



Klasifikasi *Smartphone* Berdasarkan Spesifikasi Menggunakan *Decision Tree C4.5* dan *Random Forest* dengan *Hyperparameter Tuning*

Akbar Ainurrofik, Ilham Saifudin*, Rosita Yanuarti

Program Studi Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Jember

akbarrofik51@gmail.com, ilham.saifudin@unmuhjember.ac.id*, rosita.yanuarti@unmuhjember.ac.id

Abstract

The rapid growth of smartphone products has produced substantial variation in technical specifications and price segments, making manual grouping increasingly complex. This study identifies the most informative specification using the C4.5 Decision Tree and develops a smartphone price-segment classification model using Random Forest optimized through hyperparameter tuning. The dataset contains 3,260 smartphone records and six input features: RAM capacity, internal storage, number of processor cores, battery capacity, main-camera resolution, and screen size. Prices in Indian rupees were converted into Indonesian rupiah and used only to generate three class labels: Entry-Level, Mid-Range, and Flagship; price was excluded from model inputs to prevent data leakage. The dataset was divided using a stratified 80:20 split. C4.5 analysis identified internal storage as the root node at a threshold of 192 GB with a Gain Ratio of 0.3804. The baseline Random Forest achieved 85.12% accuracy, 82.92% macro precision, 80.81% macro recall, and 81.77% macro F1-score. GridSearchCV with five-fold StratifiedKfold selected 100 trees, max_depth 20, min_samples_split 2, min_samples_leaf 1, max_features sqrt, and bootstrap=True. The tuned model maintained 85.12% accuracy while improving macro precision to 83.44%, macro recall to 81.03%, and macro F1-score to 82.15%. The tuned model was selected because it produced a better balance across classes and was subsequently implemented in a web-based classification system.

Keywords: C4.5 Decision Tree, GridSearchCV, Hyperparameter Tuning, Random Forest, Smartphone Classification.

1. Pendahuluan

Perkembangan *smartphone* di Indonesia berlangsung cepat dan menghasilkan banyak perangkat dengan variasi kapasitas RAM, penyimpanan internal, prosesor, kamera, layar, baterai, serta harga. Banyaknya alternatif tersebut membuat proses pengelompokan *smartphone* ke dalam segmen tertentu menjadi semakin kompleks jika dilakukan secara manual. Sistem pendukung berbasis data dibutuhkan agar karakteristik teknis perangkat dapat diproses secara terstruktur, konsisten, dan objektif menurut Yuni & Artaningsih (2022)

Klasifikasi dalam *machine learning* dapat digunakan untuk mempelajari hubungan antara atribut masukan dan kelas target yang telah ditentukan. Dalam konteks *smartphone*, spesifikasi teknis diperlakukan sebagai fitur, sedangkan segmen harga diperlakukan sebagai kelas yang diprediksi. Pendekatan *supervised learning* sesuai untuk kondisi tersebut karena model dilatih menggunakan data berlabel dan selanjutnya digunakan untuk memprediksi data baru (Maisyaroh et al., 2025). Kelas yang digunakan dalam penelitian ini terdiri atas *Entry-Level*, *Mid-Range*, dan *Flagship*, sebagaimana pengelompokan kelas *smartphone* yang juga digunakan dalam penelitian klasifikasi sebelumnya.

Penelitian Yakuf Al Jarkhi (2025) menunjukkan bahwa algoritma *Naive Bayes* dapat digunakan untuk mengklasifikasikan kisaran harga *smartphone* berdasarkan atribut RAM, ROM, ukuran layar, kamera, dan kapasitas baterai. Hasil tersebut menegaskan bahwa spesifikasi teknis merupakan informasi yang relevan untuk menentukan kelas harga. Namun, *Naive Bayes* menggunakan asumsi independensi antarfitur, sedangkan atribut *smartphone* pada praktiknya dapat saling berkaitan. Kapasitas RAM, penyimpanan, kemampuan prosesor, kamera, dan teknologi perangkat sering kali berkembang secara bersamaan pada kelas produk yang lebih tinggi.

Decision Tree C4.5 membentuk aturan keputusan yang mudah diinterpretasikan serta menentukan atribut pemisah melalui *Entropy*, *Information Gain*, *Split Information*, dan *Gain Ratio*. Metode ini sesuai untuk menunjukkan

atribut yang paling informatif pada tahap awal pembentukan pohon. Penelitian Ramadhani et al. (2025) juga menunjukkan bahwa C4.5 dapat menghasilkan model klasifikasi yang mudah dipahami. Dalam penelitian ini, C4.5 digunakan secara khusus untuk menganalisis *root node*, bukan sebagai model utama yang dibandingkan berdasarkan metrik klasifikasi. Penelitian Sari et al. (2022) menerapkan algoritma C4.5 untuk proses klasifikasi menggunakan beberapa atribut kondisi dan menghasilkan model keputusan berdasarkan pola data yang tersedia.

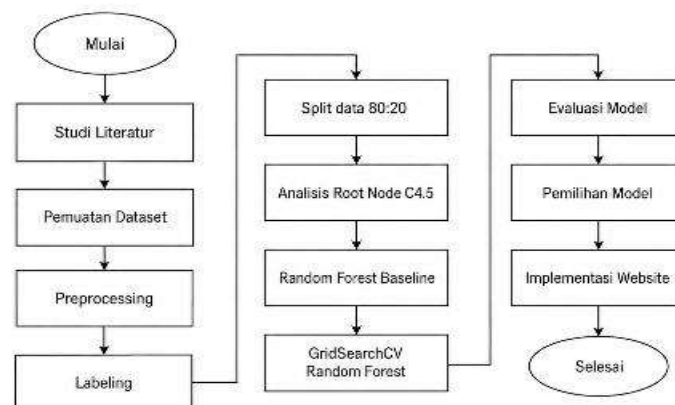
Random Forest dipilih sebagai model klasifikasi utama karena menggabungkan banyak pohon keputusan melalui pendekatan *ensemble learning*. Penelitian Gunawan et al. (2021) juga menjelaskan bahwa penggunaan banyak pohon dapat meningkatkan stabilitas prediksi dan mengurangi ketergantungan pada satu struktur pohon. Selain itu, penelitian yang telah dilakukan oleh ChayyChayy Biliqisth & Ikhsanuddin (2025) menunjukkan bahwa *Random Forest* mampu memberikan performa klasifikasi yang tinggi dan konsisten dibandingkan dengan beberapa algoritma lain. Pada penelitian yang dilakukan oleh Michael Lauw et al. (2023) diterapkan *Random Forest* untuk melakukan klasifikasi dengan memanfaatkan sejumlah atribut data sebagai variabel masukan. Hasil penelitian tersebut menunjukkan bahwa *Random Forest* mampu menghasilkan performa klasifikasi yang baik dan relevan untuk penelitian ini. Walaupun demikian, kinerja *Random Forest* dipengaruhi oleh konfigurasi *hyperparameter*, seperti jumlah pohon, kedalaman maksimum, jumlah minimum sampel untuk pemisahan, jumlah minimum sampel pada daun, jumlah fitur yang dipertimbangkan, dan penggunaan *bootstrap*.

Penelitian Nur Fauzi et al. (2025), Fakhriza et al. (2025), dan Setyowati et al. (2025) menunjukkan bahwa pengaturan *hyperparameter* yang tepat dapat meningkatkan kinerja *Random Forest*. *RandomizedSearchCV* dapat menjadi alternatif yang efisien untuk ruang pencarian yang luas, menurut Reynaldi et al. (2025). Namun, ruang parameter pada penelitian ini dibatasi menjadi 4 kombinasi, sehingga *GridSearchCV* dipilih agar seluruh kombinasi dapat diuji secara sistematis melalui validasi silang. Pemilihan model tidak hanya didasarkan pada *Accuracy*, tetapi juga pada *F1-Score Macro* agar setiap kelas memperoleh bobot evaluasi yang setara.

Berdasarkan permasalahan tersebut, penelitian ini menerapkan *Decision Tree C4.5* untuk menentukan atribut *pada node akar* dan *Random Forest* sebagai model klasifikasi utama. *Random Forest* diuji dalam dua skenario, yaitu skenario *baseline* dan skenario *penyesuaian hyperparameter*. Penelitian ini bertujuan: (1) mengimplementasikan C4.5 dan *Random Forest* untuk klasifikasi *smartphone* berdasarkan spesifikasi; dan (2) menganalisis pengaruh penyesuaian *hyperparameter* terhadap performa *Random Forest*. Model terbaik kemudian diterapkan pada *website* agar pengguna dapat melakukan klasifikasi melalui antarmuka yang lebih mudah digunakan.

2. Metode Penelitian

Penelitian menggunakan pendekatan kuantitatif eksperimental. Pendekatan tersebut dipilih karena proses penelitian melibatkan pengolahan data numerik, pelatihan model, pengujian kombinasi parameter, serta pengukuran performa menggunakan Confusion Matrix dan metrik *Accuracy*, *Precision*, *Recall*, serta *F1-Score* Ningrum et al. (2025). Tahapan penelitian meliputi studi literatur, pemuatan *dataset*, *preprocessing*, *labeling*, pembagian data 80:20, analisis *root node* C4.5, pelatihan *Random Forest baseline*, *GridSearchCV*, evaluasi model, pemilihan model, dan implementasi *website*. Gambar 1 merupakan alur penelitian klasifikasi *smartphone*.



Gambar 1. Alur penelitian klasifikasi *smartphone*

2.1 Dataset dan Preprocessing

Dataset sekunder diperoleh dari Kaggle dalam format CSV dan terdiri atas 3.260 data *smartphone* dengan 20 atribut awal. Tahap *preprocessing* dilakukan untuk menyiapkan data agar bersih, konsisten, dan sesuai dengan kebutuhan pemodelan yang dijelaskan dalam penelitian Muchalim Danu Warta et al. (2025). Atribut yang digunakan sebagai fitur model adalah *ram*, *storage*, *processor_cores*, *battery*, *main_camera*, dan *screen*. Merek dan nama model dipertahankan sebagai informasi identitas, sedangkan harga hanya digunakan untuk membentuk label.

Nama kolom diseragamkan, data duplikat diperiksa, dan nilai numerik yang tidak logis diubah menjadi *missing value*. Pemeriksaan menemukan 175 nilai kosong pada atribut jumlah inti prosesor. Nilai tersebut tidak langsung diisi pada seluruh dataset, tetapi ditangani melalui SimpleImputer dengan strategi median dalam *pipeline* agar proses imputasi hanya mempelajari data latih. Pendekatan median dipilih karena lebih *tahan* terhadap *outlier*. Harga yang masih menggunakan INR dikonversi menjadi IDR dengan nilai tukar Rp190 per INR. Tabel 1 menunjukkan Karakteristik dataset dan pembagian data.

Tabel 1. Karakteristik *dataset* dan pembagian data

Keterangan	Nilai
Jumlah data	3.260
Jumlah fitur <i>input</i>	6
Atribut target	<i>segment_smartphone</i>
<i>Entry-Level</i>	2.020 data (1.616 latih; 404 uji)
<i>Mid-Range</i>	909 data (727 latih; 182 uji)
<i>Flagship</i>	331 data (265 latih; 66 uji)
Pembagian data	80% latih dan 20% uji secara <i>stratified</i>
<i>Random state</i>	123

2.2 Pembentukan Label

Dataset tidak menyediakan target klasifikasi secara langsung. Oleh karena itu, label dibentuk berdasarkan harga setelah dikonversi dari INR ke IDR. Menurut James et al. (2023) harga merupakan variabel numerik, sedangkan target klasifikasi harus berbentuk kategori kualitatif. Aturan pembentukan label ditunjukkan pada Persamaan (1).

$$\begin{aligned} \text{Entry Level} &= \text{jika harga IDR} < \text{Rp}3.000.000 \\ \text{Mid - Range} &= \text{jika } \text{Rp}3.000.000 \leq \text{harga IDR} < \text{Rp}8.000.000 \\ &= \text{jika harga IDR} \geq \text{Rp}8.000.000 \end{aligned} \quad (1)$$

Harga tidak dimasukkan ke dalam fitur model karena penggunaannya sebagai fitur akan menyebabkan *data leakage*: model dapat mempelajari batas harga yang sama dengan aturan pembentukan label, bukan hubungan antara spesifikasi dan segmen *smartphone*.

2.3 Decision Tree C4.5

C4.5 digunakan untuk menentukan atribut paling informatif sebagai *root node*. Implementasi dilakukan dengan mengevaluasi kandidat *threshold* numerik pada setiap fitur menggunakan *Entropy*, *Information Gain*, *Split Information*, dan *Gain Ratio*. *Entropy* mengukur ketidakmurnian kelas pada himpunan data dan dirumuskan dalam Persamaan (2).

$$Entropy(S) = - \sum_{i=1}^n p_i \log_2 p_i \quad (2)$$

Information Gain mengukur pengurangan ketidakpastian setelah data dipisahkan menggunakan atribut A dan dirumuskan dalam Persamaan (3).

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^n \frac{|S_i|}{|S|} Entropy(S_i) \quad (3)$$

Split Information mengukur potensi intrinsik dari pembagian data, sebagaimana ditunjukkan pada Persamaan (4).

$$Split Information(A) = - \sum_{i=1}^n \frac{|S_i|}{|S|} \log_2 \left(\frac{|S_i|}{|S|} \right) \quad (4)$$

Gain Ratio menormalisasi *Information Gain* menggunakan *Split Information* untuk mengurangi kecenderungan memilih atribut yang menghasilkan pembagian tertentu. Persamaannya ditunjukkan pada Persamaan (5).

$$Gain Ratio(A) = \frac{Gain(S, A)}{Split Information(A)} \quad (5)$$

Atribut dengan *Gain Ratio* tertinggi dipilih sebagai *node akar*. Untuk menghindari *threshold* ekstrem yang hanya memisahkan sangat sedikit data, setiap cabang diwajibkan memuat sekurang-kurangnya 5% dari jumlah data pelatihan.

2.4 Random Forest

Random Forest membangun sejumlah pohon keputusan dari sampel *bootstrap* dan subset fitur yang dipilih secara acak. Setiap pohon menghasilkan prediksi kelas, kemudian keputusan akhir diperoleh dari kelas yang paling sering muncul. Bentuk sederhananya ditunjukkan pada Persamaan (6).

$$\hat{y} = mode(Tree1, Tree2, \dots, Treen) \quad (6)$$

Secara formal, mekanisme *majority voting* dirumuskan dalam Persamaan (7), dengan $h_t(x)$ sebagai prediksi pohon pada waktu t , C sebagai himpunan kelas, dan I sebagai fungsi indikator.

$$\hat{y} = \arg \max_{c \in C} \sum_{t=1}^T I(h_t(x) = c) \quad (7)$$

Model *baseline* dibangun menggunakan *pipeline SimpleImputer(strategy="median")* dan *RandomForestClassifier*. Parameter *random_state* ditetapkan ke 123 dan *class_weight="balanced"* digunakan agar model memberikan perhatian yang lebih proporsional terhadap kelas *Flagship* yang jumlahnya lebih sedikit.

2.5 Hyperparameter Tuning

Hyperparameter tuning dilakukan menggunakan *GridSearchCV* pada data pelatihan. Validasi internal menggunakan lima-*fold StratifiedKfold* dengan *shuffle* dan *random_state* 123. Metrik pemilihan parameter adalah *F1-Score (Macro)*. Data uji dipertahankan sebagai *hold-out* dan baru digunakan setelah proses pencarian parameter selesai. Tabel 2 menunjukkan ruang pencarian *hyperparameter Random Forest*.

Tabel 2. Ruang pencarian *hyperparameter Random Forest*

<i>Hyperparameter</i>	Nilai yang diuji
<i>n_estimators</i>	100; 150
<i>max_depth</i>	<i>None</i> ; 20
<i>min_samples_split</i>	2
<i>min_samples_leaf</i>	1
<i>max_features</i>	<i>sqrt</i>
<i>bootstrap</i>	<i>True</i>

Ruang pencarian menghasilkan 4 konfigurasi model. Masing-masing konfigurasi diuji pada lima *fold*, sehingga *GridSearchCV* menjalankan 20 proses pelatihan dan validasi.

2.6 Evaluasi Model

Kinerja model diukur menggunakan *Accuracy*, *Precision*, *Recall*, dan *F1-Score*. *Confusion Matrix* digunakan untuk memperoleh jumlah *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Rumus evaluasi mengacu pada metrik klasifikasi yang digunakan dalam penelitian *machine learning*.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (8)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (9)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (10)$$

(11)

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Karena penelitian ini merupakan klasifikasi multikelas dengan distribusi yang tidak seimbang, *Precision*, *Recall*, dan *F1-Score* dihitung menggunakan *macro average*. Setiap metrik dihitung untuk masing-masing kelas, kemudian dirata-ratakan tanpa mempertimbangkan jumlah data pada setiap kelas. Dengan demikian, kelas *Flagship* tetap memperoleh bobot evaluasi yang sama seperti kelas *Entry-Level* dan *Mid-Range*.

2.7 Implementasi Website

Model terpilih disimpan dalam berkas *random_forest_best_model.joblib* diintegrasikan ke dalam aplikasi *web* menggunakan Flask sebagai *backend*. Tailwind CSS digunakan untuk membangun antarmuka responsif. Pengguna memasukkan 6 spesifikasi numerik, kemudian sistem menampilkan hasil prediksi kelas, probabilitas kelas, analisis kontribusi spesifikasi, serta contoh *smartphone* dari kelas yang sama.

3. Hasil dan Pembahasan

3.1 Hasil Preprocessing

Pemeriksaan awal menunjukkan bahwa *dataset* ini memiliki 3.260 baris dan 20 kolom. Tidak ditemukan baris duplikat identik. Sebanyak 175 nilai kosong ditemukan pada atribut *processor_cores*, sedangkan atribut *input* lainnya tidak memiliki nilai kosong. Setelah seleksi fitur dan pembentukan label, distribusi kelas terdiri atas 2.020 *Entry-Level*, 909 *Mid-Range*, dan 331 *Flagship*. Pembagian *stratified* menghasilkan 2.608 data latih dan 652 data uji dengan proporsi kelas yang tetap terjaga.

3.2 Hasil Analisis Root Node C4.5

Perhitungan *Gain Ratio* diterapkan pada seluruh data pelatihan. Setiap kandidat *threshold* numerik dievaluasi dengan syarat ukuran minimal cabang sebesar 5% dari data latih. Hasil pengurutan atribut ditunjukkan pada Tabel 3.

Tabel 3. Hasil analisis *Gain Ratio* untuk penentuan *root node*

Peringkat	Atribut	Threshold	Information Gain	Split Information	Gain Ratio
1	Penyimpanan/ROM	192 GB	0,2494	0,6556	0,3804
2	RAM	7 GB	0,2995	0,8463	0,3539
3	Kamera utama	11 MP	0,1196	0,7298	0,1638
4	Jumlah inti prosesor	5 inti	0,0838	0,7260	0,1154
5	Ukuran layar	6,61 inci	0,0893	0,8191	0,1091
6	Kapasitas baterai	2.505 mAh	0,0439	0,5678	0,0773

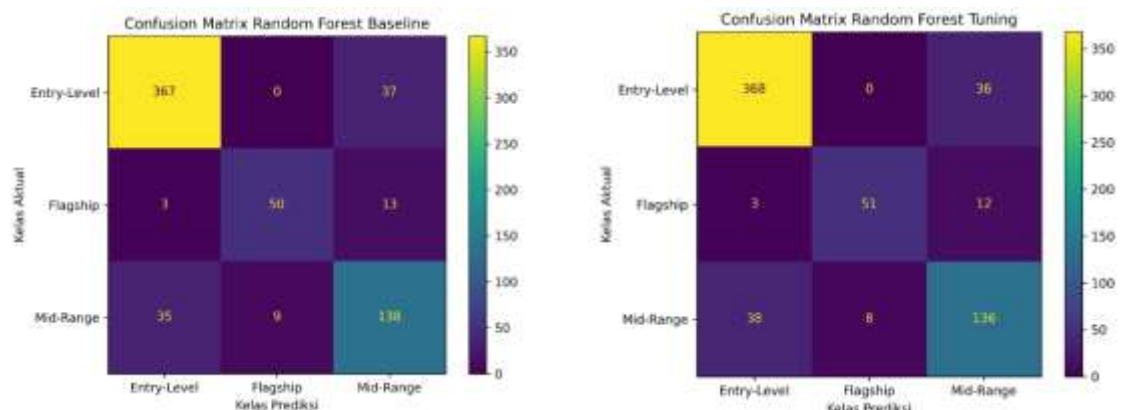
Kapasitas penyimpanan memperoleh *Gain Ratio* tertinggi sebesar 0,3804 pada *threshold* 192 GB, sehingga dipilih sebagai *node akar*. RAM memiliki *Information Gain* lebih tinggi, yaitu 0,2995, tetapi *Split Information*-nya juga lebih besar. Setelah dinormalisasi, *Gain Ratio* RAM menjadi 0,3539 dan berada pada peringkat kedua. Hasil ini menunjukkan bahwa penyimpanan merupakan atribut paling efektif untuk pemisahan awal, tetapi bukan satu-satunya atribut yang menentukan kelas *smartphone*.

3.3 Hasil Random Forest Baseline dan Tuning

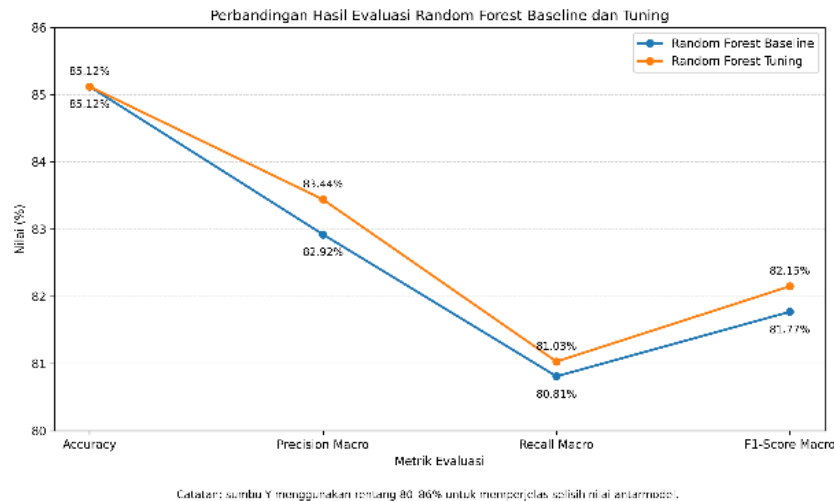
Random Forest baseline dilatih menggunakan 2.608 data latih dan dievaluasi pada 652 data uji. Model menghasilkan 555 prediksi yang benar dan 97 prediksi yang salah. *GridSearchCV* kemudian memilih konfigurasi terbaik berupa *n_estimators*=100, *max_depth*=20, *min_samples_split*=2, *min_samples_leaf*=1, *max_features*="sqrt", dan *bootstrap*=True. Hasil evaluasi kedua model ditunjukkan pada Tabel 4.

Tabel 4. Perbandingan performa *Random Forest baseline* dan *tuning*

Model	Accuracy	Precision Macro	Recall Macro	F1-Score Macro
<i>Random Forest Baseline</i>	85,12%	82,92%	80,81%	81,77%
<i>Random Forest Tuning</i>	85,12%	83,44%	81,03%	82,15%
Selisih	0,00 poin	+0,52 poin	+0,22 poin	+0,37 poin



Gambar 2. *Confusion Matrix Random Forest baseline* (kiri) dan *tuning* (kanan)



Gambar 3. Perbandingan metrik evaluasi *Random Forest baseline dan tuning*

Pada model *baseline*, prediksi yang benar terdiri atas 367 data *Entry-Level*, 50 data *Flagship*, dan 138 data *Mid-Range*. Model *tuning* menghasilkan 368 data *Entry-Level*, 51 data *Flagship*, dan 136 data *Mid-Range* yang diprediksi dengan tepat. Total prediksi benar tetap 555, sehingga *Accuracy* tidak berubah. Namun, distribusi prediksi antarkelas berubah: prediksi benar untuk *Entry-Level* dan *Flagship* masing-masing bertambah 1, sedangkan untuk *Mid-Range* berkurang 2.

3.4 Analisis Pengaruh Hyperparameter Tuning

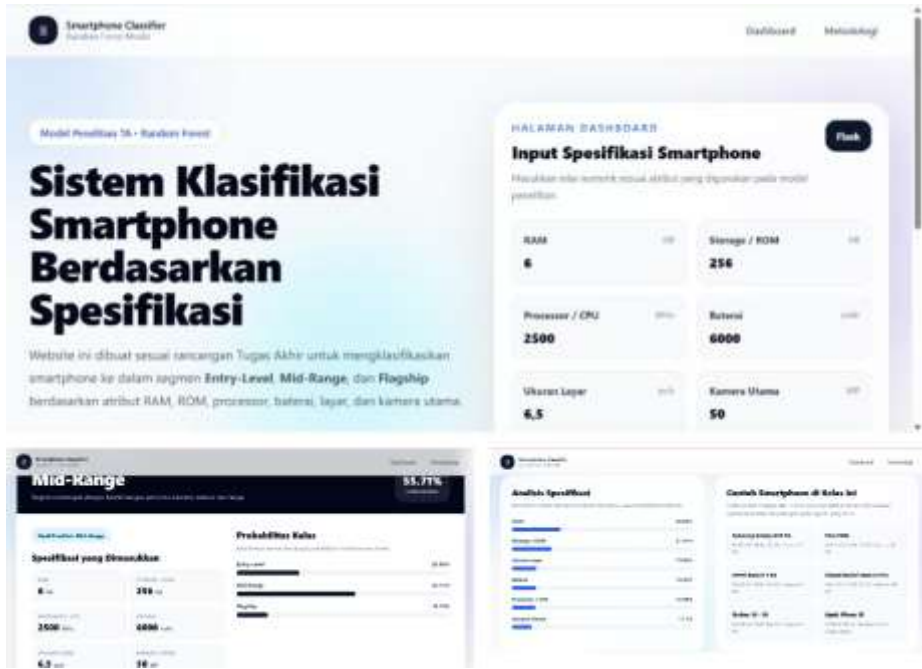
Akurasi kedua model sama karena jumlah prediksi yang benar tidak berubah. Kondisi tersebut tidak berarti bahwa *tuning* tidak berpengaruh. *Precision Macro* meningkat 0,52 poin, *Recall Macro* meningkat 0,22 poin, dan *F1-Score Macro* meningkat 0,37 poin. Perubahan ini menunjukkan bahwa konfigurasi hasil *tuning* mengatur kembali batas keputusan sehingga kesalahan antarkelas menjadi sedikit lebih seimbang.

Penggunaan *F1-Score Macro* sebagai metrik utama diperlukan karena distribusi kelas tidak seimbang. *Entry-Level* berjumlah 2.020 data, sedangkan *Flagship* hanya 331 data. *Accuracy* lebih banyak dipengaruhi oleh kelas mayoritas, sementara *F1-Score Macro* memberikan bobot yang sama pada setiap kelas. Oleh karena itu, model *tuning* dipilih sebagai model terbaik meskipun peningkatannya relatif kecil.

Kesalahan paling banyak terjadi pada kategori *Entry-Level* dan *Mid-Range*. Label kelas dibentuk berdasarkan harga, sedangkan harga sengaja tidak digunakan sebagai fitur untuk mencegah *kebocoran data*. Dua *smartphone* dengan spesifikasi yang mirip dapat memiliki harga berbeda akibat merek, jenis *chipset*, teknologi layar, waktu rilis, fitur pengisian daya, strategi pasar, dan faktor komersial lain yang tidak tercakup dalam enam fitur penelitian ini. Tumpang tindih tersebut menjelaskan mengapa sebagian data pada batas kelas sulit dipisahkan hanya berdasarkan spesifikasi yang tersedia.

3.5 Implementasi Website

Model hasil *tuning* diimplementasikan pada *situs web* klasifikasi. Halaman *dashboard* menerima masukan tentang RAM, penyimpanan/ROM, prosesor/CPU, kapasitas baterai, ukuran layar, dan kamera utama. Halaman hasil menampilkan segmen prediksi, tingkat keyakinan, probabilitas setiap kelas, spesifikasi masukan, analisis kontribusi fitur, serta contoh *smartphone* dalam kelas yang sama. Gambar 4 adalah implementasi halaman *dashboard* dan hasil klasifikasi *smartphone*.



Gambar 4. Implementasi halaman *dashboard* dan hasil klasifikasi *smartphone*

3.6 Perbandingan dengan Penelitian Terdahulu

Penelitian ini memiliki kesamaan dengan Yakuf Al Jarkhi (2025), yaitu menggunakan spesifikasi teknis untuk mengklasifikasikan kelas harga *smartphone*. Perbedaan utama terdapat pada algoritma, proses optimasi, metrik evaluasi, dan interpretasi fitur. Ringkasan perbandingan ditunjukkan pada Tabel 5.

Tabel 5. Perbandingan penelitian terdahulu dengan penelitian ini

Aspek	Yakuf Al Jarkhi (2025)	Penelitian ini
Objek	Kisaran harga <i>smartphone</i>	Segmen <i>Entry-Level</i> , <i>Mid-Range</i> , dan <i>Flagship</i>
Algoritma	<i>Naive Bayes</i>	<i>Decision Tree C4.5</i> dan <i>Random Forest</i>
Karakteristik	Probabilistik dengan asumsi independensi fitur	<i>Gain Ratio</i> dan ensemble banyak pohon
Optimasi	Tidak dijelaskan adanya <i>tuning</i>	<i>GridSearchCV</i> dengan <i>StratifiedKfold</i>
Evaluasi	Akurasi klasifikasi	<i>Accuracy</i> , <i>Precision Macro</i> , <i>Recall Macro</i> , <i>F1-Score Macro</i> , <i>Confusion Matrix</i>
Interpretasi	Spesifikasi berpengaruh terhadap kelas harga	<i>Root node C4.5</i> dan kontribusi fitur

Implementasi	Klasifikasi <i>Naive Bayes</i>	Model <i>tuning</i> diimplementasikan pada <i>website</i>
--------------	--------------------------------	---

Nilai performa kedua penelitian tidak dapat dibandingkan secara langsung sebagai bukti keunggulan mutlak karena *dataset*, fitur, pembentukan label, pembagian data, dan prosedur evaluasinya berbeda. Kontribusi penelitian ini terletak pada pemisahan fungsi metode: C4.5 digunakan untuk interpretasi *root node*, sedangkan *Random Forest* digunakan sebagai model klasifikasi utama yang dioptimalkan dan dievaluasi pada data *hold-out*.

3.7 Keunggulan dan Keterbatasan

Keunggulan penelitian adalah penggunaan harga hanya untuk pembentukan label sehingga kebocoran target dapat dicegah; pembagian data dilakukan secara *stratified*; imputasi ditempatkan dalam *pipeline*; *tuning* hanya memanfaatkan data latih; data uji dipertahankan sebagai *hold-out*; dan pemilihan model menggunakan *F1-Score Macro*. Selain itu, C4.5 memberikan interpretasi awal terhadap atribut yang paling informatif.

Keterbatasan utama adalah penggunaan kurs tetap Rp190 per INR dan batas harga yang belum tentu merepresentasikan pasar Indonesia pada waktu yang berbeda. Model hanya menggunakan enam atribut dan belum memasukkan merek, jenis *chipset*, kecepatan prosesor, tahun rilis, *refresh rate*, teknologi layar, *fast charging*, dukungan jaringan, sistem operasi, serta fitur kamera tambahan. Ruang *hyperparameter* juga masih terbatas pada 4 kombinasi dan generalisasi model belum diuji menggunakan *dataset* eksternal.

4. Kesimpulan

Penelitian ini berhasil mengimplementasikan *Decision Tree C4.5* dan *Random Forest* untuk mengklasifikasikan *smartphone* berdasarkan spesifikasinya. *Dataset* 3.260 data dibentuk menjadi tiga kelas dan dibagi secara *stratified* menjadi 2.608 data latih serta 652 data uji. Analisis C4.5 menghasilkan kapasitas penyimpanan sebagai *root node* pada *threshold* 192 GB dengan *Information Gain* 0,2494, *Split Information* 0,6556, dan *Gain Ratio* 0,3804. *Random Forest baseline* memperoleh *Accuracy* 85,12%, *Precision Macro* 82,92%, *Recall Macro* 80,81%, dan *F1-Score Macro* 81,77%. Setelah *GridSearchCV*, *Accuracy* tetap 85,12%, sedangkan *Precision Macro* meningkat menjadi 83,44%, *Recall Macro* menjadi 81,03%, dan *F1-Score Macro* menjadi 82,15%. Konfigurasi terbaik dalam ruang pencarian yang diuji adalah 100 pohon, *max_depth* 20, *min_samples_split* 2, *min_samples_leaf* 1, *max_features* *sqrt*, dan *bootstrap=True*. Model *tuning* dipilih karena memberikan keseimbangan performa antarkelas yang lebih baik dan telah diimplementasikan pada *website* klasifikasi *smartphone*. Penelitian berikutnya perlu memperluas ruang *hyperparameter*, menggunakan data harga Indonesia yang lebih aktual, menambahkan fitur teknis dan komersial, serta melakukan validasi eksternal pada *dataset* yang berbeda.

Reference

1. Adil Setiawan, Andri Armaginda Siregar, Setiawan, N., Jalaluddin Nasution, Naufal Dhiya Putra Dalimunthe, & Farhan Sardy Abdillah. (2026). Optimasi Performa Model SVM dan Random Forest untuk Klasifikasi Kanker Payudara melalui Penyesuaian Hyperparameter. *Jurnal Komputer Teknologi Informasi Sistem Informasi (JUKTISI)*, 4(3), 2141–2149. <https://doi.org/10.62712/juktisi.v4i3.789>
2. Alvina, P., & Yamasari, Y. (2025). Model Rekomendasi Lagu Berbasis Genre Menggunakan Metode Random Forest dan Decision Tree—*Journal of Informatics and Computer Science*, 07.
3. Anugerah, A., Nurhidayatullah, M., Wijaya, P., Tamba, T., Suttan, M., Razhevva, H., Lubis, B. O., Informasi, S., & Sarana Informatika, B. (2025). SISTEM REKOMENDASI JURUSAN KULIAH BAGI CALON MAHASISWA BARU UNIVERSITAS BSI MARGONDA, FAKULTAS TEKNIK DAN INFORMATIKA, MENGGUNAKAN ALGORITMA C4.5. *Jurnal Mahasiswa Teknik Informatika*, 9(2).
4. Ayu Firmada, P., Shofwatillah, L., Rahma, F., Fauzi, F., Studi Statistika, P., Muhammadiyah Semarang, U., Kedungmundu No, J., Tembalang, K., & Tengah, J. (2025). Analisis Perbandingan Decision Tree dan Random Forest untuk Klasifikasi Penjualan Produk di Supermarket. *Emerging Statistics and Data Science Journal*, 3(1).
5. Chayy Bilqisth, S., & Ikhsanuddin, R. M. (2025). ANALISIS PERBANDINGAN AKURASI KLASIFIKASI KEPUASAN SISWA TERHADAP KINERJA GURU MENGGUNAKAN ALGORITMA SVM, C4.5, DAN RANDOM FOREST. *Jurnal Teknik Elektro Dan Informatika*, 20, 37–44.
6. Devia, E. (2023). Penerapan Decision Tree Dengan Algoritma C4.5 Untuk Menentukan Rekomendasi Kenaikan Jabatan Karyawan. *Jurnal Information System*.
7. Fakhriza, F., Subekti, D., & Winar Cahyo, P. (2025). OPTIMALISASI ALGORITMA RANDOM FOREST FEATURE SELECTION DAN PENYESUAIAN PARAMETER HIPER KLASIFIKASI GENRE MUSIK. *JIKA*, 9(1). <https://www.kaggle.com/datasets/maharshipa>

8. Gunawan, H., & Catherine. (2021). C4.5, K-Nearest Neighbor, Naïve Bayes and Random Forest Algorithms Comparison to Predict Students' On Time Graduation. *Indonesian Journal of Artificial Intelligence and Data Mining (IJAIMD)*, 4(2), 62–71. <https://doi.org/10.24014/ijaidm.v4i2.10833>
9. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2023). *An Introduction to Statistical Learning with Applications in R*, Second Edition.
10. Maisyarah, R., Azizah, N., & Hidayat, N. (2025). Tren Penelitian Klasifikasi Sentimen Pelanggan Berbasis Machine Learning: Machine Learning-Based Customer Sentiment Classification Research Trends. *Jurnal Rekayasa Lampung (JRL)*, 4(3), 2830–4640. <https://doi.org/10.23960/jrl>
11. Muchalim Danu Warta, Pramono, & Joni Maulindar. (2025). *Sistem Rekomendasi Kuliner Ikonik Kota Solo Menggu*. *Jurnal Teknologi Terpadu*.
12. Ningrum, J. C., Nilogiri, A., & A'yun, Q. (2025). Perbandingan Hasil Penerapan Metode Algoritma C4.5 Dan Random Forest Pada Penyakit Tuberculosis Di Puskesmas Jajag Comparison of the Results of the Application of C4.5 and Random Forest Algorithm Methods on Tuberculosis at the Jajag Health Center. *Jurnal Smart Teknologi*, 6(5). <http://jurnal.unmuhjember.ac.id/index.php/JST>
13. Nur Fauzi, N. P., Khomsah, S., & Putra Wicaksono, A. D. (2025). Penerapan Feature Engineering dan Hyperparameter Tuning untuk Meningkatkan Akurasi Model Random Forest dalam Klasifikasi Risiko Kredit. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 12(2), 251–262. <https://doi.org/10.25126/jtiik.2025128472>
14. Pahlevi, O., Rianto, H., & Author, C. (2025). Analisis Komparatif Model Data Mining Algoritma C4.5, CHAID, dan Random Forest untuk Penilaian Kelayakan Kredit. *Computer Science (CO-SCIENCE)*, 5(1). <http://jurnal.bsi.ac.id/index.php/co-science>
15. Ramadhani, N., Muhyil Umam, A., Syahroni, W., Rachmatullah, S., & Zumam, W. (2025). Klasifikasi Pemilihan Siswa Dalam Rekomendasi Beasiswa Al-Azar Menggunakan Algoritma C4.5 di MA Tahfidh Annuqayah. *Jurnal PROCESSOR*, 20(1). <https://doi.org/10.33998/processor.2025.20.1.2179>
16. Reynaldi, R., Faisal, I., & Chiuloto, K. (2025). OPTIMASI HYPERPARAMETER DENGAN RANDOMSEARCHCV UNTUK MENINGKATKAN AKURASI KLASIFIKASI PNEUMONIA. *Jurnal Multidisiplin Dan Sains*, 1(2). <https://jurnal.compartdigital.com/index.php/judis>
17. Setyowati, S. L., Qalbi, A., Aristawidya, R., Sartono, B., & Firdawanti, A. R. (2025). Optimizing Random Forest Parameters with Hyperparameter Tuning for Classifying School-Age KIP Eligibility in West Java. *Jambura Journal of Mathematics*, 7(1), 40–48. <https://doi.org/10.37905/jjom.v7i1.28736>
18. Ulfa, A., Winarso MKom, D., & Arribé MMSi, E. (2020). SISTEM REKOMENDASI JURUSAN KULIAH BAGI CALON MAHASISWA BARU MENGGUNAKAN ALGORITMA C4.5 (Studi Kasus: Universitas Muhammadiyah Riau). *Jurnal FASILKOM*.
19. Yakuf Al Jarkhi. (2025). Klasifikasi Kisaran Harga Smartphone Berdasarkan Spesifikasi Teknis Menggunakan Algoritma Naive Bayes. *Journal of Computer, Technology and Information Systems*.
20. Yuni, E., & Artaningsih, T. (2022). Penerapan Metode Simple Additive Weighting (SAW) Pada Sistem Rekomendasi Pemilihan Handphone. *Scientia Sacra: Jurnal Sains*, 2(2). <http://pijarpemikiran.com/index.php/Scientia>
21. Dinda Novita Sari, H. O. (2022). Penerapan Data Mining Untuk Klasifikasi Gaya Belajar Siswa Menggunakan Algoritma C4.5 Application Of Data Mining For Student Learning Style Classification Using C4.5 Algorithm. *Jurnal Smart Teknologi*. <http://jurnal.unmuhjember.ac.id/index.php/JST>
22. Michael Lauw C, H. H. (2023). Combination of Smote and Random Forest Methods for Lung Cancer Classification. *International Journal of Engineering and Computer Science Applications (IJECSA)*. DOI: 10.30812/IJECSA.v2i2.3333