

Klasifikasi Kisaran Harga Smartphone Berdasarkan Spesifikasi Teknis Menggunakan Algoritma Naive Bayes

Yakuf Al Jarkhi

Program Studi : Sistem Informasi, Universitas Darwan Ali

Email : yakufaljarkhi@gmail.com

ABSTRACT— This research aims to classify smartphone price ranges based on the technical specifications of the devices using machine learning methods. The dataset consists of features such as RAM, ROM, screen size, camera resolution, and battery capacity, with the target classification of price into three classes: low, medium, and high. The Naive Bayes classification model is used for this task. Evaluation is conducted based on accuracy metrics and attribute weighting using information gain. The results show that the Naive Bayes model achieves an accuracy of 98.65%. This study demonstrates that price classification can be accurately performed by considering technical specifications, while also identifying the most influential features in determining smartphone prices.

Keywords— price classification, smartphone, machine learning, Naive Bayes, RapidMiner.

ABSTRAK— Penelitian ini bertujuan untuk mengklasifikasikan kisaran harga smartphone berdasarkan spesifikasi teknis perangkat menggunakan metode machine learning. Dataset yang digunakan terdiri dari berbagai fitur seperti RAM, ROM, ukuran layar, resolusi kamera, dan kapasitas baterai, dengan target klasifikasi harga ke dalam tiga kelas: rendah, sedang, dan tinggi. Model klasifikasi Naive Bayes digunakan untuk tugas ini. Evaluasi dilakukan berdasarkan metrik akurasi dan pembobotan atribut menggunakan information gain. Hasil menunjukkan bahwa model Naive Bayes mencapai akurasi sebesar 98,65%. Penelitian ini membuktikan bahwa klasifikasi harga dapat dilakukan secara akurat dengan mempertimbangkan spesifikasi teknis, serta memberikan informasi mengenai fitur yang paling berpengaruh dalam menentukan harga smartphone.

Kata kunci— klasifikasi harga, smartphone, machine learning, Naive Bayes, RapidMiner.

I. PENDAHULUAN

Klasifikasi merupakan proses pengelompokan data ke dalam kelas tertentu berdasarkan karakteristik atau atribut yang ada. Kategori ini telah ditentukan sebelumnya sehingga sering disebut sebagai variabel target [1]. Dalam konteks industri smartphone, harga perangkat dapat diklasifikasikan dalam beberapa kategori, misalnya low-end, mid-range, dan high-end, atau didasarkan pada rentang harga tertentu. Proses klasifikasi memerlukan analisis mendalam terhadap atribut seperti kapasitas RAM, memori internal, performa prosesor, kualitas kamera, dan fitur tambahan lain yang memengaruhi harga.

Salah satu pendekatan yang banyak digunakan adalah pemanfaatan algoritma kecerdasan buatan (Artificial Intelligence/AI), khususnya pembelajaran mesin (machine learning) [2]. Machine learning mampu mempelajari pola dari data historis sehingga dapat memprediksi kelas harga perangkat dengan lebih akurat. Beberapa metode populer untuk tugas klasifikasi adalah Decision Trees, Support Vector Machines (SVM), Random Forest, K-Nearest Neighbors (KNN), dan Naive Bayes.

Naive Bayes menjadi pilihan utama di banyak penelitian karena kemampuannya mengolah data berukuran besar secara cepat dan efisien [3]. Algoritma ini berbasis teorema Bayes yang mengasumsikan bahwa setiap atribut bersifat independen, meskipun dalam praktiknya tidak selalu demikian. Walaupun sederhana,

Naive Bayes terbukti efektif pada berbagai kasus klasifikasi, termasuk pemrosesan teks, deteksi spam, hingga prediksi harga perangkat berbasis spesifikasi teknis.

Kelebihan lainnya adalah kebutuhan data latih yang relatif sedikit untuk menghasilkan estimasi parameter yang dibutuhkan [4]. Hal ini membuatnya sangat cocok untuk digunakan dalam situasi di mana data yang tersedia terbatas atau ketika diperlukan proses pelatihan yang cepat. Berbeda dengan beberapa algoritma lain yang memerlukan waktu komputasi lebih lama dan data dalam jumlah besar untuk mencapai akurasi tinggi, Naive Bayes dapat memberikan hasil yang memadai bahkan dengan dataset yang tidak terlalu besar.

Keunggulan lain dari algoritma Naive Bayes adalah akurasi dan kecepatannya yang tinggi saat diaplikasikan ke dalam dataset [5]. Karena perhitungannya yang sederhana dan tidak memerlukan training yang kompleks, algoritma ini mampu melakukan prediksi dengan waktu pemrosesan yang singkat. Kecepatan ini sangat menguntungkan dalam aplikasi dunia nyata, di mana hasil klasifikasi sering kali dibutuhkan secara real-time atau mendekati real-time, seperti dalam sistem rekomendasi produk atau analisis harga pasar.

Kelebihan-kelebihan ini menjadikan Naive Bayes sebagai pilihan ideal untuk klasifikasi harga smartphone berbasis spesifikasi teknis, terutama ketika dihadapkan pada kebutuhan akan hasil yang cepat dan akurat. Selain itu, algoritma ini juga mudah diimplementasikan dan

dapat dikombinasikan dengan teknik pra-pemrosesan data (data preprocessing) seperti normalisasi, feature selection, atau handling missing values untuk meningkatkan performanya. Dengan demikian, Naive Bayes tidak hanya efisien secara komputasi tetapi juga fleksibel untuk disesuaikan dengan berbagai skenario analisis data.

Dalam konteks yang lebih luas, penggunaan Naive Bayes untuk klasifikasi harga smartphone dapat diperluas dengan menggabungkannya dengan pendekatan ensemble learning atau algoritma lain untuk meningkatkan akurasi prediksi. Misalnya, menggabungkan Naive Bayes dengan Random Forest dapat membantu mengatasi keterbatasan asumsi independensi fitur dan meningkatkan ketepatan klasifikasi. Selain itu, integrasi dengan teknik deep learning juga dapat dipertimbangkan jika dataset yang digunakan sangat besar dan kompleks.

Dengan demikian, penelitian dan penerapan algoritma Naive Bayes dalam klasifikasi harga smartphone berbasis spesifikasi teknis memiliki potensi besar untuk memberikan solusi yang efisien, akurat, dan skalabel. Hal ini tidak hanya bermanfaat bagi kalangan akademisi dalam pengembangan model machine learning, tetapi juga bagi industri teknologi untuk pengambilan keputusan yang lebih data-driven dalam strategi penetapan harga dan pengembangan produk di masa depan.

II. METODOLOGI PENELITIAN

A. Dataset

Dataset yang digunakan berasal dari [sumber terbuka/Kaggle], berisi beberapa atribut real, integer dan polynominal seperti:

- 1) Brand me
- 2) Rating
- 3) RAM
- 4) ROM
- 5) Mobile size
- 6) Primary cam
- 7) Selfi cam
- 8) Battery power
- 9) Price

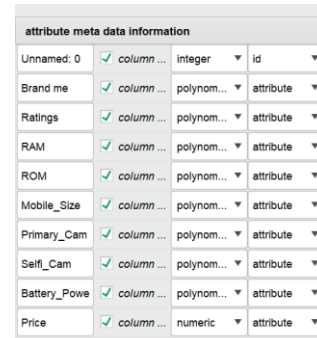
Target dari dataset adalah Price, yang dikategorikan ke dalam tiga kelas:

- 1) High > 15000
- 2) Medium <= 15000
- 3) Low <= 7000

B. Alur Proses

1) Read & Preprocessing

Tujuan dari preprocessing adalah untuk membersihkan, menormalkan, dan mempersiapkan data sehingga dapat diolah lebih efektif oleh algoritma analisis atau model machine learning [6]. Dataset dibaca melalui operator Read CSV, diikuti dengan penyesuaian attribute meta data information, tanpa label dikarenakan nantinya akan dibuat atribut baru untuk label nya.



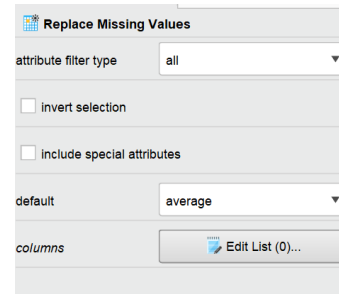
Gambar 1 attribute meta data information

Atribut Unnamed : 0 di ubah menjadi ID menggunakan operator Rename.



Gambar 2 rename attribute

Missing data di isi dengan rata-rata dikarenakan, jika data yang missing dihapus akan mempengaruhi keakuratan akurasi hasil akhir. Meskipun metode ini tergolong sederhana, pendekatan ini masih banyak digunakan karena kemudahan implementasinya serta efektivitasnya dalam mempertahankan struktur statistik data [7].



Gambar 3 replace missing values

Atribut Price di bagi menjadi 3 kategori menggunakan operator Generate Attribute menjadi Price class dengan expression atau logika, jika harga di bawah atau sama dengan 7000 maka akan di kategorika menjadi Low. Jika harga dibawah atau sama dengan 15000 maka akan dikategorikan Medium, dan jika lebih dari 15000 maka akan dikategorikan High.

Expression

```
1 if(Price <= 7000, "Low",
2 if(Price <= 15000, "Medium", "High"))
```

Gambar 4 Function description generate attribute 1

Operator Generate Attribute digunakan lagi untuk membuat atribut baru bernama Brand_phone dari atribut Brand_me yang dimana, Brand phone hanya akan menampilkan nama-nama brand smartphone nya tanpa kalimat tambahan yang ada di dalam tanda kurung dengan expression atau logika:

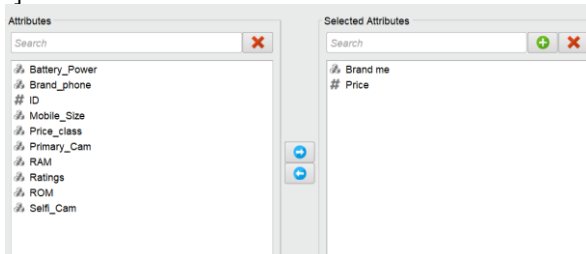
- a) \\
= Mulai dari tanda kurung
- b) .*
= Ambil semua isinya
- c) \\
= Sampai kurung tutup

d) “ = Di isi dengan kosong

```
Expression
1 replaceAll([Brand me], "\\(.*)\\", "")
```

Gambar 5 Function description generate attribute 2

Beberapa atribut di hapus menggunakan operator Select Attribut dengan type exclude attribute. Operator ini berfungsi untuk memilih atribut yang relevan yang akan digunakan dalam penelitian, sehingga hanya data yang diperlukan saja yang disertakan dalam analisis [8].



Gambar 6 Select subset

2) Data Preparation & Role Assignment

Price class dipilih menjadi label di pilih menggunakan operator Set Role.

attribute name	target role
Price_class	label

Gambar 7 Set roles

3) Split Data

Setiap model diberi data latih dan data uji dengan rasio 80:20.

ratio
0.8
0.2

Gambar 8 Partitions split data

4) Model Building

Naive Bayes adalah algoritma probabilistik berbasis Teorema Bayes dengan asumsi conditional independence antar fitur. Meskipun asumsi ini jarang terpenuhi di dunia nyata, algoritma ini tetap efektif untuk data high-dimensional. Naive Bayes merupakan salah satu metode klasifikasi yang berakar pada Teorema Bayes [9].

$$P(C|X) = \frac{P(X|C) \cdot P(C)}{P(X)}$$

Naive Bayes bekerja berdasarkan teorema Bayes, yang menghitung probabilitas suatu kelas (C) setelah mengamati data (X), dinotasikan sebagai $P(C|X)$. Probabilitas ini, juga dikenal sebagai posterior probability, menggambarkan seberapa mungkin suatu kelas C terjadi setelah memperhitungkan fitur-fitur yang diamati dalam data X . $P(X|C)$ merepresentasikan probabilitas observasi X diberikan kelas C , yang mengukur seberapa sering fitur-fitur tersebut muncul dalam kelas tertentu. Sementara itu, $P(C)$ adalah probabilitas awal (prior probability) kelas C , yang

menunjukkan distribusi kelas sebelum melihat data X . Prior probability ini biasanya dihitung berdasarkan frekuensi kemunculan kelas dalam dataset pelatihan. Terakhir, $P(X)$ adalah probabilitas marginal observasi X , yang berfungsi sebagai faktor normalisasi untuk memastikan total probabilitas seluruh kelas tetap 1. Dengan menggabungkan komponen-komponen ini, Naive Bayes mampu memprediksi kelas yang paling mungkin untuk suatu data observasi berdasarkan fitur-fiturnya.

Algoritma Naive Bayes merupakan salah satu algoritma pembelajaran induktif yang paling efektif dan efisien untuk machine learning dan data mining [10].

5) Evaluation

Performa diukur dengan metrik akurasi, presisi dan recall serta fitur penting dievaluasi dengan Weight by Information Gain untuk mencari fitur mana yang paling mempengaruhi harga smartphone pada dataset.

a) Matrik utama

- Akurasi
 Proporsi prediksi benar ($TP + TN$) dari total data. Rumusnya:

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN}$$

Akurasi merupakan metrik evaluasi yang digunakan untuk mengukur seberapa baik suatu model klasifikasi dalam memprediksi data dengan benar. Dalam rumus ini, TP (True Positive) adalah jumlah prediksi positif yang benar, TN (True Negative) adalah jumlah prediksi negatif yang benar, FP (False Positive) adalah jumlah prediksi positif yang salah, dan FN (False Negative) adalah jumlah prediksi negatif yang salah. Dengan menjumlahkan TP dan TN , kita mendapatkan total prediksi yang benar, sedangkan penyebutnya $TP + TN + FP + FN$ mewakili total prediksi yang dibuat. Akurasi memberikan gambaran proporsi prediksi yang benar secara keseluruhan, sehingga semakin tinggi nilai akurasinya, semakin baik performa model tersebut. Namun, metrik ini mungkin kurang efektif jika data tidak seimbang, karena tidak memperhitungkan distribusi kelas yang tidak merata.

- Presisi

Proporsi prediksi positif yang benar (TP) dari semua prediksi positif. Rumusnya:

$$Presisi = \frac{TP}{TP + FP}$$

Presisi merupakan metrik evaluasi yang digunakan untuk mengukur seberapa akurat model dalam memprediksi kelas positif. Dalam rumus ini, TP (True Positive) mewakili jumlah prediksi positif yang benar, sedangkan FP (False Positive) adalah jumlah prediksi positif yang salah (kelas negatif yang salah diklasifikasikan sebagai positif). Dengan membagi TP dengan jumlah total prediksi positif $TP + FP$ presisi menunjukkan proporsi prediksi positif yang benar dari semua prediksi yang diklasifikasikan sebagai positif. Nilai presisi yang tinggi mengindikasikan bahwa model jarang melakukan kesalahan dalam memprediksi kelas positif, sehingga sangat berguna dalam situasi di mana False

Positive memiliki konsekuensi yang serius, seperti dalam diagnosis medis atau deteksi spam. Namun, presisi tidak memperhitungkan False Negative, sehingga perlu digunakan bersama metrik lain seperti recall untuk evaluasi yang lebih komprehensif.

- Recall
 Proporsi kasus positif aktual yang terdeteksi dengan benar. Rumusnya:

$$Recall = \frac{TP}{TP + FN}$$

Rumus ini dihitung dengan membagi jumlah True Positives TP , yaitu kasus di mana model benar memprediksi kelas positif, dengan jumlah total instance positif yang sebenarnya $TP + FN$. False Negatives FN adalah kasus di mana model gagal mengidentifikasi instance positif. Dengan kata lain, Recall menunjukkan kemampuan model untuk menemukan semua instance yang relevan dari keseluruhan data. Nilai Recall yang tinggi menunjukkan bahwa model jarang melewatkan instance positif, sementara nilai rendah menunjukkan banyak instance positif yang tidak terdeteksi. Metrik ini sangat penting dalam situasi di mana kesalahan melewatkan instance positif memiliki konsekuensi serius, seperti dalam diagnosis medis atau deteksi kejahatan.

Menggunakan ketiga metrik ini akan membantu mengevaluasi model secara menyeluruh dan memastikan bahwa model tidak hanya akurat tetapi juga efektif [11].

b) Weight by Information Gain

- Entropy Awal
 Rumus yang diberikan adalah rumus untuk menghitung entropi $H(S)$ dalam suatu dataset S Entropi merupakan ukuran ketidakpastian atau ketidakaturan dalam dataset, yang sering digunakan dalam bidang teori informasi dan pembelajaran mesin, khususnya dalam pembuatan pohon keputusan. Rumus ini didefinisikan sebagai jumlah dari hasil kali antara probabilitas setiap kelas p_i dan logaritma basis dua dari probabilitas tersebut, yang kemudian dikalikan dengan -1.

$$H(S) = - \sum_{i=1}^n p_i \log \log_2 p_i$$

Di sini, p_i mewakili proporsi atau probabilitas kelas ke i dalam dataset S . Logaritma basis dua \log_2 digunakan karena entropi diukur dalam satuan bit. Tanda negatif di depan rumus memastikan bahwa hasil entropi selalu non-negatif, karena $\log_2 p_i$ bernilai negatif untuk $0 < p_i < 1$.

Semakin tinggi nilai entropi, semakin besar ketidakpastian atau keragaman dalam dataset. Sebaliknya, entropi rendah menunjukkan dataset yang lebih teratur atau homogen. Misalnya, jika semua contoh dalam dataset termasuk ke dalam satu kelas yang sama (misalnya, $p_i = 1$ untuk suatu kelas), maka entropinya akan bernilai 0, karena tidak ada ketidakpastian.

- Entropy setelah Split
 Entropi setelah split $H(S|A)$ adalah ukuran yang digunakan untuk menilai seberapa baik suatu fitur A dalam membagi dataset S menjadi subset-subset yang

lebih homogen. Rumus ini bekerja dengan menghitung rata-rata tertimbang dari entropi setiap subset hasil pemisahan. Secara matematis, rumusnya dinyatakan sebagai:

$$H(S|A) = \sum_{v \in Values(A)} \frac{|S_v|}{|S|} H(S_v)$$

Di mana $Values(A)$ adalah semua nilai unik dari fitur A , S_v adalah subset data yang memiliki nilai v untuk fitur A , dan $\frac{|S_v|}{|S|}$ adalah proporsi ukuran subset S_v terhadap total dataset S . Entropi setiap subset $H(S_v)$ dihitung menggunakan rumus entropi standar, yang memperhitungkan distribusi kelas dalam subset tersebut.

Nilai $H(S|A)$ yang rendah menunjukkan bahwa split berdasarkan fitur A berhasil mengurangi ketidakpastian dalam data, sehingga fitur tersebut dianggap efektif untuk pemisahan. Sebaliknya, jika nilai $H(S|A)$ mendekati entropi sebelum split, artinya fitur A kurang relevan dalam membagi data. Misalnya, dalam kasus pembuatan pohon keputusan, fitur dengan penurunan entropi terbesar (Information Gain tertinggi) akan dipilih sebagai kriteria split. Dengan demikian, rumus ini membantu dalam mengevaluasi kualitas suatu fitur untuk membangun model yang lebih akurat dan efisien.

- Information Gain

Information Gain (IG) merupakan salah satu metrik fundamental dalam pembelajaran mesin yang digunakan untuk mengukur efektivitas suatu fitur dalam mengurangi ketidakpastian saat membagi data. Konsep ini secara matematis dinyatakan sebagai selisih antara entropi awal dataset $H(S)$ dengan entropi setelah melakukan pemisahan berdasarkan fitur tertentu $H(S|A)$ dirumuskan sebagai:

$$IG(A) = H(S) - H(S|A)$$

Nilai IG yang tinggi menunjukkan bahwa fitur tersebut mampu secara signifikan menurunkan tingkat ketidakpastian dalam dataset, sehingga sangat berguna untuk proses klasifikasi. Sebaliknya, nilai IG yang rendah atau nol mengindikasikan bahwa fitur tersebut tidak memberikan kontribusi berarti dalam memisahkan data. Semakin tinggi IG , maka semakin signifikan kata tersebut dalam proses klasifikasi [12].

III. HASIL MODEL

A. Hasil Akurasi Model Naive bayes

Akurasi: 98.65%

TABEL I
 PERFORMANCE VECTOR

	True hugh	True medium	True low	Class precision
Pred high	213	3	0	98.61%
Pred medium	4	61	2	91.04%
Pred low	0	0	386	100%

Class recall 98.16% 95.31% 99.48%

Berdasarkan tabel confusion matrix di atas, dapat dijelaskan bahwa model klasifikasi yang digunakan mampu memberikan hasil prediksi yang sangat baik, terutama pada kelas Low. Hal ini terlihat dari nilai precision sebesar 100% yang menunjukkan bahwa semua data yang diprediksi sebagai Low benar-benar berasal dari kelas Low, serta recall sebesar 99,48% yang berarti hampir semua data Low berhasil dikenali dengan tepat oleh model. Pada kelas High, performa model juga tergolong sangat baik dengan nilai precision sebesar 98,61% dan recall sebesar 98,16%. Artinya, sebagian besar data yang diprediksi sebagai High memang benar berasal dari kelas tersebut, dan hampir semua data High berhasil terdeteksi.

Sementara itu, performa terendah terlihat pada kelas Medium dengan precision sebesar 91,04% dan recall sebesar 95,31%. Hal ini menunjukkan bahwa masih terdapat beberapa data kelas Medium yang salah terklasifikasi sebagai High maupun Low, sehingga akurasi pada kelas ini relatif lebih rendah dibandingkan dua kelas lainnya. Peningkatan recall yang signifikan menunjukkan bahwa model lebih baik dalam mengidentifikasi instance positif. Ini penting dalam kasus di mana false negative memiliki konsekuensi yang tinggi [13]. Secara keseluruhan, model mampu mengklasifikasikan data dengan sangat baik, dengan tingkat kesalahan yang sangat kecil terutama pada kelas Low dan High.

B. Analisis Fitur yang Paling Berpengaruh

Hasil dari Weight by Information Gain menunjukkan bahwa fitur seperti Brand phone dan Mobile size memiliki bobot tertinggi dalam menentukan kisaran harga smartphone.

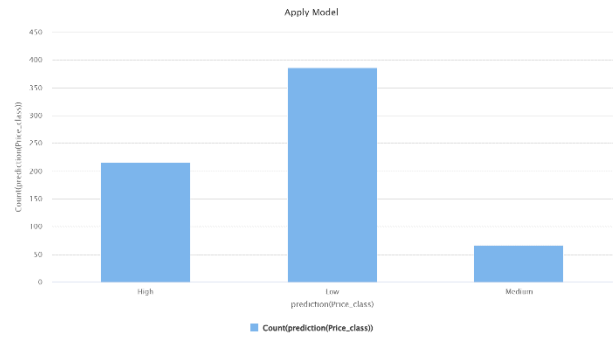
TABEL II
ATTRIBUTE WEIGHT

Attribute	Weight
Selfi cam	0.390
RAM	0.390
Primary cam	0.403
Battery power	0.580
ROM	0.591
Ratings	0.608
Mobile size	1.109
Brand phone	1.281

Citra merek memengaruhi positif pada keputusan pembelian. Artinya, makin kuat citra suatu merek, makin besar peluang pelanggan memilih smartphone tersebut [14].

C. VISUALISASI

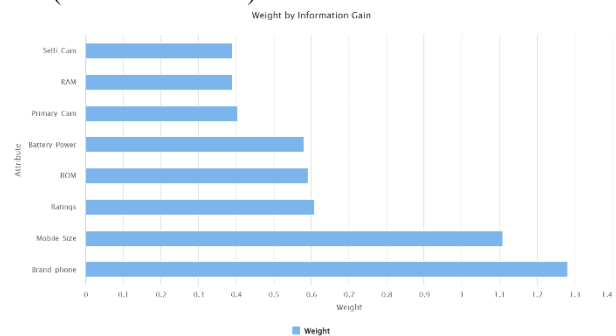
1. Distribusi Hasil Prediksi Kelas Harga Smartphone (Low, Medium, High)



Gambar 9 Bar chart prediksi price class

Grafik batang tersebut menunjukkan distribusi hasil prediksi kelas harga smartphone (Low, Medium, High) yang diperoleh dari model Naive Bayes. Pada grafik terlihat bahwa kelas Low memiliki jumlah prediksi paling banyak, yaitu mendekati 400 data. Kelas High berada di urutan kedua dengan jumlah sekitar 210 data, sedangkan kelas Medium merupakan yang paling sedikit dengan jumlah prediksi hanya sekitar 67 data. Distribusi ini menggambarkan bahwa mayoritas smartphone dalam dataset diprediksi masuk ke kategori harga rendah, sementara hanya sebagian kecil yang termasuk kategori harga sedang. Hasil ini konsisten dengan distribusi data pada label aslinya dan menunjukkan bahwa model mampu memetakan kelas dengan cukup baik.

2. Bobot Fitur terhadap Klasifikasi Harga Smartphone (Information Gain)



Gambar 10 Bar chart bobot fitur berdasarkan information gain

Grafik batang di atas menunjukkan hasil pembobotan atribut (feature importance) menggunakan metode Weight by Information Gain pada dataset smartphone. Dari grafik terlihat bahwa Brand phone memiliki bobot paling tinggi, yaitu sekitar 1,28, diikuti oleh Mobile size dengan bobot 1,10. Fitur-fitur lain seperti Ratings, ROM, Battery power, Primary cam, RAM, dan Selfi cam memiliki bobot yang lebih rendah dengan nilai di bawah 0,7. Hal ini menunjukkan bahwa atribut Brand phone dan Mobile size adalah dua faktor yang paling berpengaruh dalam menentukan klasifikasi kelas harga smartphone (Low, Medium, High). Sementara itu, fitur lain tetap berkontribusi tetapi dengan pengaruh yang relatif lebih kecil terhadap hasil prediksi model. Bar charts di pilih karena lebih cocok untuk membaca representasi data numerik[15].

IV. KESIMPULAN

Penelitian ini menunjukkan bahwa klasifikasi kisaran harga smartphone dapat dilakukan secara akurat menggunakan algoritma machine learning. Model Naive Bayes sangat relevan untuk klasifikasi cepat. Identifikasi fitur penting memberikan wawasan praktis bagi produsen maupun pengguna dalam memahami kontribusi spesifikasi terhadap harga perangkat smartphone.

V. REFERENSI

- [1] H. F. Putro, R. T. Vuldari, and W. L. Y. Saptomo, "Penerapan Metode Naive Bayes Untuk Klasifikasi Pelanggan," *J. Teknol. Inf. dan Komun.*, vol. 8, no. 2, 2020, doi: 10.30646/tikomsin.v8i2.500.
- [2] T. H. Salsabila, T. M. Indrawati, and R. A. Fitri, "Meningkatkan Efisiensi Pengambilan Keputusan Publik melalui Kecerdasan Buatan," *J. Internet Softw. Eng.*, vol. 1, no. 2, p. 21, 2024, doi: 10.47134/pjise.v1i2.2401.
- [3] Burhanuddin and Edi Yusuf, "Penerapan Model Klasifikasi dalam Kelayakan Pemilihan Bangunan Rumah KPR berbasis Naive Bayes," *Sisfo J. Ilm. Sist. Inf.*, vol. 9, no. 1, pp. 38–50, 2025, doi: 10.29103/sisfo.v9i1.22139.
- [4] I. Alwiah Musdar, "Aplikasi Prediksi Kerusakan Smartphone Menggunakan Metode Naive Bayes Dan Laplace Smoothing," *Jtriste*, vol. 5, no. 2, pp. 8–16, 2018, [Online]. Available: <https://www.statista.com/statistics/262205/market-share-held-by->
- [5] M. F. Nugroho and S. Wibowo, "Fitur Seleksi Forward Selection Untuk Menentukan Atribut Yang Berpengaruh Pada Klasifikasi Kelulusan Mahasiswa Fakultas Ilmu Komputer UNAKI Semarang Menggunakan Algoritma Naive Bayes," *J. Inform. Upgris*, vol. 3, no. 1, pp. 63–70, 2017, doi: 10.26877/jiu.v3i1.1669.
- [6] G. S. K. Ranjan, A. Kumar Verma, and S. Radhika, "K-Nearest Neighbors and Grid Search CV Based Real Time Fault Monitoring System for Industries," *2019 IEEE 5th Int. Conf. Conver. Technol. I2CT 2019*, vol. 4, no. January, pp. 273–281, 2019, doi: 10.1109/I2CT45611.2019.9033691.
- [7] U. Ardhita, N. Sulistianingsih, and A. Rahman, "Penanganan Missing Data Hujan Kabupaten Sumbawa," vol. 5, no. 1, pp. 39–44, 2025.
- [8] F. Ramadhan, "Implementasi Algoritma Fp-Growth Untuk Analisis Pola Asosiasi Penjualan Pada Mini Resto Juragan Nasbak," *J. Inform. dan Tek. Elektro Terap.*, vol. 13, no. 2, 2025, doi: 10.23960/jitet.v13i2.6188.
- [9] T. Imandasari, E. Irawan, A. P. Windarto, and A. Wanto, "Algoritma Naive Bayes Dalam Klasifikasi Lokasi Pembangunan Sumber Air," *Pros. Semin. Nas. Ris. Inf. Sci.*, vol. 1, no. September, p. 750, 2019, doi: 10.30645/senaris.v1i0.81.
- [10] G. M. C. Batubara, A. Desiani, and A. Amran, "Klasifikasi Jamur Beracun Menggunakan Algoritma Naive Bayes dan K-Nearest Neighbors," *J. Ilmu Komput. dan Inform.*, vol. 3, no. 1, pp. 33–42, 2023, doi: 10.54082/jiki.68.
- [11] N. H. Setyawan and N. Wakhidah, "Analisis Perbandingan Metode Logistic Regression, Random Forest, Gradient Boosting Untuk Prediksi Diabetes," *JUPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.)*, vol. 10, no. 1, pp. 150–162, 2025, doi: 10.29100/jupi.v10i1.5743.
- [12] A. E. Sari, S. Widowati, and K. M. Lhaksana, "Klasifikasi Ulasan Pengguna Aplikasi Mandiri Online di Google Play Store dengan Menggunakan Metode Information Gain dan Naive Bayes Classifier," *e-Proceeding Eng.*, vol. 6, no. 2, pp. 9143–9157, 2019, [Online]. Available: <https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/viewFile/9933/9790>
- [13] M. Gilang Saputra and B. Jati Santoso, "Implementation of Feature Selection Using Boruta to Improve the Accuracy of the Lapsar Prediction Model," vol. 5, no. July, pp. 886–895, 2025.
- [14] M. O. Ayuningsih and D. P. Widyaningtyas, "Pengaruh Kualitas Produk, Citra Merek, Dan Harga Terhadap Keputusan Pembelian Smartphone Merek Iphone Di Kota Semarang," *J. Ilm. Bid. Sos. Ekon. Budaya, Teknol. dan Pendidik.*, vol. 3, no. 2, pp. 104–128, 2025, [Online]. Available: [file:///C:/Users/USER/Downloads/276 \(1\).pdf](file:///C:/Users/USER/Downloads/276%20(1).pdf)
- [15] F. Alfiah et al., *Pemodelan dan Visualisasi Data*, no. June. 2025. [Online]. Available: https://books.google.co.id/books?id=1VxnEQAAQBAJ&newbks=0&printsec=frontcover&pg=PA2&dq=visualisasi+data+adalah&hl=id&source=newbks_fb&redir_esc=y#v=onepage&q=visualisasi+data+adalah&f=false