

Prediksi Harga Laptop Berdasarkan Spesifikasi Menggunakan Algoritma K-Nearest Neighbour

Adelwin Amnur[#], Ronal Hadi[#], Hanriyawan Adnan Mooduto[#], Rika Idmayanti[#],
Raemon Syaljumairi[#], Ervan Asri[#]

[#] Jurusan Teknologi Informasi, Politeknik Negeri Padang, Padang, 25142, Indonesia
E-mail: adelwin[at]pnp.ac.id

ABSTRACTS

The development of information technology is driving the increasing need for laptops as the primary computing device for various groups. Laptop prices vary greatly, influenced by specifications such as processor type, RAM capacity, storage, and additional features. This research aims to develop a laptop price prediction system based on specifications using the K-Nearest Neighbor (K-NN) algorithm, integrated into a web-based application with a Laravel API backend and MySQL database. The K-NN algorithm works by finding data with the most similar characteristics to determine the price estimate. The laptop price dataset used underwent preprocessing, including converting categorical data to numerical data so it could be processed by the model. Testing was conducted with varying K values of 3, 5, 7, and 9, using the metrics of accuracy, Mean Absolute Error (MAE), and Root Mean Square Error (RMSE). The test results showed that K = 5 provided the best performance with an accuracy of 0.89, MAE of 0.15, and RMSE of 0.18. Too small a K value makes the model sensitive, while too large a K value makes the model too general, making K = 5 the optimal point. This model is capable of producing predictions with a low error rate and high accuracy, proving the effectiveness of K-NN in predicting laptop prices based on specifications such as processor brand, processor type, RAM, storage capacity, and screen size. The success of this research also proves that the integration of Flutter, Laravel API, and MySQL can support the development of a responsive, interactive, and beneficial laptop price prediction system for both potential buyers and sellers.

*Manuscript received Sep 5, 2025;
revised Sep 17, 2025. accepted Sep
20, 2025 Date of publication Sep
30, 2025. International Journal,
JITSI : Jurnal Ilmiah Teknologi
Sistem Informasi licensed under a
Creative Commons Attribution-
Share Alike 4.0 International
License*



ABSTRAK

Perkembangan teknologi informasi mendorong meningkatnya kebutuhan laptop sebagai perangkat komputasi utama bagi berbagai kalangan. Harga laptop sangat bervariasi, dipengaruhi oleh spesifikasi seperti jenis prosesor, kapasitas RAM, penyimpanan, dan fitur tambahan. Penelitian ini bertujuan mengembangkan sistem prediksi harga laptop berdasarkan spesifikasi menggunakan algoritma K-Nearest Neighbor (K-NN) yang diintegrasikan dalam aplikasi berbasis web dengan backend Laravel API dan database MySQL. Algoritma K-NN bekerja dengan mencari data yang memiliki karakteristik paling mirip untuk menentukan estimasi harga. Dataset harga laptop yang digunakan melewati tahap preprocessing, termasuk konversi data kategorikal ke numerik agar dapat diolah oleh model. Pengujian dilakukan dengan variasi nilai K, yaitu 3, 5, 7, dan 9, menggunakan metrik akurasi, Mean Absolute Error (MAE), dan Root Mean Square Error (RMSE). Hasil pengujian menunjukkan K = 5 memberikan performa terbaik dengan akurasi 0.89, MAE 0.15, dan RMSE 0.18. Nilai K terlalu kecil membuat model sensitif, sedangkan nilai K terlalu besar membuat model terlalu general, sehingga K = 5 menjadi titik optimal. Model ini mampu menghasilkan prediksi dengan tingkat kesalahan rendah dan akurasi tinggi, membuktikan efektivitas K-NN dalam memprediksi harga laptop berdasarkan

spesifikasi seperti merk prosesor, tipe prosesor, RAM, kapasitas penyimpanan, dan ukuran layar. Keberhasilan penelitian ini juga membuktikan bahwa integrasi Flutter, Laravel API, dan MySQL dapat mendukung pengembangan sistem prediksi harga laptop yang responsif, interaktif, dan bermanfaat bagi calon pembeli maupun penjual.

Keywords / Kata Kunci — *KNN; Laptop; Klasifikasi; Harga*

CORRESPONDING AUTHOR

Adelwin Amnur
Jurusan Teknologi Informasi, Politeknik Negeri Padang, Padang, 25142, Indonesia
Email: adelwin[at]pnp.ac.id

1. PENDAHULUAN

Perkembangan teknologi informasi yang pesat telah meningkatkan kebutuhan akan perangkat komputasi seperti laptop. Laptop menjadi salah satu perangkat yang banyak digunakan oleh berbagai kalangan, mulai dari pelajar, mahasiswa, pekerja kantor, hingga profesional di bidang teknologi. Namun, harga laptop sangat bervariasi tergantung pada spesifikasi perangkat keras seperti jenis prosesor, kapasitas RAM, ukuran penyimpanan, kartu grafis, dan fitur tambahan lainnya. Variasi harga ini sering kali membuat calon pembeli kesulitan dalam menentukan laptop yang sesuai dengan kebutuhan dan anggaran mereka. Oleh karena itu, diperlukan sebuah sistem yang dapat membantu dalam memprediksi harga laptop berdasarkan spesifikasi secara akurat.

Salah satu metode yang dapat digunakan untuk prediksi harga laptop adalah algoritma K-Nearest Neighbor (K-NN). Algoritma ini bekerja dengan cara mencari data dengan karakteristik yang paling mirip untuk menentukan estimasi harga. Dengan memanfaatkan dataset harga laptop yang sudah ada, sistem dapat memberikan prediksi harga berdasarkan spesifikasi yang dimasukkan oleh pengguna. Keakuratan algoritma K-NN dalam menangani masalah klasifikasi dan regresi menjadikannya pilihan yang tepat untuk diterapkan dalam prediksi harga laptop. Selain itu, metode ini relatif mudah diimplementasikan dan dapat memberikan hasil yang cukup akurat jika data yang digunakan berkualitas.

Seiring dengan perkembangan aplikasi berbasis web, diperlukan sistem yang mudah diakses oleh pengguna kapan saja dan di mana saja. Oleh karena itu, penelitian ini akan mengembangkan aplikasi berbasis PHP sebagai antarmuka pengguna dan diintegrasikan dengan database MySQL. Dengan kombinasi teknologi ini, sistem prediksi harga laptop dapat diakses dengan mudah dan memberikan informasi yang cepat serta akurat kepada pengguna.

Penelitian ini akan difokuskan pada laptop yang tersedia di kawasan Kota Padang. Kota Padang sebagai salah satu pusat perdagangan di Sumatera Barat memiliki berbagai toko elektronik yang menjual berbagai merek dan model laptop dengan harga yang beragam. Dengan cakupan penelitian yang dibatasi pada wilayah ini, sistem yang dikembangkan dapat memberikan estimasi harga yang lebih relevan bagi masyarakat setempat. Selain itu, hasil penelitian ini juga dapat dimanfaatkan oleh pelaku usaha untuk menentukan harga jual laptop berdasarkan kondisi pasar di Kota Padang.

Dengan adanya sistem prediksi harga laptop ini, diharapkan calon pembeli dapat memperoleh informasi harga yang lebih akurat dan transparan berdasarkan spesifikasi yang mereka butuhkan. Selain itu, sistem ini juga dapat membantu toko atau penjual dalam menetapkan harga yang lebih kompetitif di pasaran. Penelitian ini juga diharapkan dapat menjadi referensi bagi pengembangan sistem serupa di bidang lain, seperti prediksi harga Laptop atau perangkat elektronik lainnya.

2. METODOLOGI PENELITIAN

Algoritme klasifikasi Nearest Neighbor, juga dikenal sebagai k-Nearest Neighbor (KNN), menggunakan data terdekat untuk memprediksi data baru yang belum dikenal (data uji). Algoritme ini bekerja dengan mencari sejumlah tetangga terdekat dari data uji dan kemudian menentukan kelas data uji berdasarkan kelas tetangga terdekat yang paling umum yang ditemukan (data latih). Baik data numerik maupun kategorikal dapat diproses dengan jarak dekat. Namun, untuk data kategorikal, perhitungan jarak perbedaan atau kesamaan tidak dapat dilakukan dengan operasi matematik yang sama seperti yang dapat dilakukan untuk data numerik. Pada data dengan dimensi rendah atau sedang, tetangga terdekat lebih efektif. Karena jumlah data yang digunakan lebih besar, algoritme ini juga efektif untuk dataset kecil hingga sedang.

K-Nearest Neighbour (KNN) untuk memprediksi harga laptop berdasarkan spesifikasinya. Algoritma ini memungkinkan sistem untuk melakukan estimasi harga dengan membandingkan spesifikasi laptop yang dimasukkan pengguna dengan data historis yang telah dikumpulkan.

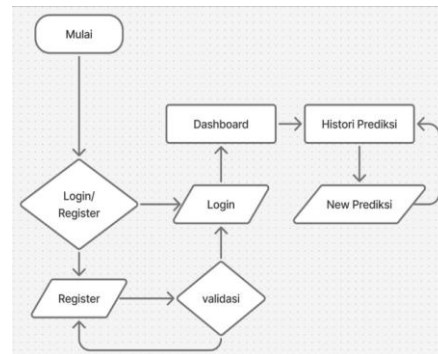
Dengan menggunakan metode KNN, sistem dapat menemukan pola harga berdasarkan spesifikasi yang paling mirip dalam dataset, sehingga prediksi yang dihasilkan lebih akurat dan realistis. Solusi ini sangat berguna bagi

pembeli, penjual, maupun analis pasar yang ingin mengetahui estimasi harga suatu laptop sebelum melakukan transaksi. Dataset harga laptop yang dikumpulkan dari berbagai sumber, khususnya di wilayah Kota Padang. Algoritma K-Nearest Neighbor (K-NN) yang akan digunakan dalam prediksi harga laptop berdasarkan spesifikasi yang dimasukkan pengguna.

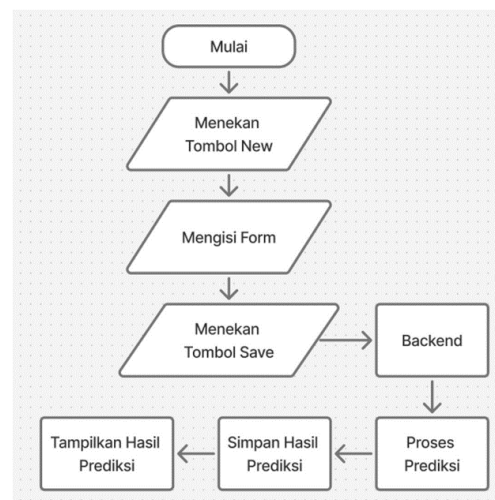
Design Sistem



GAMBAR 1. Flowchart Model Prediksi



GAMBAR 2. Flowchart Sistem



GAMBAR 3. Flowchart Proses Prediksi

Flowchart Gambar 1 ini menjelaskan alur pengolahan data yang dimulai dengan pengambilan dataset, dilanjutkan dengan preprocessing data, pelatihan model menggunakan algoritma K- Nearest Neighbor, Evaluasi hasil dan mengambil hasil terbaik, hingga penyimpanan model dalam format “.joblib”. Pada Gambar 2 Flowchart ini menjelaskan alur proses keseluruhan sistem. Awalnya Pengguna diwajibkan untuk Login. Jika pengguna memilih registrasi, data akun baru akan dimasukkan dan disimpan ke dalam database, kemudian sistem menampilkan pesan bahwa registrasi berhasil. Jika memilih login, pengguna memasukkan username dan password yang kemudian divalidasi oleh sistem. Apabila data valid, pengguna diarahkan ke Dashboard dan bisa menambahkan prediksi baru di Halaman Histori Prediksi. Gambar 3 menjelaskan alur proses Prediksi. Jika Pengguna menekan tombol New pada halaman Histori maka akan muncul tampilan form, Pengguna mengisi form keseluruhan dan menekan tombol prediksi. Setelah menekan tombol prediksi sistem akan mengirim inputan form ke Backend untuk diproses. Kemudian hasil prediksi akan disimpan dalam histori dan ditampilkan kepada pengguna di halaman histori.

Transformasi Data

Dalam proses prediksi harga laptop menggunakan algoritma K-Nearest Neighbour (KNN), data spesifikasi laptop yang berbentuk teks harus dikonversi ke dalam bentuk numerik agar dapat diolah oleh algoritma secara optimal. Transformasi ini dilakukan dengan berbagai metode seperti label encoding, one-hot encoding, dan normalisasi tergantung pada jenis atributnya. Berikut adalah rincian proses transformasi data untuk setiap komponen spesifikasi laptop. Prosesor memiliki beberapa atribut yang perlu di transformasi. Pada hasil analisis didapatkan 5 atribut yang digunakan

Atributnya yaitu merk, inti, generasi, frekuensi, dan cache, tetapi yang dilakukan transformasi adalah merk dan inti. Berikut ini tabel transformasi data yang digunakan:

No	Merk	Nilai Transformasi
1	Intel	1
2	AMD	2
3	Apple	3

No	Inti/Core	Nilai Transformasi
1	i7	1
2	i5	2
3	i3	3
4	Ryzen9	4
5	Ryzen7	5
6	Ryzen3	6

3. HASIL DAN PEMBAHASAN

Implementasi Antarmuka

Pada halaman login, pengguna diminta untuk memasukkan email dan kata sandi yang telah terdaftar sebelumnya. Kemudian diverifikasi oleh sistem untuk memastikan akun. Jika validasi berhasil, pengguna akan diarahkan ke halaman Histori Prediksi untuk mengakses Halaman Histori Prediksi dan Melakukan Prediksi. Pada halaman Register, pengguna diminta untuk memasukkan username, email dan kata sandi. Kata sandi yang dibuat akan hashing untuk menjaga kerahasiaan dan keamanan dari pengguna. Pengguna akan diarahkan ke halaman Login untuk melakukan login pada akun yang baru dibuat.

Pada Riwayat Prediksi bagi pengguna baru akan kosong, namun akan bertambah jika pengguna membuat prediksi baru dengan mengklik tombol 'New'. Pada Riwayat Transaksi ini hanya bisa dilihat pada riwayat pengguna itu sendiri, sehingga pengguna tidak bisa melihat riwayat pengguna lain dan begitupun sebaliknya. Terkecuali admin, bisa melihat keseluruhan riwayat. Jika ada user yang dihapus admin, pada riwayat user tersebut diganti menjadi 'Deleted user'. Tabel dataset menampilkan keseluruhan dataset. Admin bisa menambah data baru, mengubah dan menghapus data. Perubahan data tidak akan mempengaruhi model Random Forest karena Training Model hanya dilakukan sekali secara offline.

#	Brand	Model	Processor Vendor	Processor	RAM	Battery	Screen	Front Camera	Back Camera	Year	Price	
1	Tecno	Suratman	Apple	sadaf	50 GB	5000 mAh	6.6"	Single	Single	2035	\$155	
2	Infinix	ss	Huawei	Snapdragonssssssss	64 GB	10000 mAh	12"	Single	Single	2030	\$1	
3	Samsung	Sumsang	Qualcomm	Snapdragon	64 GB	10000 mAh	12"	Dual	Quad	2030	\$1	
4	Samsung	Sumsang	Qualcomm	Snapdragon	64 GB	10000 mAh	12"	Dual	Quad	2030	\$1	
5	Samsung	Sumsang	Qualcomm	Snapdragon	64 GB	10000 mAh	12"	Dual	Quad	2030	\$1	
6	Oppo	Reno13 F4G	MediaTek	MediaTek Helio G100	8 GB	5800 mAh	6.6"	Single	Triple	2025	\$499	
7	Oppo	Reno13 F	MediaTek	MediaTek Helio G100	8 GB	5800 mAh	6.6"	Single	Triple	2025	\$499	
8	Oppo	Reno13 Pro	MediaTek	MediaTek Dimensity 8350	12 GB	5800 mAh	6.8"	Single	Triple	2025	\$799	
9	Oppo	Reno13	MediaTek	MediaTek Dimensity 8350	12 GB	5600 mAh	6.5"	Single	Dual	2025	\$699	
10	POCO	X7	MediaTek	MediaTek Dimensity 8200	6 GB	5000 mAh	6.6"	Single	Single	2025	\$329	

GAMBAR 4. Halaman Dataset

Analisis Algoritma

1. Model K-Nearest Neighbour

Pandas digunakan untuk mengelola dan memanipulasi data, terutama data dalam bentuk tabel (DataFrame). numpy adalah library untuk operasi matematika dan array numerik, biasa digunakan untuk manipulasi data secara efisien. Fungsi train_test_split digunakan untuk membagi data menjadi data latih (training) dan data uji (testing). RandomForestRegressor adalah model machine learning yang digunakan untuk regresi (prediksi nilai kontinu). Ia bekerja dengan membuat banyak pohon keputusan (decision trees) dan menggabungkan hasilnya.

mean_squared_error: menghitung galat rata-rata kuadrat. r2_score: mengukur seberapa baik model menjelaskan varians data (semakin mendekati 1, semakin baik). OneHotEncoder digunakan untuk mengubah data kategorikal (teks) menjadi bentuk numerik biner yang bisa diproses oleh model. ColumnTransformer digunakan untuk menerapkan transformasi yang berbeda ke kolom yang berbeda dari dataset. Pipeline digunakan untuk membuat alur proses machine learning yang terstruktur dan berurutan, misalnya: preprocessing → training model → evaluasi. joblib digunakan untuk menyimpan dan memuat model machine learning ke/dari file. Berguna jika kamu ingin menyimpan model yang sudah dilatih tanpa perlu melatih ulang.

Library Python

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer from sklearn.pipeline import Pipeline
import numpy as np import joblib
```

2. Persiapan Data

Data diambil dari file .csv dengan separator/pemisah semicolon (;) dengan *code* `df = pd.read_csv('p.csv', sep=';')`

3. Pemilihan dan pembersihan data

Code Pemilihan

```
kolom_pilihan = [
    'company', 'ram', 'processor_vendor', 'battery_capacity',
    'screen_size', 'Launched_price', 'front_camera', 'back_camera'
]
df_selected = df[kolom_pilihan].dropna()
df_selected['ram'] = df_selected['ram'].astype(int)
```

Ini adalah daftar kolom yang dianggap relevan untuk digunakan dalam analisis atau pelatihan model machine learning. `df[kolom_pilihan]`: memilih kolom-kolom tertentu dari DataFrame `df`. `dropna()`: menghapus baris yang memiliki nilai kosong (NaN) di kolom mana pun yang dipilih. Tujuan: membersihkan data agar tidak ada nilai yang hilang sebelum melatih model.

Code Pembersihan data

```
df_clean = df_selected.copy()
for col in df_clean.select_dtypes(include=np.number).columns:
    Q1 = df_clean[col].quantile(0.25)
    Q3 = df_clean[col].quantile(0.75)
    IQR = Q3 - Q1
    batas_bawah = Q1 - 1.5 * IQR
    batas_atas = Q3 + 1.5 * IQR
    df_clean = df_clean[(df_clean[col] >= batas_bawah) &
    (df_clean[col] <= batas_atas)]
```

Tujuannya memakai IQR untuk Menghapus outlier dari setiap kolom numerik pada dataset `df_selected`, dan menyimpannya dalam `df_clean`.

4. Definisi Fitur dan Encode

`x` sebagai fitur (input) dan `y` sebagai target (label) kolom yang ingin diprediksi. `kolom_numerik`: Fitur numerik yang akan digunakan apa adanya (`passthrough`). `kolom_kategorikal`: Fitur kategorikal (termasuk kamera yang mungkin dikodekan sebagai label) yang akan diubah ke bentuk biner/numerik melalui One-Hot Encoding. `ColumnTransformer`: digunakan untuk menerapkan transformasi berbeda ke kolom yang berbeda. ('num', 'passthrough', `kolom_numerik`): kolom numerik tidak diubah (langsung diteruskan). ('cat', `OneHotEncoder(handle_unknown='ignore')`, `kolom_kategorikal`): kolom kategorikal dikodekan menjadi fitur biner (one-hot). `handle_unknown='ignore'`: menjaga agar data uji tidak error jika ada kategori yang belum pernah muncul di data latih. `remainder='passthrough'`: kolom yang tidak disebutkan akan tetap disertakan (tanpa diubah).

Code Definisi Fitur dan Encoding

```
X = df_clean.drop('Launched_price', axis=1)
y = df_clean['Launched_price']
kolom_numerik = ['ram', 'battery_capacity', 'screen_size']
kolom_kategorikal = ['company', 'processor_vendor', 'front_camera', 'back_camera']
preprocessor = ColumnTransformer(
    transformers=[
        ('num', 'passthrough', kolom_numerik),
        ('cat', OneHotEncoder(handle_unknown='ignore'), kolom_kategorikal)
    ],
    remainder='passthrough' # Menjaga agar urutan kolom tetap
)
```

5. Pelatihan model

Code Pelatihan Model

```
model_pipeline = Pipeline(steps=[('preprocessor', preprocessor), ('regressor',
    RandomForestRegressor(n_estimators=100, random_state=42))])
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, shuffle=True, random_state=42)
model_pipeline.fit(X_train, y_train)
```

Pipeline menyusun langkah-langkah preprocessing dan model menjadi satu kesatuan alur. 'preprocessor': langkah pertama, akan memproses data menggunakan ColumnTransformer yang sudah dibuat sebelumnya (OneHotEncoder + passthrough). 'regressor': langkah kedua, menggunakan Random Forest Regressor sebagai model prediksi harga (Launched_price). Pada split data memisahkan data latih (80%) dan data uji (20%). shuffle=True: mengacak data sebelum membagi. random_state=42: supaya pembagian selalu konsisten dan reproducible saat dijalankan berulang. Preprocessor akan menerapkan transformasi pada X_train dan hasilnya digunakan oleh Random Forest Regressor. Kemudian model akan belajar hubungan antara fitur (X_train) dan target (y_train)

6. Evaluasi model

Code Evaluasi Model

```
y_pred = model_pipeline.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print("Evaluasi Model (Random Forest):")
print(f"Root Mean Squared Error (RMSE) pada data uji: {np.sqrt(mse):.2f}")
print(f"R2 Score pada data uji: {r2:.4f}")
```

7. Simpan Model

Code Simpan Model

```
joblib.dump(model_pipeline, 'model.joblib')
print("\nModel telah disimpan sebagai 'model.joblib'")
```

model yang sudah ada akan disimpan menjadi model.joblib untuk dipakai pada backend

4. KESIMPULAN

Algoritma Random Forest digunakan untuk menganalisis spesifikasi laptop dan memprediksi harganya. Teknik ini terbukti efektif dalam memodelkan hubungan yang kompleks dan non-linear antara variabel spesifikasi seperti RAM, prosesor, kamera, ukuran layar, dan kapasitas baterai dengan harga laptop.

Untuk situasi ini, algoritma Random Forest lebih baik daripada metode regresi linear konvensional karena kemampuan untuk menangani data yang sangat besar, menemukan fitur yang paling penting, dan mengurangi risiko overfitting.

Proses persiapan data sangat penting untuk keberhasilan model prediksi, dan mencakup encoding pada data kategorikal dan pembersihan outlier dengan metode Interquartile Range (IQR). Untuk mengevaluasi kinerja model, metrik Root Mean Squared Error (RMSE) dan R2 Score digunakan.

Sistem prediksi ini diubah menjadi aplikasi web interaktif yang menggunakan backend Flask dan frontend Vue.js, dengan basis data MySQL. Dengan menggunakan aplikasi ini, pengguna dapat melakukan registrasi, membuat prediksi harga menggunakan input spesifikasi, dan melihat riwayat prediksi yang tersimpan.

REFERENSI

- [1] R. Pratama dan A. Setiawan, "Penerapan Algoritma K-Nearest Neighbor Pada Data Kategorik Untuk Klasifikasi Harga Laptop," *Jurnal Sistem Manajemen*, vol. 10, no. 2, pp. 45-56, 2023. Tersedia: <https://jsm.fmipa.unila.ac.id/index.php/jsm/article/view/39>
- [2] S. Rahmawati, "Sistem Rekomendasi Pemilihan Laptop Menggunakan Metode Content Based Filtering dan K-Nearest Neighbor," *Repository Universitas Muhammadiyah Jember*, 2022. Tersedia: <https://repository.unmuhjember.ac.id/21755>
- [3] B. P. Lubis, Hidra Amnur, and Deddy Prayama, "Implementasi Jaringan Syaraf Tiruan untuk Prediksi Cuaca pada PLTA Sumatera Barat", *jitsi*, vol. 3, no. 2, pp. 36 - 41, Jun. 2022.
- [4] H. A. Putra, "Rancang Bangun Aplikasi Penjualan Berbasis Android dengan Flutter dan Laravel Menggunakan Metode RAD pada Bengkel Mobil XCTOS," *Jurnal Media Publikasi*, vol. 7, no. 3, pp. 123-130, 2023. Tersedia: <https://journal.mediapublikasi.id/index.php/oktal/article/view/3225>
- [5] B. Suryanto, "Prediksi Harga Penjualan Mobil Bekas dengan Menggunakan Algoritma K- Nearest Neighbors dan Linear Regression," *Repository Universitas Multimedia Nusantara*, 2021. Tersedia: <https://kc.umn.ac.id/id/eprint/34053>
- [6] Hidra Amnur, A. K. Vadreass, and M. Ridwan, "Aplikasi Pendeteksi Kematangan Tanaman Menggunakan Metode Transformasi Ruang Warna HSI (Hue, Saturation, Intensity) dan K-NN (K- Nearest Neighbor)", *jitsi*, vol. 5, no. 4, pp. 161 -167, Dec. 2024.
- [7] T. Wijaya dan M. Lestari, "Perbandingan Akurasi Prediksi Harga Rumah Algoritma K- Nearest Neighbor dengan Algoritma Extreme Gradient Boost (XGBoost)," *Repository Universitas Mercu Buana*, 2022. Tersedia: <https://repository.mercubuana.ac.id/81505>
- [8] Rina, "Algoritma K-Nearest Neighbor (KNN): Penjelasan dan Implementasi untuk Klasifikasi Kanker", *medium.com*, 2023, Tersedia: <https://esairina.medium.com/algoritma-k- nearest-neighbor-knn- penjelasan-dan-implementasi-untuk-klasifikasi-kanker-ff9b7fbe0a4>
- [9] P. DuBois, *MySQL: The Complete Reference*. McGraw-Hill Education, 2013.
- [10] B. Schwartz, P. Zaitsev, and V. Tkachenko, *High Performance MySQL: Optimization, Backups, and Replication*. O'Reilly Media, 2012
- [11] L. Ullman, *MySQL, Second Edition: Visual QuickStart Guide*. Peachpit Press, 2011.
- [12] E. Redmond and J. Wilson, *Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement*. Pragmatic Bookshelf, 2012.
- [13] *MySQL Documentation*, "MySQL 8.0 Reference Manual," Oracle Corporation, 2024. [Online]. Available: <https://dev.mysql.com/doc/>.
- [14] F. Khoirudin, "SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN SMARTPHONE MENGGUNAKAN METODE K – NEAREST NEIGHBOR (KNN) BERBASIS WEB", *jitsi*, vol. 4, no. 2, pp. 82 - 90, Jun. 2023.
- [15] F. Khoirudin, "Sistem Pendukung Keputusan Pemilihan Smartphone Menggunakan Metode K – Nearest Neighbor (KNN) Berbasis Web", *jitsi*, vol. 4, no. 2, pp. 82 - 90, Jun. 2023.
- [16] Fitri Purwaningtias, M. Ariandi, and Suyanto, "Prototype Prediksi Persediaan Menggunakan Metode Weight Moving Average ", *jitsi*, vol. 4, no. 4, pp. 147 - 153, Dec. 2023.