machine* atau LGBM merupakan pengembangan dari algoritma *gradient boosting* yang dirancang untuk efisiensi dan skalabilitas. Prinsip dasar *gradient boosting* adalah membangun model secara bertahap, di mana setiap pohon keputusan baru berfungsi untuk memperbaiki kesalahan prediksi dari model sebelumnya. Berbeda dengan metode *boosting* konvensional yang tumbuh secara *level-wise*, LGBM menggunakan strategi pertumbuhan pohon *leaf-wise* dengan kedalaman yang dibatasi. Pendekatan ini memungkinkan pemilihan cabang dengan potensi perbaikan terbesar pada setiap iterasi, sehingga menghasilkan peningkatan akurasi yang signifikan [14]. Secara matematis, tujuan dari algoritma ini adalah untuk meminimalkan fungsi objektif regularisasi, seperti yang ditunjukkan pada persamaan (1).

$$L(\theta) = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k) \quad (1)$$

dengan  $l(y_i, \hat{y}_i)$  sebagai fungsi *loss* antara nilai aktual dan prediksi, dan  $\Omega(f_k)$  sebagai regularisasi untuk pohon ke- $k$ . Untuk meningkatkan efisiensi, LGBM memperkenalkan dua mekanisme utama, yaitu *Gradient-based One-Side Sampling* (GOSS) yang mempertahankan sampel dengan gradien terbesar, serta *Exclusive Feature Bundling* (EFB) yang menggabungkan fitur jarang muncul agar dimensi data lebih ringkas. Kedua mekanisme ini membuat LGBM mampu mengolah data berukuran besar dengan kecepatan tinggi tanpa mengorbankan kinerja prediksi.

### 2.2. Random Forest

*Random Forest* adalah algoritma *ensemble* yang menggabungkan sejumlah pohon keputusan (*decision tree*) untuk memperoleh hasil prediksi yang lebih stabil dan akurat. Berbeda dengan satu pohon keputusan yang rentan terhadap *overfitting*, *Random Forest* memanfaatkan prinsip *bagging* atau *bootstrap aggregating*. Setiap pohon dibangun dari sampel data yang dipilih secara acak dengan penggantian, sementara pada setiap pemisahan cabang hanya dipertimbangkan subset fitur tertentu. Strategi ini menciptakan keragaman antar pohon dan mengurangi korelasi prediksi, sehingga model secara keseluruhan menjadi lebih *robust* [15].

Pada kasus klasifikasi, hasil prediksi akhir diperoleh melalui prinsip voting mayoritas, di mana setiap pohon memberikan suara untuk suatu kelas, kemudian kelas yang paling banyak dipilih ditetapkan sebagai prediksi akhir. Secara matematis, hal ini ditunjukkan pada persamaan (2).

$$\hat{y} = \sum_{b=1}^B 1\{h_b(x) = c\} \#(2)$$

Dengan  $h_b(x)$  menyatakan hasil prediksi pohon ke- $b$ ,  $B$  jumlah total pohon,  $1\{\cdot\}$  fungsi indikator yang bernilai 1 jika kondisi benar dan 0 jika salah, serta  $c$  merepresentasikan setiap kelas yang mungkin muncul. Persamaan ini menunjukkan bahwa prediksi akhir  $\hat{y}$  adalah kelas  $c$  yang memperoleh suara terbanyak dari seluruh pohon.

Sementara itu, untuk kasus regresi, hasil prediksi dihitung sebagai rata-rata dari seluruh prediksi pohon. Rumus untuk kasus regresi ditunjukkan pada persamaan (3).

$$\hat{y} = \frac{1}{B} \sum_{b=1}^B h_b(x) \#(3)$$

yang berarti setiap pohon berkontribusi sama besar terhadap prediksi akhir. Dengan pendekatan ini, *Random Forest* mampu mengurangi varians model yang biasanya tinggi pada *decision tree* tunggal, sekaligus meningkatkan kemampuan generalisasi terhadap data baru.

### 2.3. Extreme Gradient Boosting (XGBoost)

*Extreme Gradient Boosting* atau XGBoost merupakan algoritma *boosting* berbasis gradien yang dirancang dengan optimasi khusus agar lebih efisien, fleksibel, dan akurat dibandingkan implementasi *boosting* pendahulunya. Konsep dasarnya sama dengan *gradient boosting*, yaitu membangun model secara aditif dengan menambahkan pohon keputusan baru untuk memperbaiki error model sebelumnya. Keunggulan XGBoost terletak pada penggunaan regularisasi eksplisit, teknik *sparsity-aware* untuk menangani data yang jarang (*sparse*), serta kemampuan paralelisasi yang memungkinkan untuk mempercepat pelatihan [16]. Secara matematis, fungsi objektif pada XGBoost ditunjukkan pada persamaan (4).

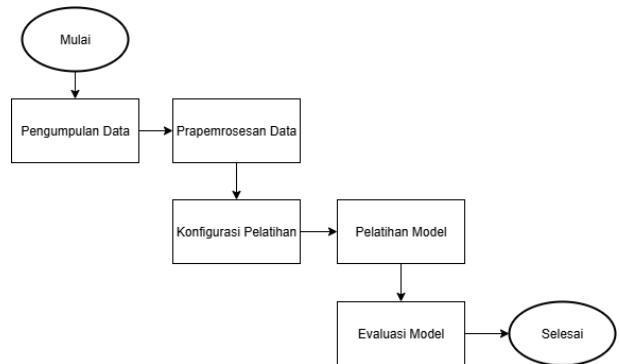
$$L(\theta) = \sum_{i=1}^n l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \#(4)$$

Dengan  $l(y_i, \hat{y}_i)$  adalah fungsi *loss*,  $\hat{y}_i^{(t-1)}$  adalah nilai prediksi hingga iterasi ke- $(t-1)$ , dan  $f_t(x)$  adalah pohon baru yang ditambahkan pada iterasi ke- $t$ . Komponen  $\Omega(f_t)$  merepresentasikan regularisasi yang membatasi jumlah daun serta bobot pohon agar model tidak *overfitting*. Dengan rancangan ini, XGBoost dapat menjaga generalisasi model pada data yang belum pernah dilihat sebelumnya.

## 3. Metodologi

Tahapan pada penelitian ini dibagi menjadi beberapa langkah, seperti ditunjukkan pada Gambar 1. Langkah-langkah tersebut meliputi proses pengumpulan data,

prapemrosesan data, konfigurasi pelatihan yang mendefinisikan skenario eksperimen, pelatihan model dengan algoritma yang telah ditentukan, dan diakhiri dengan evaluasi model untuk mengukur performa dari model terlatih. Rincian dari setiap tahapan tersebut akan diuraikan lebih lanjut pada subbab berikutnya.



Gambar 1 Diagram Alir Metode Penelitian

### 3.1. Pengumpulan Data

Data yang digunakan pada penelitian ini adalah data harian harga cabai rawit merah di Kota Tasikmalaya pada pasar tradisional dari rentang waktu 1 Januari 2020 hingga 30 September 2025 yang diperoleh dari Pusat Informasi Harga Pangan Nasional (PIHPS). Data ini berbentuk deret waktu univariat yang menyatakan harga cabai rawit merah pada setiap satuan waktu [17].

### 3.2. Pra-pemrosesan Data

Setelah data dikumpulkan, format data terlebih dahulu diubah dari bentuk awal horizontal menjadi vertikal, karena pencatatan data dari PIHPS memiliki satuan waktu sebagai kolom. Kemudian dilakukan penanganan *missing value* pada beberapa tanggal dengan melakukan *forward fill* pada tanggal tanpa data harga, yaitu data harga pada hari sebelum terjadi *missing value* digunakan untuk mengisi nilai yang hilang. Kemudian dilakukan *backward fill*, yaitu kebalikan dari *forward fill* sebagai antisipasi adanya nilai yang hilang pada tanggal-tanggal awal.

### 3.3. Konfigurasi Pelatihan

Untuk menguji konsistensi dan keandalan performa model terhadap variasi jumlah data latih, penelitian ini dirancang dengan tiga skenario eksperimen. Setiap skenario menggunakan rasio pembagian data latih (*training set*) dan data uji (*testing set*) yang berbeda. Pembagian data dilakukan secara kronologis, di mana data dengan periode waktu yang lebih awal digunakan untuk melatih model, dan data dengan periode waktu yang lebih baru digunakan untuk proses pengujian. Pendekatan ini sangat penting untuk mensimulasikan kondisi peramalan di dunia nyata dan mencegah kebocoran data.

1. Skenario 1 (Rasio 60:40): 60% data historis terawal digunakan untuk melatih model, dan 40% data sisanya yang lebih baru digunakan untuk pengujian.
2. Skenario 2 (Rasio 70:30): 70% data historis terawal digunakan untuk melatih model, dan 30% data sisanya digunakan untuk pengujian.
3. Skenario 3 (Rasio 80:20): 80% data historis terawal digunakan untuk melatih model, dan 20% data sisanya digunakan untuk pengujian.

Untuk menemukan konfigurasi optimal pada setiap skenario, dilakukan proses *hyperparameter tuning* menggunakan *GridSearchCV*. Metode ini bekerja dengan menguji seluruh kombinasi parameter dalam ruang pencarian (*parameter search space*) yang telah ditentukan, sehingga menghasilkan konfigurasi terbaik meskipun membutuhkan waktu komputasi yang lebih besar, pada penelitian ini total kombinasi parameter yang akan digunakan untuk setiap model adalah 135 kombinasi.

Validasi silang pada proses *tuning* ini menggunakan *TimeSeriesSplit* dengan 3 pembagian (*n\_splits*=3). Penggunaan *TimeSeriesSplit* memastikan bahwa pada setiap iterasi validasi, data latih selalu mendahului data validasi secara kronologis, sehingga menjaga integritas temporal dari data.

Dalam penelitian ini digunakan pendekatan *sliding window* untuk membentuk data pelatihan. Setiap himpunan data latih (X) terdiri dari 30 observasi harga berurutan, sedangkan target (y) adalah harga pada hari ke-31. Dengan mekanisme ini, model dilatih untuk memprediksi harga cabai rawit merah pada hari berikutnya (H+1) berdasarkan pola pergerakan harga dalam 30 hari sebelumnya.

Untuk model LGBM, ruang pencarian *hyperparameter* yang dieksplorasi ditunjukkan pada Tabel 1. Proses optimasi untuk model ini berfokus pada pengendalian kompleksitas pohon dan laju pembelajaran untuk mencapai keseimbangan antara akurasi dan pencegahan *overfitting*.

Tabel 1 *Hyperparameter* Model LGBM

<i>Hyperparameter</i>	Nilai
N estimators	100, 500, 1000
Learning rate	0.01, 0.05, 0.1
Num leaves	10, 20, 30, 40, 50
Max depth	-1, 10, 15

Parameter-parameter yang dioptimalkan memiliki peran penting. N\_estimators menentukan jumlah pohon keputusan yang dibangun secara berurutan, di mana peningkatan jumlah pohon dapat meningkatkan akurasi, namun jika berlebihan dapat meningkatkan waktu komputasi dan risiko *overfitting*. Learning\_rate berfungsi sebagai parameter regularisasi yang mengatur bobot kontribusi dari setiap pohon baru. Nilai yang lebih kecil akan memperlambat proses pembelajaran, memaksa model untuk membuat perbaikan secara lebih bertahap,

nilai yang lebih kecil sering kali menghasilkan model yang lebih andal. Sementara itu, Num\_leaves dan Max\_depth secara langsung mengendalikan kompleksitas dari setiap pohon individu. Num\_leaves menjadi pengendali utama pada LGBM yang tumbuh secara *leaf-wise*, sedangkan Max\_depth membatasi kedalamannya untuk mencegah pohon menangkap *noise* pada data latih.

Untuk model *Random Forest*, ruang pencarian *hyperparameter* yang ditunjukkan pada Tabel 2 difokuskan untuk mengoptimalkan keragaman antar pohon dan mengontrol kompleksitasnya, yang merupakan kelebihan dari algoritma ini.

Tabel 2 *Hyperparameter* Model *Random Forest*

<i>Hyperparameter</i>	Nilai
N estimators	100, 500, 1000
Max features	'sqrt', 'log2', 1
Max depth	10, 20, 30, 40, 50
Min samples split	2, 5, 7

Parameter N\_estimators menentukan jumlah total pohon di dalam *forest*; semakin banyak pohon, semakin stabil hasilnya, namun dengan biaya komputasi yang lebih tinggi. Max\_features adalah parameter kritis yang mengatur jumlah fitur yang dipertimbangkan secara acak di setiap simpul. Membatasi jumlah fitur ini akan meningkatkan keragaman antar pohon, yang pada akhirnya mengurangi varians dari prediksi keseluruhan. Kemudian Max\_depth dan Min\_samples\_split adalah parameter regularisasi yang mencegah setiap pohon menjadi terlalu dalam dan kompleks. Min\_samples\_split menentukan jumlah sampel minimum yang harus ada di sebuah simpul agar dapat dilakukan pemisahan, secara efektif menghentikan pertumbuhan pohon pada data yang terlalu spesifik.

Terakhir, untuk model XGBoost, ruang pencarian parameter yang ditunjukkan pada Tabel 3 dirancang untuk memanfaatkan kekuatan algoritma ini dalam hal kinerja dan regularisasi.

Tabel 3 *Hyperparameter* Model XGBoost

<i>Hyperparameter</i>	Nilai
N estimators	100, 500, 1000
Learning rate	0.01, 0.05, 0.1
Gamma	0, 0.1, 0.2
Max depth	3, 5, 7, 9, 11

Seperti pada model *boosting* lainnya, N\_estimators dan Learning\_rate berfungsi untuk mengatur proses pembelajaran sekuensial. Max\_depth secara langsung membatasi kompleksitas setiap pohon. Kemudian terdapat *hyperparameter* Gamma yang merupakan parameter regularisasi dari XGBoost, *hyperparameter* ini menentukan batas minimum penurunan *loss* yang diperlukan untuk membuat pemisahan pada daun. Nilai Gamma yang lebih tinggi akan mendorong model untuk membangun pohon yang lebih sederhana, sehingga membuat model lebih konservatif dan tahan terhadap *overfitting*. Kombinasi dari parameter-parameter ini

memungkinkan pencarian model yang tidak hanya akurat tetapi juga memiliki kemampuan generalisasi yang baik.

### 3.4. Pelatihan Model

Setelah konfigurasi pelatihan dipersiapkan, langkah berikutnya adalah melakukan pelatihan model. Menggunakan algoritma LGBM, *Random Forest*, dan XGBoost. Ketiga algoritma tersebut akan dilatih dengan konfigurasi *hyperparameter* yang telah dijelaskan sebelumnya untuk menemukan algoritma terbaik dari setiap model yang akan dilatih, juga dengan mempertimbangkan performa model pada ketiga skenario pembagian data.

### 3.5. Evaluasi Model

Setelah setiap model telah dilatih, maka performa dari masing-masing model akan dievaluasi menggunakan metrik *Mean Absolute Error* (MAE), *Root Mean Squared Error* (RMSE), *Mean Absolute Percentage Error* (MAPE), dan koefisien determinasi ( $R^2$ ). Keempat metrik ini dipilih untuk memberikan gambaran yang komprehensif mengenai kualitas prediksi, baik dari segi rata-rata kesalahan, sensitivitas terhadap kesalahan besar, tingkat kesalahan dalam skala persentase, maupun kemampuan model dalam menjelaskan variasi data aktual.

#### 1. *Mean Absolute Error* (MAE)

*Mean Absolute Error* mengukur rata-rata selisih absolut antara nilai aktual dan nilai prediksi, dengan rumus yang ditunjukkan pada persamaan (5).

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \#(5)$$

Di mana  $y_i$  adalah nilai aktual,  $\hat{y}_i$  adalah nilai prediksi, dan  $n$  adalah jumlah sampel. Nilai MAE yang lebih kecil menandakan kesalahan prediksi rata-rata yang lebih rendah.

#### 2. *Root Mean Squared Error* (RMSE)

*Root Mean Squared Error* mengukur rata-rata selisih dengan penekanan pada kesalahan yang lebih besar karena menggunakan kuadrat selisih, lalu diakarkan kembali. Rumus dari RMSE ditunjukkan pada persamaan (6).

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \#(6)$$

Di mana  $y_i$  adalah nilai aktual,  $\hat{y}_i$  adalah nilai prediksi, dan  $n$  adalah jumlah sampel. Nilai RMSE yang rendah menunjukkan bahwa prediksi mendekati nilai aktual, dengan

penekanan penalti yang lebih besar pada error yang ekstrem.

#### 3. *Mean Absolute Percentage Error* (MAPE)

*Mean Absolute Percentage Error* menyatakan tingkat kesalahan dalam bentuk persentase terhadap nilai aktual, sehingga lebih mudah dipahami secara relatif. Rumus dari MAPE ditunjukkan pada persamaan (7).

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \#(7)$$

Metrik ini menunjukkan rata-rata deviasi prediksi dalam bentuk persentase dari nilai sebenarnya.

#### 4. Koefisien Determinasi ( $R^2$ )

Koefisien determinasi atau  $R^2$  digunakan untuk menilai seberapa besar variasi data aktual yang dapat dijelaskan oleh model. Rumus dari koefisien determinasi ditunjukkan pada persamaan (8).

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y}_i)^2} \#(8)$$

Dengan  $\bar{y}$  adalah rata-rata nilai aktual. Nilai  $R^2$  yang semakin mendekati 1 menunjukkan kemampuan model yang semakin baik dalam menjelaskan data. Dengan menggunakan kombinasi metrik ini, evaluasi model dapat dilakukan secara lebih menyeluruh sehingga hasil yang diperoleh tidak hanya mempertimbangkan rata-rata kesalahan, tetapi juga variasi kesalahan besar, interpretasi dalam skala persentase, serta kualitas penjelasan model terhadap data aktual.

## 3. Hasil Percobaan

Setelah setiap model selesai dilatih, maka didapatkan hasil

Bab ini menguraikan hasil dari pengujian model prediksi harga cabai rawit merah di Kota Tasikmalaya. Evaluasi dilakukan melalui tiga skenario yang didasarkan pada rasio pembagian data latih dan data uji yang berbeda, yaitu 60:40, 70:30, dan 80:20. Untuk setiap skenario, model LGBM, *Random Forest*, dan XGBoost dioptimalkan terlebih dahulu menggunakan *GridSearchCV* guna menemukan konfigurasi *hyperparameter* yang paling efektif.

Pengujian diawali dengan skenario pembagian data 60:40. Proses optimasi untuk skenario ini menghasilkan konfigurasi *hyperparameter* terbaik bagi setiap algoritma, yang dapat dilihat pada Tabel 4.

Tabel 4 Konfigurasi *Hyperparameter GridSearch* Terbaik pada Rasio Split 60:40

Algoritma	Hyperparameter
LGBM	{'learning_rate': 0.01, 'max_depth': -1, 'n_estimators': 500, 'num_leaves': 10}
Random Forest	{'max_depth': 10, 'max_features': 1.0, 'min_samples_split': 7, 'n_estimators': 100}
XGBoost	{'gamma': 0, 'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 100}

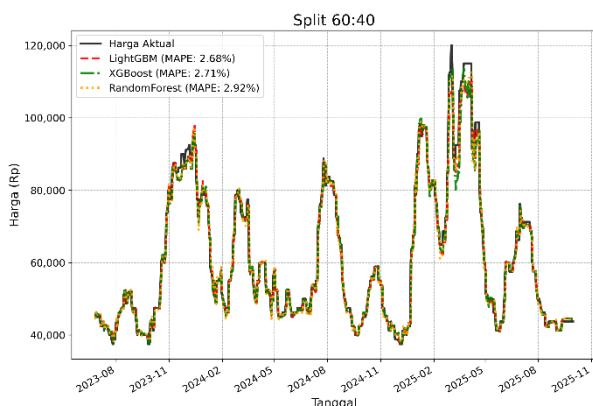
Dengan menerapkan konfigurasi tersebut, ketiga model dilatih dan diuji kinerjanya. Hasil evaluasi yang mencakup metrik MAE, RMSE, MAPE, dan  $R^2$  ditampilkan pada Tabel 5.

Tabel 5 Hasil *GridSearch* Terbaik pada Rasio Split 60:40

Algoritma	MAE	RMSE	MAPE	$R^2$
LGBM	1771.711 4	3074.613 7	0.026 8	0.975 1
Random Forest	1862.470 5	2995.124 0	0.029 2	0.976 4
XGBoost	1797.214 8	3144.892 7	0.027 1	0.974 0

Berdasarkan metrik evaluasi pada Tabel 5, ketiga model menunjukkan kinerja yang sangat kompetitif dan hampir setara. Model LGBM unggul dalam hal kesalahan persentase rata-rata dengan MAPE terendah sebesar 0.0268. Sementara itu, model *Random Forest* tercatat memiliki RMSE terendah sebesar 2995.1240 dan  $R^2$  tertinggi sebesar 0.9764, yang mengindikasikan model ini memiliki performa paling baik dalam menjelaskan variasi data. Perbedaan performa yang minim ini menunjukkan bahwa ketiga model sangat andal untuk kasus ini. Kemudian, untuk memberikan gambaran visual, Gambar 2 menyajikan perbandingan antara hasil prediksi ketiga model dengan data harga aktual.

Hasil Prediksi Model untuk Split 60:40



Gambar 2 Hasil Prediksi Harga Pada Rasio Split 60:40

Gambar 2 menunjukkan bahwa ketiga model yang diuji mampu menangkap pola umum dari data harga

aktual dengan sangat baik. Tingginya akurasi ini, yang tercermin dari nilai  $R^2$  di atas 0.97 untuk semua model, menyebabkan kurva prediksi dari LGBM, *Random Forest*, dan XGBoost tampak tumpang tindih dan sulit dibedakan secara visual. Meskipun demikian, visualisasi ini mengonfirmasi bahwa ketiga pendekatan tersebut memiliki kapabilitas yang sangat tinggi.

Selanjutnya, pengujian dilanjutkan dengan rasio pembagian data 70:30 untuk menguji konsistensi performa model pada porsi data latih yang lebih besar. Konfigurasi *hyperparameter* terbaik yang ditemukan untuk skenario ini ditampilkan pada Tabel 6.

Tabel 6 Konfigurasi *Hyperparameter GridSearch* Terbaik pada Rasio Split 70:30

Algoritma	Hyperparameter
LGBM	{'learning_rate': 0.05, 'max_depth': -1, 'n_estimators': 100, 'num_leaves': 10}
Random Forest	{'max_depth': 20, 'max_features': 1.0, 'min_samples_split': 7, 'n_estimators': 1000}
XGBoost	{'gamma': 0, 'learning_rate': 0.05, 'max_depth': 3, 'n_estimators': 100}

Dengan menerapkan konfigurasi *hyperparameter* tersebut, setiap model dilatih kembali menggunakan 70% data sebagai data latih dan kemudian diuji kemampuannya dalam memprediksi 30% sisa data. Kinerja prediktif dari masing-masing model diukur secara kuantitatif menggunakan serangkaian metrik evaluasi yang sama, yaitu MAE, RMSE, MAPE, dan  $R^2$ . Hasil evaluasi dari pengujian ini ditunjukkan pada Tabel 7.

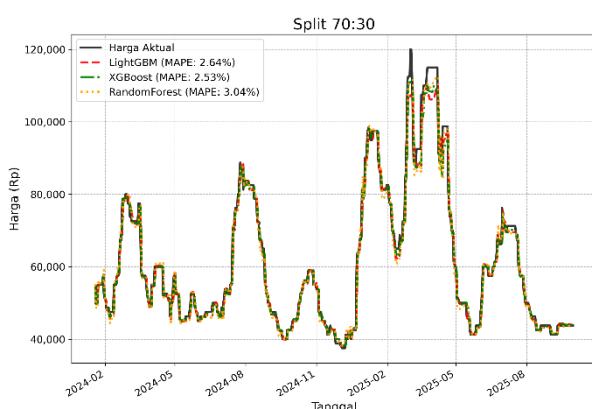
Tabel 7 Hasil *GridSearch* Terbaik pada Rasio Split 70:30

Algoritma	MAE	RMSE	MAPE	$R^2$
LGBM	1780.717 7	3212.393 5	0.026 4	0.974 2
Random Forest	1969.760 0	3276.220 5	0.030 4	0.972 0
XGBoost	1709.805 2	3139.705 5	0.025 3	0.973 0

Data pada Tabel 7 menunjukkan bahwa model XGBoost memberikan performa terbaik pada skenario ini. XGBoost berhasil memiliki nilai MAE sebesar 1709.8052, RMSE sebesar 3139.7055, dan MAPE terendah sebesar 0.0253. Di sisi lain, model LGBM tercatat memiliki nilai  $R^2$  tertinggi sebesar 0.9742. Hasil ini mengindikasikan bahwa dengan porsi data latih yang lebih besar, XGBoost mampu meminimalkan error prediksi secara lebih efektif, meskipun semua model tetap menunjukkan kinerja yang sangat baik.

Lalu sama seperti sebelumnya, untuk memberikan gambaran visual yang lebih baik, perbandingan antara hasil prediksi ketiga model dengan data harga aktual pada rasio *split* 70:30 ditunjukkan pada Gambar 3.

Hasil Prediksi Model untuk Split 70:30



Gambar 3 Hasil Prediksi Harga Pada Rasio Split 70:30

Serupa dengan skenario sebelumnya, Gambar 3 secara visual menegaskan kemampuan tinggi dari ketiga model dalam mengikuti pola data historis. Perbedaan performa yang tipis antar model tidak dapat dengan mudah diidentifikasi dari grafik saja. Namun, grafik ini berhasil menunjukkan bahwa model-model tersebut mampu beradaptasi dengan baik terhadap fluktuasi harga, meskipun terdapat beberapa tanggal dimana nilai prediksi tidak tumpang tindih dengan harga aktual.

Tahap pengujian terakhir bertujuan untuk mengevaluasi kinerja model dalam kondisi yang paling ideal, yaitu dengan memanfaatkan porsi data latih terbesar. Untuk itu, digunakan rasio pembagian data 80:20, yang memungkinkan model untuk belajar dari sebagian besar data historis yang tersedia. Proses optimasi *hyperparameter* kembali dijalankan pada pembagian data ini untuk memastikan setiap model beroperasi pada performa puncaknya. Konfigurasi optimal yang dihasilkan untuk setiap algoritma ditunjukkan pada Tabel 8.

Tabel 8 Konfigurasi *Hyperparameter* GridSearch Terbaik pada Rasio Split 80:20

Algoritma	Hyperparameter
LGBM	{'learning_rate': 0.05, 'max_depth': -1, 'n_estimators': 100, 'num_leaves': 10}
Random Forest	{'max_depth': 10, 'max_features': 1.0, 'min_samples_split': 7, 'n_estimators': 500}
XGBoost	{'gamma': 0, 'learning_rate': 0.01, 'max_depth': 3, 'n_estimators': 500}

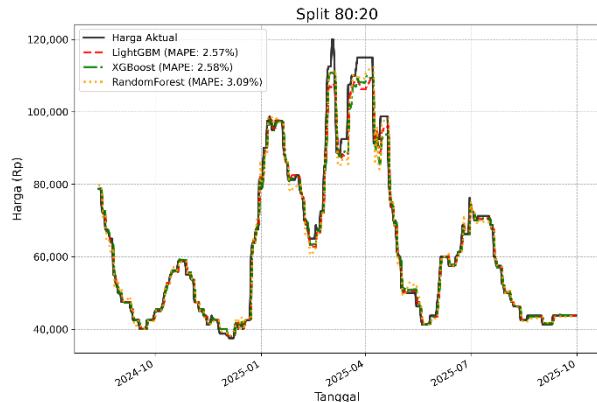
Setelah mendapatkan konfigurasi terbaik, ketiga model dilatih menggunakan 80% data. Kemampuan generalisasi model kemudian diuji pada 20% data sisa yang belum pernah dilihat sebelumnya. Hasil evaluasi dari pengujian dari skenario terakhir ini ditunjukkan pada Tabel 9.

Tabel 9 Hasil GridSearch Terbaik pada Rasio Split 80:20

Algoritma	MAE	RMSE	MAP E	R <sup>2</sup>
LGBM	1865.399	3461.447	0.025	0.975
Random Forest	2118.947	3560.003	0.030	0.974
XGBoost	1848.506	3380.580	0.025	0.976

Pada skenario dengan pembagian data 80:20, hasil evaluasi pada Tabel 9 menunjukkan performa yang terdistribusi di antara model-model unggulan. Model XGBoost tercatat paling unggul pada tiga metrik, yaitu R<sup>2</sup> tertinggi sebesar 0.9766, MAE sebesar 1848.5060 dan RMSE terendah sebesar 3380.5804. Di sisi lain, model LGBM menunjukkan keunggulan pada metrik MAPE dengan nilai terendah, yaitu 0.0257. Hal ini menunjukkan bahwa meskipun XGBoost unggul dalam hal akurasi absolut dan varians, LGBM tetap memiliki keunggulan dalam hal kesalahan prediksi relatif walaupun tidak terlalu signifikan. Terakhir, hasil visualisasi perbandingan antara hasil prediksi ketiga model dengan data harga aktual pada rasio split 80:20 ditunjukkan pada Gambar 4.

Hasil Prediksi Model untuk Split 80:20



Gambar 4 Hasil Prediksi Harga Pada Rasio Split 80:20

Gambar 4 memperlihatkan bahwa ketiga model memiliki konsistensi yang baik dalam menghasilkan prediksi yang sangat mendekati data aktual. Berdasarkan hasil pengujian pada ketiga rasio pembagian data, seluruh model menunjukkan performa yang tinggi dengan nilai R<sup>2</sup> di atas 0.97. Model LGBM menghasilkan nilai MAPE terendah pada skenario 60:40 sebesar 0.0268 dan 80:20 sebesar 0.0257, sedangkan XGBoost memperoleh MAPE terendah pada skenario 70:30 sebesar 0.0253. Sementara itu, Random Forest tetap menunjukkan hasil yang stabil meskipun sedikit lebih tinggi pada sebagian besar metrik error. Secara keseluruhan, LGBM dan XGBoost sama-sama memberikan akurasi prediksi yang unggul dengan variasi performa yang bergantung pada proporsi data latih dan data uji.

Selain metrik akurasi, aspek efisiensi komputasi juga menjadi pertimbangan penting dalam implementasi

praktis, khususnya terkait waktu yang dibutuhkan untuk melatih model. Aspek ini penting karena model yang lebih cepat dapat diiterasi dan diperbarui lebih sering. Untuk itu, dilakukan pencatatan waktu pelatihan untuk setiap algoritma hingga ditemukan *hyperparameter* optimal. Hasil pengukuran waktu pelatihan tersebut ditampilkan pada Tabel 10.

Tabel 10 Waktu Pencarian *Hyperparameter* Terbaik Pada Setiap Rasio *Split*

Rasio <i>Split</i>	Algoritma	Waktu Pelatihan (detik)
60:40	LGBM	136.91
	Random Forest	954.37
	XGBoost	678.95
70:30	LGBM	143.43
	Random Forest	1093.32
	XGBoost	764.29
80:20	LGBM	166.92
	Random Forest	1209.79
	XGBoost	855.24

Data pada Tabel 10 menunjukkan perbedaan efisiensi yang sangat signifikan di antara ketiga model. Model LGBM secara konsisten tercatat memiliki waktu pencarian *hyperparameter* yang jauh lebih cepat dibandingkan Random Forest dan XGBoost di semua skenario. Sebagai contoh, pada rasio 80:20, LGBM hanya membutuhkan 166.92 detik, sementara XGBoost memerlukan 855.24 detik dan Random Forest memerlukan 1209.79 detik. Efisiensi komputasi yang superior ini, ditambah dengan akurasi prediktif yang kompetitif, memperkuat posisi LGBM sebagai model yang paling optimal untuk studi kasus ini.

#### 4. Kesimpulan

Penelitian ini telah berhasil melakukan perbandingan kinerja antara tiga model *machine learning*, yaitu LGBM, Random Forest, dan XGBoost, untuk tugas memprediksi harga harian cabai rawit merah di Kota Tasikmalaya. Melalui serangkaian pengujian yang sistematis dengan tiga skenario pembagian data yang berbeda, serta evaluasi menggunakan metrik performa yang komprehensif, penelitian ini menghasilkan temuan-temuan penting mengenai efektivitas masing-masing algoritma dalam menangani data deret waktu harga komoditas. Berdasarkan analisis hasil percobaan yang telah dipaparkan pada bagian sebelumnya, maka dapat ditarik beberapa kesimpulan sebagai berikut:

1. Ketiga model yang diuji menunjukkan kapabilitas yang sangat baik dalam memprediksi harga cabai rawit merah. Hal ini dibuktikan dengan perolehan nilai koefisien determinasi ( $R^2$ ) yang konsisten di atas 0.97 pada seluruh skenario pengujian, yang mengindikasikan bahwa model mampu menjelaskan sebagian besar variasi pada data harga aktual.

2. Meskipun ketiga model memiliki performa yang kompetitif, model LGBM menunjukkan keunggulan paling konsisten pada metrik MAPE, dengan nilai kesalahan relatif terendah di seluruh rasio pembagian data. Selain itu, LGBM juga memiliki waktu pelatihan tercepat dibandingkan Random Forest dan XGBoost, sehingga dinilai paling efisien secara komputasi.
3. Model XGBoost tercatat memiliki nilai RMSE dan  $R^2$  yang sedikit lebih baik pada beberapa skenario tertentu, menandakan kemampuan generalisasi yang kuat terhadap pola nonlinier dalam data. Sementara itu, Random Forest menunjukkan performa yang stabil, namun secara umum berada di bawah dua model berbasis *boosting* tersebut.
4. Berdasarkan keseluruhan hasil evaluasi, LGBM dapat disimpulkan sebagai model paling optimal secara keseluruhan karena mampu mencapai keseimbangan terbaik antara akurasi relatif dan efisiensi waktu pelatihan.

Untuk pengembangan selanjutnya, penelitian ini dapat diperkaya dengan melibatkan variabel-variabel eksternal (multivariat) yang berpotensi memengaruhi harga cabai, seperti data cuaca, informasi hari libur nasional, volume distribusi, maupun data inflasi. Penambahan fitur-fitur tersebut diharapkan dapat meningkatkan akurasi model prediksi dan memberikan pemahaman yang lebih komprehensif mengenai faktor-faktor yang mendorong dinamika harga di pasar.

#### REFERENSI

- [1] A. D. Zahara T, N. S. Wisnijati, and E. Siswati, “ANALISIS PRODUKSI DAN PRODUKTIVITAS CABAI RAWIT (*Capsicum frutescens* L) di INDONESIA,” *Jurnal Ilmiah Sosio Agribis*, vol. 21, no. 1, Jun. 2021, doi: 10.30742/jisa21120211345.
- [2] O. Helbawanti, W. A. Saputro, and A. N. Ulfa, “Pengaruh Harga Bahan Pangan Terhadap Inflasi Di Indonesia,” *Agrisaintifika: Jurnal Ilmu-Ilmu Pertanian*, vol. 5, no. 2, pp. 107–116, Dec. 2021, doi: 10.32585/ags.v5i2.1859.
- [3] “Harga Cabai Tinggi Picu Inflasi di Tasikmalaya, Pemkot dan BI Stabilkan dengan Panen Raya.” Accessed: Oct. 03, 2025. [Online]. Available: <https://regional.kompas.com/read/2025/01/10/074240878/harga-cabai-tinggi-picu-inflasi-di-tasikmalaya-pemkot-dan-bi-stabilkan>
- [4] “Melonjaknya Harga Cabai Picu Kenaikan Inflasi di Tasikmalaya.” Accessed: Oct. 03, 2025. [Online]. Available: <https://www.detik.com/jabar/bisnis-d-7725030/melonjaknya-harga-cabai-picu-kenaikan-inflasi-di-tasikmalaya>
- [5] “Di Penghujung Tahun 2024, Kota Tasikmalaya Mengalami Inflasi Sebesar 0,43% - News and Press Release - BPS-Statistics Indonesia Tasikmalaya Municipality.” Accessed: Oct. 03, 2025. [Online]. Available: <https://tasikmalayakota.bps.go.id/en/news/2025/01/03/>

- 494/di-penghujung-tahun-2024--kota-tasikmalaya-mengalami-inflasi-sebesar-0-43-persen-.html
- [6] R. Rahmanta, S. F. Ayu, E. F. Fadillah, and R. S. Sitorus, "Pengaruh Fluktuasi Harga Komoditas Pangan Terhadap Inflasi di Provinsi Sumatera Utara," *JURNAL AGRICA*, vol. 13, no. 2, Sep. 2020, doi: 10.31289/agrica.v13i2.4063.
- [7] J. A. Ruslan, A. Mutolib, R. R. Bahar, and R. Salam, "PERILAKU PASAR KOMODITAS INFLASI DAN FAKTOR PENENTUNYA DI KOTA TASIKMALAYA," *JURNAL AGRIMANSION*, vol. 25, no. 1, pp. 141–149, Jun. 2024, doi: 10.29303/agrimansion.v25i1.1601.
- [8] A. Y. Labolo, A. Bode, I. Colanus, R. Drajana, and J. Karim, "COMPARASI ALGORITMA FORECASTING SVM, K-NN DAN NN UNTUK PREDIKSI HARGA CABAI KOTA GORONTALO," 2023. doi: 10.54314/jssr.v6i2.1112.
- [9] L. Susanti, S. J. Pririzki, Z. Zeleansi, D. Desy, and D. Y. Dalimunthe, "PREDIKSI HARGA CABAI RAWIT MERAH SEBAGAI KEBUTUHAN PANGAN MASYARAKAT DI KOTA PANGKALPINANG," 2022.
- [10] V. Komaria, N. El Maidah, and M. A. Furqon, "Prediksi Harga Cabai Rawit di Provinsi Jawa Timur Menggunakan Metode Fuzzy Time Series Model Lee," *Komputika : Jurnal Sistem Komputer*, vol. 12, no. 2, pp. 37–47, Sep. 2023, doi: 10.34010/komputika.v12i2.10644.
- [11] D. Riando and A. Afiyati, "Implementasi Algoritma XGBoost untuk Memprediksi Harga Jual Cabai Rawit di DKI Jakarta," *Eduvest - Journal of Universal Studies*, vol. 4, no. 9, pp. 7877–7889, Sep. 2024, doi: 10.59188/eduvvest.v4i9.3784.
- [12] M. H. Sukmana Wibowo, M. D. Al Ayubi, and E. Rilvani, "KLASIFIKASI VOLATILITAS HARGA DAGING AYAM DAN CABE RAWIT MERAH DENGAN DECISION TREE," *Jurnal Komputer dan Teknologi*, vol. 4, no. 2, pp. 138–153, Jul. 2025, doi: 10.64626/jukomtek.v4i2.455.
- [13] A. Winata, M. Dolok Lauro, and T. Handhayani, "PERBANDINGAN LSTM DAN ELM DALAM MEMPREDIKSI HARGA PANGAN KOTA TASIKMALAYA," *Jurnal Ilmu Komputer dan Sistem Informasi*, vol. 11, no. 2, Aug. 2023, doi: 10.24912/jiksi.v11i2.26015.
- [14] G. Ke *et al.*, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 3149–3157. [Online]. Available: <https://github.com/Microsoft/LightGBM>.
- [15] A. Cutler, D. R. Cutler, and J. R. Stevens, "Random Forests," in *Ensemble Machine Learning*, New York, NY: Springer New York, 2012, pp. 157–175. doi: 10.1007/978-1-4419-9326-7\_5.
- [16] T. Chen and C. Guestrin, "XGBoost," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA: ACM, Aug. 2016, pp. 785–794. doi: 10.1145/2939672.2939785.
- [17] "Berdasarkan Komoditas - PIHPS." Accessed: Oct. 03, 2025. [Online]. Available: <https://www.bi.go.id/hargapangan/TabelHarga/PasarTradisionalKomoditas>

**Jason Permana**, saat ini sebagai mahasiswa program studi Teknik Informatika, Universitas Tarumanagara.

**Rakha Naufal Sujana**, saat ini sebagai mahasiswa program studi Teknik Informatika, Universitas Tarumanagara.

**Kane Pradipa Komala**, saat ini sebagai mahasiswa program studi Teknik Informatika, Universitas Tarumanagara.

**Benny Dwiyanto**, saat ini sebagai mahasiswa program studi Teknik Informatika, Universitas Tarumanagara.