

PREDIKSI PENJUALAN TOKO SNACK MENGGUNAKAN MODEL *FORECASTING NAÏVE BAYES* : STUDI KASUS PADA [TOKO SNACK IDA]

Muhammad Haikal Adzim^{1*}, Ghufron²

Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Sultan Agung
Semarang^{1,2}

[1haikalikal053@gmail.com](mailto:haikalikal053@gmail.com) , [2ghufron@std.unissula.ac.id](mailto:ghufron@std.unissula.ac.id)

Received: 02-04- 2025

Revised: 20-04-2025

Approved: 28-04-2025

ABSTRAK

Toko Snack Ida di Pasar Comal, Kabupaten Pemalang, menghadapi kendala dalam mengelola stok barang. Untuk mengatasi permasalahan tersebut, penelitian ini menerapkan metode naïve bayes sebagai pendekatan forecasting guna mengklasifikasikan pola penjualan produk snack. Tujuan dari implementasi metode naïve bayes dalam penelitian ini adalah untuk membangun sistem yang mampu memprediksi kategori penjualan produk seperti laris atau kurang laris berdasarkan pola historis penjualan yang dianalisis dari variabel – variabel seperti nama produk, rasa, jenis, kemasan, bulan dan tahun. Berdasarkan data historis sebanyak 1.620 entri, model naïve bayes dilatih menggunakan 1.296 data latih dan diuji pada 324 data uji. Hasil pengujian menunjukkan bahwa model berhasil mengklasifikasikan 123 data sebagai kategori laris dan 201 data kategori kurang laris. Hasil penelitian ini menunjukkan performa evaluasi model dengan menghasilkan akurasi sebesar 0,59, presisi 0,59, recall 0,59 dan f1-score 0,58.

Kata kunci : prediksi penjualan, naïve bayes, toko snack

PENDAHULUAN

Industri makanan ringan atau snack memiliki peran yang *signifikan* dalam pasar *ritel*, dengan pertumbuhan yang terus meningkat seiring dengan perubahan gaya hidup dan *preferensi* konsumen. Di tengah persaingan yang semakin ketat, manajemen toko *snack* memerlukan strategi yang efektif untuk mengoptimalkan penjualan, mengelola stok dengan baik, dan meningkatkan kepuasan pelanggan. Sistem prediksi merupakan sistem yang digunakan untuk kegiatan prediksi atau memperkirakan segala sesuatu yang akan terjadi. Prediksi digunakan sebagai informasi, ulasan dan pengetahuan berdasarkan faktor-faktor yang mempengaruhi maupun berdasarkan fakta yang ada [1]. *Forecasting* digunakan untuk memprediksi jumlah barang yang akan dibutuhkan dimasa mendatang berdasarkan pengalaman penjualan barang dimasa lampau [2]. *Forecasting* adalah prosedur untuk menghasilkan informasi faktual mengenai situasi sosial di masa depan berdasarkan informasi yang sudah tersedia mengenai masalah kebijakan [3].

Forecasting atau peramalan stok barang memiliki potensi untuk mempengaruhi tingkat kelancaran penjualan barang tersebut, karena membantu dalam merencanakan produksi, pengadaan, dan strategi pemasaran yang lebih tepat sesuai dengan permintaan pasar yang diperkirakan. Prediksi adalah suatu proses *forecasting* yang diterapkan untuk memperkirakan suatu variabel di periode waktu mendatang, berdasarkan hasil data dari periode waktu sebelumnya [4]. *Naïve Bayes* adalah metode klasifikasi probabilistik yang sederhana, di mana probabilitas diperkirakan dengan menghitung *frekuensi* dan kombinasi nilai dari data yang ada dalam dataset yang diberikan [5]. Klasifikasi merupakan proses pengelompokkan dan prediksi data baru berdasarkan atribut atau variabel dengan acuan dari data yang sudah tersedia. Dalam klasifikasi, terdapat dua jenis data yaitu data latih (*training data*) dan data uji (*testing*

data) [6]. Ditemukan oleh ilmuwan inggris, *thomas bayes*, metode ini digunakan untuk memperkirakan probabilitas kejadian dimasa depan berdasarkan data masa lalu [7]. Dengan menggunakan metode *Naïve Bayes* untuk meramalkan penjualan stok snack, pengelola dapat memperoleh pemahaman yang lebih baik mengenai seberapa laku stok tersebut dipasaran.

Toko Snack Ida dibangun pada tahun 2002. Toko snack ida menghadapi permasalahan dalam pengelolaan barang yang kurang stabil, sehingga sering kali mengalami kesulitan dalam menentukan jumlah stok yang optimal. Untuk mengatasi hal tersebut, penelitian ini berencana mengembangkan sebuah aplikasi prediksi yang dapat menganalisis data penjualan dan memberikan estimasi mengenai tingkat keberhasilan penjualan suatu produk, apakah termasuk kategori laris atau kurang laris. dengan adanya aplikasi ini, diharapkan pengelolaan stok dapat lebih efektif sehingga mengurangi resiko kelebihan atau kekurangan barang serta meningkatkan efesiensi operasional toko. Dalam metode ini, penulis terlibat langsung dalam aktivitas sehari – hari untuk mengamati objek penelitian, sambil mencatat berbagai kejadian yang terjadi di pt. tunas dwipa matra way kanan [8]. Mengumpulkan data untuk melatih dan menguji model *naïve bayes* sangat penting. Semakin banyak data yang tersedia, semakin banyak baik model dalam menyelesaikan permasalahan[9]. Algoritma *naïve bayes* adalah klasifikasi data yang bekerja secara efektif dengan memaksimalkan pengawasan dalam estimasi probabilitas secara akurat, berdasarkan asumsi bahwa setiap atribut bersifat *independen* secara kondisional terhadap atribut lainnya jika nilai *output* telah diketahui[10]. Proses pemodelan bertujuan untuk membangun pola dan menguji model *naïve bayes* dengan membagi dataset menjadi dua yaitu data *training* dan data *testing* dengan proporsi 80:20, dimana 80% digunakan untuk pelatihan dan 20% untuk pengujian model[11]. Algoritma *naïve bayes* digunakan untuk mengelompokkan ulasan produk ke dalam tiga kategori sentimen, yaitu positif, negatif dan netral. Evaluasi kinerja model menunjukkan bahwa tingkat akurasi yang mencapai adalah 75,90%. Selain itu, nilai presisi sebesar 66,45% dan recall sebesar 97,42% juga menjadi aspek penting yang diperhatikan[12].

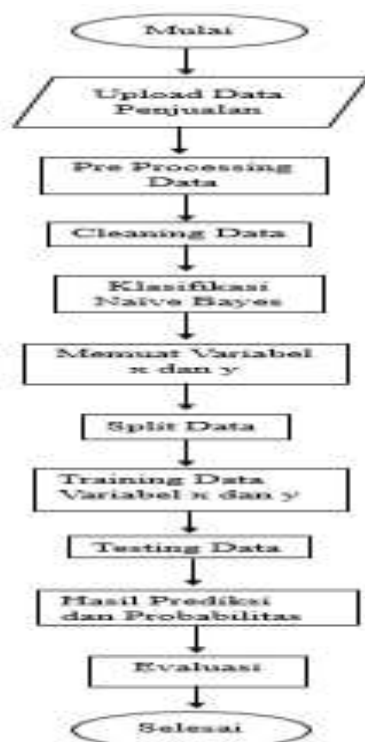
Tujuan dari pengembangan aplikasi prediksi ini adalah untuk mengembangkan model prediksi penjualan snack menggunakan metode *naïve bayes* serta menerapkan pada toko snack ida guna menganalisis *tren* penjualan berdasarkan data *historis* dari tahun 2017 hingga mei 2024, meningkatkan keuntungan dengan menyesuaikan jumlah produk yang tersedia sesuai dengan permintaan konsumen. Pengumpulan data penjualan makananan ringan dari januari hingga desember dalam rentang tahun 2018 – 2021 menjadi sumber informasi yang berharga untuk analisis. Dengan menerapkan algoritma *naïve bayes* yang merupakan salah satu metode umum dalam analisis *prediktif*, toko ini dapat memperkirakan harga penjualan makanan ringan dengan lebih akurat[13]. *Website* adalah sekumpulan halaman *web* yang saling terkait dan diakses melalui internet menggunakan *browser web*. *Streamlite* merupakan *web application framework* yang dirancang untuk mempermudah *developer* dalam mengembangkan dan membangun aplikasi web, khususnya dibidang *machine learning* [14]. *Python* juga mendukung paradigma pemrograman berorientasi objek (*object oriented programming*) dan memiliki semantik dinamis yang meningkatkan keterbacaan sintaksnya[15]. *Numpy* adalah salah satu pustaka dalam bahasa pemrograman *python* yang digunakan untuk melakukan berbagai perhitungan matematis, seperti aljabar, statistika dan perhitungan kompleks lainnya[16].

Manfaat dari penelitian ini membantu pemilik toko dalam mengoptimalkan

strategi penjualan dengan menganalisis data penjualan untuk mengidentifikasi produk yang berpotensi laris dan kurang laris. Dengan demikian, pemilik toko dapat mengelola stok secara lebih efisien, mengurangi resiko kelebihan atau kekurangan barang serta meningkatkan keuntungan melalui perencanaan yang lebih tepat berdasarkan pola permintaan konsumen. Pada langkah ini akan dilakukan analisis dan pengumpulan data yang dibutuhkan dalam pembangunan *system*, agar *system* yang dibangun sesuai dengan kebutuhan pengguna [17]. Hasil penelitian dari rata-rata perkembangan volume penjualan sejumlah 1.025 unit maka melalui analisis rasionalisasi peralaman, diketahui *forecasting* volume penjualan tahun 2017 adalah 913 unit, sedangkan metode yang tepat untuk digunakan oleh Dealer Cahaya Motor Indah Muara Tembesi dalam melakukan peramalan adalah metode *least square* dan metode *trend moment*, di mana rata-rata kesalahan peramalan dari trend moment adalah 483,92 [18].

Sistem ini dirancang untuk membantu pemilik toko dalam mengelola penjualan serta memprediksi tren penjualan pada tahun mendatang yaitu 2025. Pengembangan ini dilakukan menggunakan *framework streamlite* yang dirancang khusus untuk keperluan analisis sentimen[19]. Proses prediksi produksi karet akan memanfaatkan teknologi data mining menggunakan Algoritma *Naïve Bayes*. Algoritma *Naive Bayes* merupakan salah satu metoda pembelajaran mesin yang memanfaatkan perhitungan probabilitas dan statistik dengan memprediksi probabilitas di masa depan berdasarkan pengalaman di masa sebelumnya [20]. Selain itu, sistem ini dapat digunakan oleh pengguna untuk mengelola toko dengan lebih efektif serta memantau laporan penjualan secara sistematis.

METODE PENELITIAN



Gambar 2. 1 Flowchart Sistem

Dalam sistem aplikasi penjualan, data diinputkan ke dalam sistem. Tahap pertama dalam pemrosesan adalah menghapus kolom yang tidak diperlukan. Setelah

itu, data melakukan proses pembersihan yaitu mengubah data berbentuk teks menjadi format *numerik*. Selanjutnya data diklasifikasikan menggunakan algoritma *naïve bayes*. Pada tahap ini, dilakukan persiapan variabel *x* dan variabel *y*. setelah variabel disiapkan, data dibagi menjadi dua bagian yaitu data latih dan data uji. Data latih digunakan untuk melatih model pada variabel *x* dan *y*. setelah proses pelatihan selesai, model diuji menggunakan data uji untuk menghasilkan prediksi serta probabilitas dari data yang diuji. Tahap terakhir adalah evaluasi hasil prediksi untuk menilai performa model. Setelah melakukan evaluasi proses prediksi telah selesai.

Implementasi Algoritma Naïve Bayes

Dalam penelitian ini, proses pengolahan data untuk membangun model prediksi penjualan dilakukan menggunakan *google colab* yang berfungsi sebagai platform untuk melatih dan menguji guna memperoleh hasil prediksi.

1. Data Penjualan

Penelitian ini menggunakan data penjualan snack ditoko ida dari tahun 2017 hingga mei 2024. Data tersebut diperoleh melalui wawancara dengan pemilik toko. Jumlah data yang digunakan dalam penelitian ini adalah 1.620 *entri* dengan 11 kolom. Data yang digunakan dalam analisis untuk memprediksi berasal dari produk aoka.

2. Preprocessing Data

Preprocessing data merupakan tahap awal dalam analisis data yang bertujuan untuk menyiapkan data agar dapat digunakan dalam model prediksi. Model *development* (pengembangan model) dimana proses ini membangun, melatih dan mengoptimalkan model *meachine learning* menggunakan data yang telah diproses. Tujuan model *development* membuat model yang dapat memprediksi penjualan snack dengan akurat, mengoptimalkan performa model sebelum digunakan aplikasi, memastikan model dapat bekerja dengan baik pada data. Proses ini mencakup pembersihan data dengan menghapus kolom yang tidak diperlukan, seperti no, isi, jumlah produk dan harga produk.

- a. Kolom no tidak digunakan karena hanya berfungsi sebagai nomor urut data dan tidak memiliki pengaruh terhadap prediksi penjualan.
- b. Kolom isi tidak digunakan karena tidak memberikan informasi langsung yang dapat digunakan dalam model klasifikasi. Model lebih berfokus pada fitur yang memiliki korelasi kuat dengan hasil prediksi.
- c. Kolom jumlah produk tidak digunakan karena sudah terdapat atribut lain yang memiliki kesamaan, seperti, jumlah penjualan. Penghapusan kolom ini bertujuan untuk menghindari redudansi data yang dapat menyebabkan bias dalam model.
- d. Kolom harga produk tidak digunakan karena dalam konteks prediksi, harga bukan merupakan faktor utama dalam menentukan kategori penjualan. Farktor ini lebih berkaitan dengan strategi bisnis yang berada di luar cakupan prediksi.

Selanjutnya data yang relevan, seperti nama produk, rasa, jenis, kemasan, bulan, tahun, dan jumlah penjualan, dipersiapkan untuk diproses dalam sistem. Faktor – faktor yang mempengaruhi penjualan seperti nama produk, rasa, jenis, kemasan, bulan, tahun, dan jumlah penjualan akan dianalisis untuk melihat pengaruhnya terhadap tingkat penjualan. Setelah itu, model *naïve bayes* dilatih dengan mempertimbangkan faktor – faktor yang mempengaruhi penjualan, kemudian diuji menggunakan data testing untuk mengevaluasi kinerja model berdasarkan akurasi, *confusion matrix*, dan *clasification report*.

```
[ ] def drop_data():
    print("Proses drop kolom yang tidak diperlukan")
    global data
    # Periksa apakah kolom ada sebelum dihapus
    columns_to_drop = ['No', 'Jenis', 'Jumlah Produk', 'Harga Produk']
    existing_columns = data.columns
    columns_to_drop = [col for col in columns_to_drop if col in existing_columns]

    data = data.drop(columns=columns_to_drop)
    import time
    time.sleep(10)
    print("Drop Berhasil")
    time.sleep(5)
    print("Proses selesai")

# Periksa apakah kolom ada sebelum dihapus dari fungsinya juga
columns_to_drop = ['No', 'Jenis', 'Jumlah Produk', 'Harga Produk']
existing_columns = data.columns
columns_to_drop = [col for col in columns_to_drop if col in existing_columns]

data = data.drop(columns=columns_to_drop)
time.sleep(10)
print("Drop Berhasil")
time.sleep(5)
print("Proses selesai")
```

Gambar 2. 2 Drop Kolom

Pada gambar 2. 2 kode tersebut berfungsi untuk menghapus kolom yang tidak diperlukan dari sebuah *dataframe* bernama *data*, dengan terlebih dahulu memeriksa keberadaan kolom yang akan dihapus agar tidak terjadi *error*. Selain itu, terdapat jeda waktu menggunakan *time.sleep()* untuk memberikan simulasi proses sebelum mencetak pesan keberhasilan dan penyelesaian proses.

3. Cleaning Data

Cleaning data adalah proses pembersihan, memperbaiki dan mengorganisir data agar siap digunakan untuk analisis atau pemodelan. Dalam proses ini, bertujuan untuk mengubah data kategorikal menjadi *numerik* agar dapat digunakan dalam model prediksi. Untuk data yang akan diubah ke dalam *numerik* yaitu nama produk, rasa, jenis, kemasan, bulan, tahun dan jumlah penjualan.

```
from IPython.display import HTML
def cleaning_data():
    global data
    nam = data['nama_produk']
    nam = nam.drop_duplicates()
    nam = [nam for nam in nam]
    # Mengurangi huruf (1, len(nam) + 1) untuk memastikan jumlah nilai yang benar
    l = [1 for i in range(1, len(nam) + 1)]
    data['nama_produk'] = data['nama_produk'].replace(to_replace=nam, value=l)
    time.sleep(10)
    print("Cleaning data nama produk berhasil")

    ras = data['rasa']
    ras = ras.drop_duplicates()
    ras = [ras for ras in ras]
    # Mengurangi huruf (1, len(ras) + 1) untuk memastikan jumlah nilai yang benar
    l = [1 for i in range(1, len(ras) + 1)]
    data['rasa'] = data['rasa'].replace(to_replace=ras, value=l)
    time.sleep(10)
    print("Cleaning Data Rasa Berhasil")

    jen = data['jenis']
    jen = jen.drop_duplicates()
    jen = [jen for jen in jen]
    # Mengurangi huruf (1, len(jen) + 1) untuk memastikan jumlah nilai yang benar
    l = [1 for i in range(1, len(jen) + 1)]
    data['jenis'] = data['jenis'].replace(to_replace=jen, value=l)
    time.sleep(10)
    print("Cleaning Data jenis Berhasil")

    kem = data['kemasan']
    kem = kem.drop_duplicates()
    kem = [kem for kem in kem]
    # Mengurangi huruf (1, len(kem) + 1) untuk memastikan jumlah nilai yang benar
    k = [k for k in range(1, len(kem) + 1)]
    data['kemasan'] = data['kemasan'].replace(to_replace=kem, value=k)
    time.sleep(10)
    print("Cleaning Data kemasan Berhasil")

    bul = data['bulan']
    bul = bul.drop_duplicates()
    bul = [bul for bul in bul]
    # Mengurangi huruf (1, len(bul) + 1) untuk memastikan jumlah nilai yang benar
    l = [1 for i in range(1, len(bul) + 1)]
    data['bulan'] = data['bulan'].replace(to_replace=bul, value=l)
    time.sleep(10)
    print("Cleaning Data bulan berhasil")

    # Validasi kolom 'tahun'
    if 'tahun' in data.columns:
        tab = data['tahun'].drop_duplicates().reset_index(drop=True)
        tab_mapping = {v: m + 1 for m, v in enumerate(tab)}
        data['tahun'] = data['tahun'].map(tab_mapping)
        time.sleep(2)
        print("Cleaning Data tahun Berhasil")

    # Validasi kolom 'jumlah_penjualan'
    if 'jumlah_penjualan' in data.columns:
        # Perbaiki jika penjumlahan 0 atau 1
        data['jumlah_penjualan'] = data['jumlah_penjualan'].replace({
            'Laris': 1,
            'Kurang Laris': 0
        })
        time.sleep(2)
        print("Cleaning Data jumlah penjualan berhasil")

    time.sleep(1)
    print("Proses selesai")
```

Gambar 2. 3 Cleaning Data

Gambar 2. 3 kode ini melakukan proses *cleaning* data dengan mengganti nilai unik dalam beberapa kolom kategorikal dengan angka. Langkah – langkahnya adalah

sebagai berikut :

- Menghapus duplikasi pada setiap kolom kategorikal seperti nama produk, rasa, jenis, kemasan, bulan dan tahun.
- Mengganti nilai unik dalam kolom dengan angka berurutan.
- Mengonversi jumlah penjualan dari kategori laris "1" dan kurang laris "2".
- Menambah jeda waktu (*time.sleep*) setelah setiap langkah untuk memberi waktu pemrosesan.
- Menampilkan pesan keberhasilan untuk setiap tahap *cleaning*.

4. Klasifikasi Naïve Bayes

Tahap awal dalam klasifikasi menggunakan algoritma *naïve bayes* dimulai dengan mengimpor pustaka yang diperlukan, yaitu *gaussiannb* dari *sklearn.naive.bayes*.

```
#import algoritma naive bayes
from sklearn.naive_bayes import GaussianNB

#import model gaussian
modelnb = GaussianNB()
```

Gambar 2. 4 Klasifikasi *Naive Bayes*

Gambar 2. 4 kode ini mengimpor kelas *gaussiannb* dari modul *sklearn.naive_bayes*, yang merupakan implementasi algoritma *naïve bayes* dengan distribusi *gaussian*. Selanjutnya, membuat sebuah *instance* dari model *gaussiannb* dengan nama *modelnb*, yang artinya dapat digunakan untuk pelatihan dan prediksi data.

5. Memuat variabel x dan y

Siapkan variabel *x* yaitu nama produk, rasa, jenis, kemasan, bulan dan tahun serta variabel *y* yaitu jumlah penjualan. Kemudian muat data variabel *independen* ke dalam variabel *x*. selanjutnya, muat data variabel *dependen*, yaitu jumlah penjualan yang dikategorikan sebagai laris atau kurang laris, ke dalam variabel *y*.

```
#memuat data variabel independen ke dalam variabel x
x = data.drop(["Jumlah Penjualan"], axis=1) # Ubah 'sumbu' menjadi 'sumbu'
print(x)
```

Gambar 2. 5 Memuat Variabel *x*

Gambar 2. 5 kode tersebut memuat data variabel *independen* ke dalam variabel *x* dengan menghapus kolom "jumlah penjualan" dari *dataframe* data pada sumbu kolom (*axis=1*) dan kemudian mencetak hasilnya.

```
# memuat data variabel dependen, yaitu variabel jumlah penjualan dengan laris atau kurang laris ke dalam variabel x
y = data["Jumlah Penjualan"]
print(y)
```

Gambar 2. 6 Memuat Variabel *Dependen*

Gambar 2. 6 kode tersebut memuat variabel *dependen* "jumlah penjualan" dari *dataframe* data ke dalam variabel *y*, yang berisi kategori seperti laris atau kurang laris, lalu mencetak hasilnya.

6. Split Data

Setelah memuat data variabel *x* dan *y*, langkah selanjutnya adalah mengimpor

library machine learning scikit-learn, yang akan digunakan untuk proses pelatihan dan pengujian model. Setelah itu, data x dan y dibagi menjadi dua bagian, yaitu data *training* untuk melatih model dan data *testing* untuk menguji akurasi model dalam melakukan prediksi. Dalam pembagian data ini, proses dilakukan secara acak sebanyak 80% data digunakan untuk pelatihan, sementara 20% data digunakan untuk pengujian.

```
#import library machine learning sklearn  
from sklearn.model_selection import train_test_split
```

Gambar 2. 7 Library Sklearn

Gambar 2. 7 kode tersebut mengimpor modul *train_test_split* dari *scikit-learn* (*sklearn*), yang digunakan untuk membagi dataset menjadi data pelatihan (*training*) dan data pengujian (*testing*) dalam proses *meachine learning*.

```
#meload data x dan y ke dalam data training dan juga data test  
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, random_state = 123)
```

Gambar 2. 8 Split Data

Gambar 2. 8 kode tersebut membagi dataset menjadi data pelatihan (x_{train} , y_{train}) dan data pengujian (x_{test} , y_{test}) menggunakan fungsi *train_test_split()*, dimana 20% data digunakan untuk pengujian ($test_size=0.2$), sementara 80% digunakan pelatihan. parameter $random_state=123$ memastikan bahwa pembagian data selalu konsisten setiap kali kode dijalankan.

7. Training Data variabel x dan y

Data yang digunakan untuk melatih model terdiri dari variabel x dan y , yang dipilih secara acak. Sebanyak 80% dari total data, yaitu $0,8 * 1.620 = 1.296$ data, digunakan sebagai data training untuk proses pembelajaran model.

```
x_train.head(5)
```

Gambar 2. 9 X train

Gambar 2. 9 kode x_{train} berfungsi untuk menampilkan lima baris pertama dari *dataframe* x_{train} , yang merupakan bagian dari data pelatihan setelah pembagian dataset menggunakan *train_test_split()*.

```
y_train.head(5)
```

Gambar 2. 10 Y Train

Gambar 2. 10 kode y_{train} digunakan untuk menampilkan lima baris pertama dari *dataframe* atau *series* y_{train} , yang berisi data target (*label*) dari data pelatihan setelah pembagian dataset menggunakan *train_test_split()*.

8. Testing Data

Data yang digunakan untuk pengujian dipilih secara acak sebanyak 20% dari total data. Sebanyak $0,2 * 1.620 = 324$ data digunakan sebagai data *testing* untuk mengevaluasi apakah model dapat memprediksi kategori laris atau kurang laris dengan akurat.

```
x_test.head(5)
```

Gambar 2. 11 x Test

Gambar 2. 11 kode x_{test} digunakan untuk menampilkan lima baris pertama dari *dataframe* x_{test} , yang berisi data uji untuk variabel *independen* setelah proses pembagian dataset menggunakan *train_test_split()*.

9. Output prediksi dan probabilitas

Hasil prediksi pada *testing* berupa nilai probabilitas yang menunjukkan kemungkinan suatu data masuk ke dalam kategori laris atau kurang laris. pada proses pengujian ini, sebanyak 324 data digunakan untuk mengukur sejauh mana model dapat melakukan prediksi dengan akurat berdasarkan pola yang telah dipelajari dari data *training*.

```
#prediksi
y_pred = nbtrain.predict(x_test)
y_pred
```

Gambar 2. 12 Kode Prediksi

Gambar 2. 12 kode ini menggunakan model *naïve bayes* yang telah dilatih (*nbtrain*) untuk memprediksi nilai berdasarkan data uji (*x_test*) dan menyimpan hasilnya dalam variabel *y_pred*. setelah itu, *y_pred* akan menampilkan hasil prediksi tersebut.

```
array([[2, 2, 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 1, 2, 2, 2, 1, 2, 2, 2, 1, 2, 2,
2, 1, 1, 2, 1, 1, 1, 1, 2, 1, 2, 2, 2, 1, 2, 1, 2, 2, 1, 2, 2, 1, 2, 2,
2, 1, 1, 2, 2, 1, 2, 2, 2, 1, 2, 1, 2, 1, 2, 2, 1, 1, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 1, 1, 1, 2, 1, 1, 2, 1, 1, 2,
2, 2, 2, 1, 1, 1, 2, 2, 1, 2, 1, 2, 2, 2, 1, 2, 2, 2, 1, 2, 2, 1, 1,
2, 1, 1, 2, 2, 1, 2, 3, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1,
1, 1, 2, 1, 2, 2, 1, 1, 2, 2, 2, 1, 2, 2, 2, 2, 2, 1, 2, 2, 1, 2,
2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 1, 1, 2, 2, 2, 2, 1, 2, 2, 1, 1, 2,
2, 2, 2, 1, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 1, 2, 2, 1, 1, 2, 1, 2,
1, 2, 2, 2, 2, 3, 1, 2, 2, 2, 1, 2, 2, 2, 1, 2, 1, 1, 2, 1, 1, 2,
2, 1, 1, 2, 1, 2, 1, 1, 2, 1, 1, 1, 1, 1, 2, 2, 2, 2, 1, 2, 1, 1, 2,
1, 2, 2, 2, 2, 1, 2, 2, 1, 2, 2, 1, 2, 2, 2, 2, 2, 1, 1, 2, 2,
2, 1, 1, 1, 2, 2, 1, 2, 2, 2, 1, 2, 1, 2, 1, 2, 1, 2])
```

Gambar 2. 13 Hasil Prediksi

Hasil prediksi ditampilkan dalam bentuk *array* yang berisi kode 1 (laris) dan kode 2 (kurang laris) dengan total sebanyak 324 data.

```
#probabilitas dari data test
probabilitas
```

Gambar 2. 14 kode Probabilitas

Gambar 2. 14 kode ini bertujuan untuk menampilkan probabilitas hasil prediksi dari data uji. Probabilitas ini menunjukkan seberapa besar kemungkinan setiap sampel dalam data uji termasuk ke dalam masing – masing kelas.

```
array([[0.49023256, 0.50976744],
[0.4977631, 0.5022369],
[0.53129456, 0.46870544],
[0.47543859, 0.52456141],
[0.47960982, 0.52039018],
[0.43557217, 0.56442783],
[0.46883508, 0.53116492],
[0.46557562, 0.53442438],
[0.49716725, 0.50283275],
[0.49140346, 0.50859654],
[0.56699972, 0.43300028],
[0.40691701, 0.59308297],
[0.38071967, 0.61928033],
[0.49342325, 0.50657675],
[0.49082859, 0.50917141],
[0.62129991, 0.37870007],
[0.44918573, 0.55081427],
[0.45950311, 0.54049689],
[0.4774389, 0.5225611],
[0.53526058, 0.46473942],
[0.42607937, 0.57392063],
[0.45608306, 0.54391694],
[0.4667071, 0.5332929],
[0.58944335, 0.41055665],
[0.59698297, 0.40301703],
[0.41022952, 0.58977048],
[0.50871467, 0.49128533],
```

Gambar 2. 15 Hasil Probabilitas

Gambar 2. 15 hasil probabilitas diatas ditampilkan dalam bentuk *array* dua dimensi, dimana setiap baris merepresentasikan probabilitas suatu sampel dalam data

uji terhadap dua kelas yang tersedia. Setiap nilai dalam *array* menunjukkan peluang suatu sampel termasuk dalam masing – masing kategori, dengan jumlah total probabilitas pada setiap kelas baris mendekati 1.

10. Evaluasi model

Setelah mendapatkan hasil prediksi dan dilatih (model development), tahap selanjutnya adalah evaluasi model. Tujuan evaluasi model adalah mengukur seberapa baik model prediksi bekerja dalam memproses data. Evaluasi ini memastikan bahwa model memiliki tingkat akurasi yang baik, mampu membuat prediksi yang tepat, serta tidak mengalami overfitting atau underfitting. Evaluasi dilakukan dengan menghitung metrik seperti akurasi, presisi, *recall*, dan *f1-score* serta menggunakan *confusion matrix* untuk melihat jumlah prediksi yang benar dan salah.

a. Akurasi

$$akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

b. Presisi

$$precision = \frac{TP}{TP + FP} \quad (3)$$

c. Recall

$$Recall = \frac{TN}{TN + FP} \quad (4)$$

d. F1-score

$$f1 - score = 2 * \frac{precision * recall}{precision + recall} \quad (5)$$

```
# Evaluasi Akurasi
#y_true = label asli pada data uji
y_true = y_test # Assign y_test to y_true
accuracy = accuracy_score(y_true, y_pred)
print(f"Akurasi: {accuracy:.2f}")
```

Gambar 2. 16 Kode Evaluasi Akurasi

```
# Evaluasi Precision, Recall, F1-Score
precision = precision_score(y_true, y_pred, average='weighted')
recall = recall_score(y_true, y_pred, average='weighted')
f1 = f1_score(y_true, y_pred, average='weighted')

print(f"Precision: {precision:.2f}")
print(f"Recall: {recall:.2f}")
print(f"F1-Score: {f1:.2f}")
```

Gambar 2. 17 Kode Evaluasi

Gambar 2. 16 Berdasarkan hasil evaluasi, model menghasilkan akurasi sebesar 0.59 dengan *precision* 0.59, *recall* 0.59 dan *f1-score* 0.58, yang menunjukkan bahwa hasil prediksi sesuai dengan label asli yang terdapat pada data uji. Nilai *precision*, *recall*, *f1-score* dengan metode *average* yang mempertimbangkan distribusi kelas dalam data uji untuk memberikan evaluasi yang lebih seimbang terhadap performa model.

```
# Laporan Klasifikasi
print("\nLaporan Klasifikasi:\n")
print(classification_report(y_true, y_pred))
```

Gambar 2. 18 Kode Laporan Klasifikasi

Gambar 2. 18 kode tersebut menampilkan laporan klasifikasi yang berisi metrik evaluasi seperti *presisi*, *recall*, dan *f1-score* dengan menggunakan fungsi `classification_report(y_true, y_pred)` dari pustaka *sklearn*.

Laporan Klasifikasi:					
	precision	recall	f1-score	support	
1	0.62	0.47	0.53	163	
2	0.57	0.71	0.63	161	
accuracy			0.59	324	
macro avg	0.59	0.59	0.58	324	
weighted avg	0.59	0.59	0.58	324	

Gambar 2. 19 Hasil Klasifikasi Laporan

Gambar 2. 19 hasil klasifikasi menunjukkan bahwa model memiliki akurasi sebesar 59% dengan nilai *precision*, *recall* dan *f1-score* untuk masing – masing kelas.

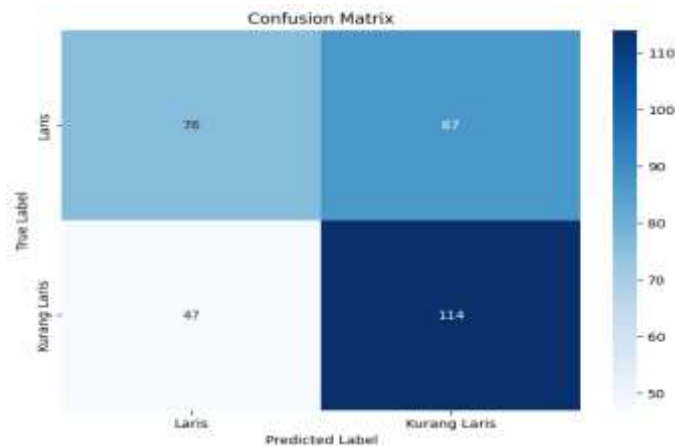
- Kelas 1 (laris) memiliki *precision* 62%, *recall* 47% dan *f1-score* 53% menunjukkan bahwa model lebih sering salah dalam mengidentifikasi kelas ini.
- Kelas 2 (kurang laris) memiliki *precision* 57%, *recall* 71%, dan *f1-score* 63%, yang berarti model lebih baik dalam mendeteksi kelas ini dibanding kelas 1.
- *Macro avg* dan *weighted avg* masing – masing bernilai sekitar 59%, yang menunjukkan performa rata – rata model terhadap seluruh kelas.

Secara keseluruhan model cenderung lebih ke akurat dalam mengidentifikasi produk yang kurang laris dibandingkan dengan laris.

```
# Visualisasi Confusion Matrix
plt.figure(figsize=(8, 5))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=['Laris', 'Kurang Laris'], yticklabels=['Laris', 'Kurang Laris'])
plt.xlabel('Predicted label')
plt.ylabel('True label')
plt.title('Confusion Matrix')
plt.show()
```

Gambar 2. 20 Kode *Confusion Matrix*

Gambar 2. 20 kode ini digunakan untuk membuat visualisasi *confusion matrix* menggunakan *matplotlib* dan *seaborn* dengan tampilan *heatmap* berwarna biru. Label pada sumbu *x* menunjukkan hasil prediksi model, sedangkan label pada sumbu *y* menunjukkan nilai sebenarnya, sehingga memudahkan dalam mengevaluasi performa klasifikasi.



Gambar 2. 21 Visualiasi Confusion Matrix

Gambar 2. 21 menunjukkan *confusion matrix*, yang menggambarkan performa model klasifikasi dalam membedakan kategori laris dan kurang laris.

- 76 produk yang sebenarnya laris diprediksi dengan benar sebagai laris (*true positive*)
- 87 produk yang sebenarnya laris salah diprediksi sebagai kurang laris (*false negative*)
- 47 produk yang sebenarnya kurang laris salah diprediksi sebagai laris (*false positive*)
- 144 produk yang sebenarnya kurang laris diprediksi benar sebagai kurang laris (*true negative*)

Hasil ini menunjukkan bahwa model lebih sering salah memprediksi produk laris sebagai kurang laris yang dapat berdampak pada keputusan bisnis terkait stok dan strategi penjualan.

HASIL DAN DISKUSI

Hasil sistem

Sistem ini menggunakan aplikasi *streamlite* untuk menampilkan prediksi penjualan dengan cara yang sederhana dan interaktif.

1. Halaman login

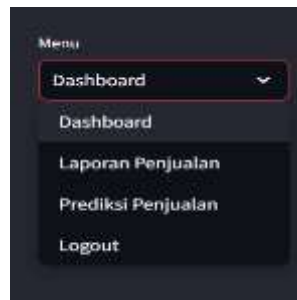


Gambar 3. 1 Halaman *Login*

Pertama kali membuka sistem, pengguna harus login terlebih dahulu dengan

username "admin" dan password "admin123".

2. Halaman menu



Gambar 3. 2 Menu

Setelah berhasil *login*, pengguna akan melihat menu utama yang berisi dashboard, laporan penjualan, prediksi penjualan dan *logout* untuk mengelola data serta melakukan analisis penjualan.

Halaman utama (Dashboard)



Gambar 3. 3 Dashboard

Pengguna dapat memilih menu *dashboard* pada halaman utama untuk melihat deskripsi mengenai aplikasi prediksi penjualan serta informasi terkait fitur sistem.

3. Halaman Laporan Penjualan



Gambar 3. 4 Halaman Laporan Penjualan

Pengguna dapat memilih menu laporan penjualan untuk melihat data penjualan yang telah tercatat dalam sistem, termasuk laporan penjualan berdasarkan tahun tertentu.

a. Upload data penjualan



Gambar 3. 5 Upload File

Langkah pertama, pengguna harus mengunggah data penjualan melalui fitur upload data penjualan untuk dapat diproses dalam sistem.

b. Pilih tahun



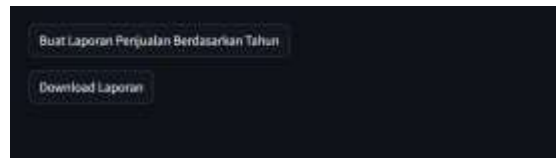
Laporan penjualan untuk tahun 2024:

No	Nama Produk	Rasa	Jenis	Kemasan	Ud	Jumlah Produk	Harga Produk	Bulan	Ta
1	Aoka	Cokelat	Roti	Sachet	60 Pcs	Rendah	Rp. 112.000	Mei	20
2	Aoka	Strawberry	Roti	Sachet	60 Pcs	Rendah	Rp. 112.000	Mei	20
3	Aoka	Pandan	Roti	Sachet	60 Pcs	Rendah	Rp. 112.000	Mei	20
4	Aoka	Bunam	Roti	Sachet	60 Pcs	Rendah	Rp. 112.000	Mei	20
5	Aoka	Nanas	Roti	Sachet	60 Pcs	Rendah	Rp. 112.000	Mei	20
6	Aoka	Cokelat	Roti	Sachet	60 Pcs	Rendah	Rp. 112.000	Mei	20
T	Aoka	Strawberry	Roti	Sachet	60 Pcs	Rendah	Rp. 112.000	Mei	20

Gambar 3. 6 Pilih Tahun

Pengguna perlu memilih tahun pada sistem, misalnya memilih tahun 2024, untuk menampilkan dan menganalisis data penjualan berdasarkan periode yang diinginkan.

c. Pilih buat laporan



Gambar 3. 7 Buat Laporan

Pengguna dapat memilih fitur buat laporan penjualan berdasarkan tahun untuk menampilkan ringkasan dan analisis penjualan sesuai dengan tahun yang dipilih serta *download* laporan tersebut

d. Unduh file laporan



Gambar 3. 8 Download Laporan

Setelah proses unduh selesai, sistem akan menyimpan laporan penjualan berdasarkan tahun 2024 dalam bentuk *file word*.

e. Laporan penjualan



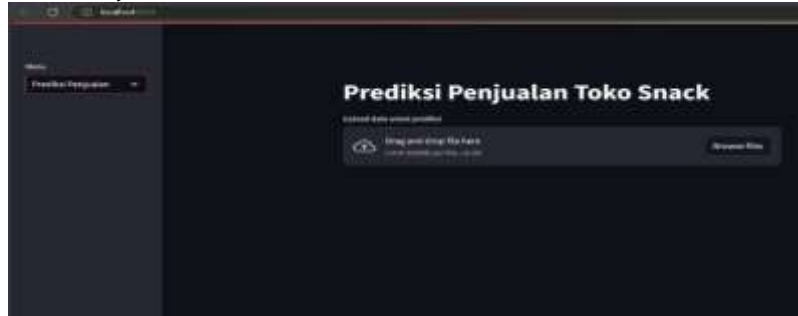
Laporan Penjualan Toko Snack Ibu Ida

No	Nama Produk	Rasa	Jenis	Kemasan	Ud	Jumlah Produk	Harga Produk	Bulan	Tahun	Jumlah Penjualan
1	Aoka	Cokelat	Roti	Sachet	60 Pcs	Rendah	Rp. 112.000	Mei	2024	Laris
2	Aoka	Strawberry	Roti	Sachet	60 Pcs	Rendah	Rp. 112.000	Mei	2024	Laris
3	Aoka	Pandan	Roti	Sachet	60 Pcs	Rendah	Rp. 112.000	Mei	2024	Sangat Laris
4	Aoka	Bunam	Roti	Sachet	60 Pcs	Rendah	Rp. 112.000	Mei	2024	Sangat Laris
5	Aoka	Nanas	Roti	Sachet	60 Pcs	Rendah	Rp. 112.000	Mei	2024	Sangat Laris
6	Aoka	Cokelat	Roti	Sachet	60 Pcs	Rendah	Rp. 112.000	Mei	2024	Laris
7	Aoka	Strawberry	Roti	Sachet	60 Pcs	Rendah	Rp. 112.000	Mei	2024	Laris
8	Aoka	Pandan	Roti	Sachet	60 Pcs	Rendah	Rp. 112.000	Mei	2024	Laris
9	Aoka	Bunam	Roti	Sachet	60 Pcs	Rendah	Rp. 112.000	Mei	2024	Sangat Laris
10	Aoka	Nanas	Roti	Sachet	60 Pcs	Rendah	Rp. 112.000	Mei	2024	Sangat Laris
11	Aoka	Cokelat	Roti	Sachet	60 Pcs	Rendah	Rp. 112.000	Agst	2024	Laris
12	Aoka	Strawberry	Roti	Sachet	60 Pcs	Normal	Rp. 112.000	Agst	2024	Laris
13	Aoka	Pandan	Roti	Sachet	60 Pcs	Normal	Rp. 112.000	Agst	2024	Laris
14	Aoka	Bunam	Roti	Sachet	60 Pcs	Normal	Rp. 112.000	Agst	2024	Sangat Laris
15	Aoka	Nanas	Roti	Sachet	60 Pcs	Normal	Rp. 112.000	Agst	2024	Sangat Laris
16	Aoka	Cokelat	Roti	Sachet	60 Pcs	Rendah	Rp. 112.000	Agst	2024	Laris
17	Aoka	Strawberry	Roti	Sachet	60 Pcs	Normal	Rp. 112.000	Agst	2024	Laris
18	Aoka	Pandan	Roti	Sachet	60 Pcs	Normal	Rp. 112.000	Agst	2024	Laris
19	Aoka	Bunam	Roti	Sachet	60 Pcs	Normal	Rp. 112.000	Agst	2024	Sangat Laris
20	Aoka	Nanas	Roti	Sachet	60 Pcs	Normal	Rp. 112.000	Agst	2024	Sangat Laris

Gambar 3. 9 Laporan Penjualan

File laporan penjualan ini menjelaskan mengenai data penjualan pada tahun 2024, mencakup informasi seperti nama produk, rasa, jenis, kemasan, jumlah produk, harga produk, bulan, serta kategori jumlah penjualan yaitu laris dan kurang laris pada produk.

4. Halaman Prediksi Penjualan



Gambar 3. 10 Halaman Prediksi Penjualan

Pengguna dapat memilih *menu* prediksi penjualan untuk melakukan prediksi penjualan berdasarkan data yang telah dimasukkan ke dalam sistem.

a. Upload data



Gambar 3. 11 Upload File

Langkah pertama dalam melakukan prediksi adalah mengunggah data penjualan dalam format *excel* ke dalam sistem.

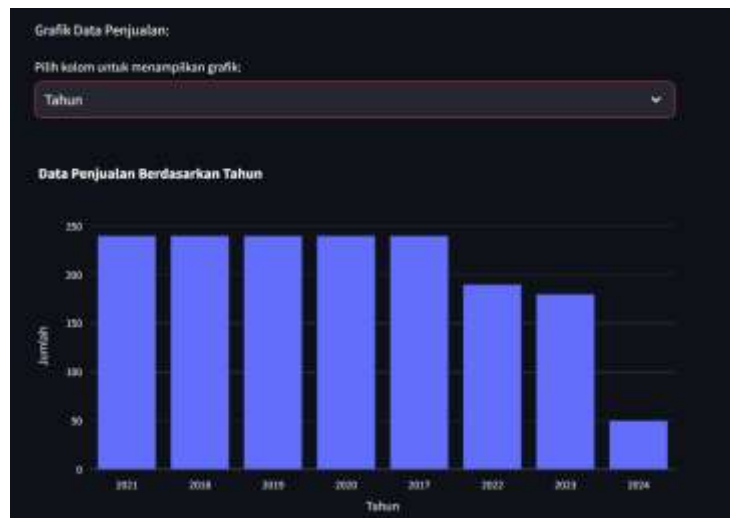
b. Data yang diunggah



Gambar 3. 12 Data yang diunggah

Setelah data diunggah, sistem akan menampilkan data yang telah dimasukkan.

c. Pilih grafik penjualan



Gambar 3. 13 Grafik Penjualan

Kemudian pengguna dapat memilih grafik data penjualan yang ditampilkan berdasarkan tahun.

d. Atribut data



Gambar 3. 14 Atribut Data

Selanjutnya, centang atribut data pada tampilan untuk menampilkan value.

e. Menghapus kolom

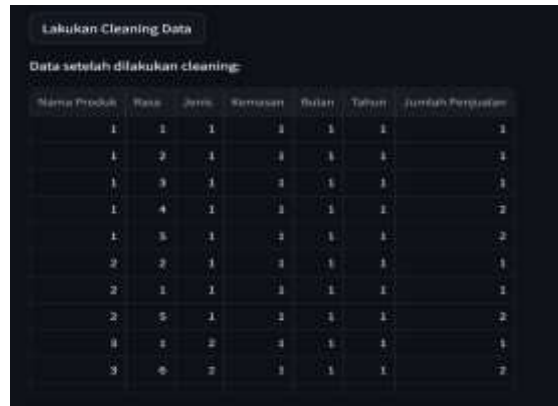
Hapus Kolom tidak diperlukan:

Nama Produk	Rasa	Jenis	Kemasan	Bulan	Tahun	Jumlah Penjualan
Aoka	Cokelat	Roti	Sachet	Mei	2024	Laris
Aoka	Strawberry	Roti	Sachet	Mei	2024	Laris
Aoka	Pandan	Roti	Sachet	Mei	2024	Laris
Aoka	Duren	Roti	Sachet	Mei	2024	Kurang Laris
Aoka	Nanas	Roti	Sachet	Mei	2024	Kurang Laris
Aoka	Cokelat	Roti	Sachet	Mei	2024	Laris
Aoka	Strawberry	Roti	Sachet	Mei	2024	Laris
Aoka	Pandan	Roti	Sachet	Mei	2024	Laris
Aoka	Duren	Roti	Sachet	Mei	2024	Kurang Laris
Aoka	Nanas	Roti	Sachet	Mei	2024	Kurang Laris

Gambar 3. 15 Kolom yang dihapus

Sistem telah menghapus beberapa kolom yaitu no, isi, jumlah produk, dan harga produk serta menampilkan kolom yang akan dianalisis dalam sistem yaitu nama produk, rasa, jenis, kemasan, bulan, tahun, dan jumlah penjualan.

f. Cleaning data



Lakukan Cleaning Data

Data setelah dilakukan cleaning:

Nama Produk	Rasa	Jenis	Kemasan	Bulan	Tahun	Jumlah Penjualan
1	1	1	1	1	1	1
1	2	1	1	1	1	1
1	3	1	1	1	1	1
1	4	1	1	1	1	2
1	5	1	1	1	1	2
2	2	1	1	1	1	1
2	1	1	1	1	1	1
2	5	1	1	1	1	2
3	1	2	1	1	1	1
3	6	2	1	1	1	2

Gambar 3. 16 Cleaning Data

Setelah data dianalisis dalam sistem, sistem melakukan proses pembersihan data (cleaning) dengan mengubah teks menjadi numerik. Dengan data dalam bentuk numerik, sistem dapat mengelolanya menggunakan algoritma naïve bayes.

g. Tahap klasifikasi algoritma naïve bayes




Tahapan Klasifikasi Algoritma Naive Bayes

Jumlah data pada variabel x: 1620 (6 kolom)

Nama Produk	Rasa	Jenis	Kemasan	Bulan	Tahun
1	1	1	1	1	1
1	2	1	1	1	1
1	3	1	1	1	1
1	4	1	1	1	1
1	5	1	1	1	1
1	1	1	1	1	1
1	2	1	1	1	1
1	3	1	1	1	1
1	4	1	1	1	1
1	6	1	1	1	1

Gambar 3. 17 Tahap Klasifikasi Algoritma Naive Bayes

Setelah melalui proses pembersihan data (cleaning), tahap selanjutnya adalah klasifikasi menggunakan algoritma naïve bayes. Pada tahap ini, data disiapkan dengan variabel x dan y untuk proses prediksi. Jumlah data pada variabel x adalah 1.620 dengan 6 kolom



Jumlah data pada variabel y: 1620 (1 kolom)

Jumlah Penjualan

1
1
1
2
2
1
1
1
2
2

Gambar 3. 18 Variabel y

Kemudian, variabel y disiapkan dengan 1.620 data dalam 1 kolom, yaitu jumlah penjualan. Variabel ini digunakan untuk menentukan hasil prediksi, apakah termasuk kategori 1 (laris) atau 2 (kurang laris).

h. Pembagian data

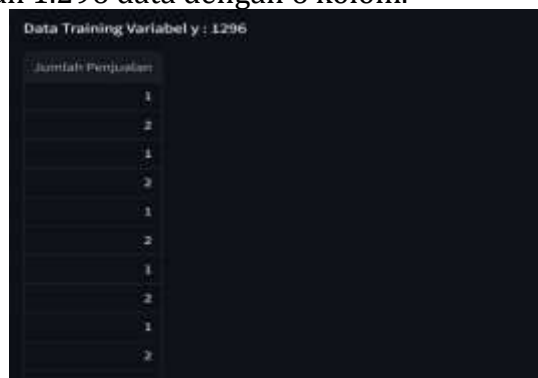


Pembagian Data Training dan Testing
 Jumlah data untuk training: 1296 (80%)
 Jumlah data untuk testing: 324 (20%)
 Data Training Variabel x (1296 data, 6 kolom):

Nama Produk	Rasa	Jenis	Kemasan	Bulan	Tahun
1	5	1	1	2	5
1	3	1	1	3	5
1	3	1	1	10	2
1	5	1	1	7	6
1	4	1	1	4	7
1	1	1	1	4	5
1	5	1	1	9	4
1	5	1	1	7	6
1	5	1	1	5	8
1	2	1	1	5	7

Gambar 3. 19 Pembagian Data

Selanjutnya, dilakukan pembagian data, dimana data training terdiri dari 1.296 data (80%) dan testing sebanyak 324 data (20%). Data yang digunakan untuk training data pada variabel x berjumlah 1.296 data dengan 6 kolom.



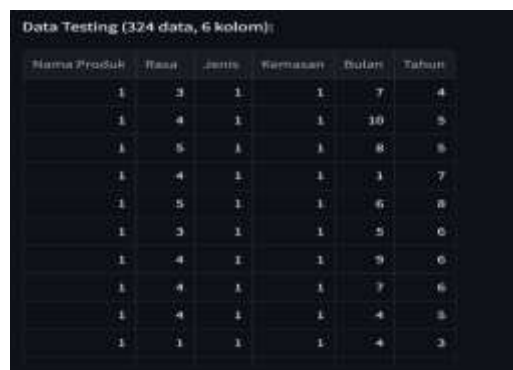
Data Training Variabel y : 1296
 Jumlah Penjualan:

1
2
1
2
1
2
1
2
1
2

Gambar 3. 20 Variabel y

Kemudian, pada variabel y terdapat 1.296 data yang digunakan dalam proses pelatihan model

i. Testing data



Data Testing (324 data, 6 kolom):

Nama Produk	Rasa	Jenis	Kemasan	Bulan	Tahun
1	3	1	1	7	4
1	4	1	1	10	5
1	5	1	1	8	8
1	4	1	1	1	7
1	5	1	1	6	8
1	3	1	1	5	6
1	4	1	1	3	6
1	4	1	1	7	6
1	4	1	1	4	3
1	1	1	1	4	3

Gambar 3. 21 Testing Data

Setelah itu, data diuji (testing) menggunakan 324 data, yaitu 20% dari keseluruhan data, pada variabel x. proses ini bertujuan untuk memprediksi apakah produk tersebut termasuk dalam kategori 1 (laris) atau 2 (kurang laris).

j. Hasil prediksi



Nama Produk	Rasa	Jenis	Kemasan	Bahan	Tahun	Prediksi Jumlah Penjualan	Kategori Prediksi
1	3	1	1	1	7	4	2 Kurang Laris
1	4	1	1	10	5		2 Kurang Laris
1	5	1	1	8	5		1 Laris
1	4	1	1	1	7		2 Kurang Laris
1	5	1	1	6	8		2 Kurang Laris
1	5	1	1	5	6		2 Kurang Laris
1	4	1	1	8	6		2 Kurang Laris
1	4	1	1	7	6		2 Kurang Laris
1	6	1	1	4	5		2 Kurang Laris
1	1	1	1	4	3		2 Kurang Laris

Gambar 3. 22 Hasil Prediksi

Setelah data diuji, sistem telah menghasilkan prediksi penjualan untuk tahun 2025 dengan 324 data yang diprediksi masuk ke dalam kategori 1 (laris) dan 2 (kurang laris). Jumlah produk berdasarkan prediksi dalam kategori 1 (laris) sebanyak 123, sedangkan dalam kategori 2 (kurang laris) sebanyak 201.

k. Probabilitas prediksi



Probabilitas Laris	Probabilitas Kurang Laris
0.4902	0.5098
0.4978	0.5022
0.5313	0.4687
0.4754	0.5246
0.4796	0.5204
0.4356	0.5644
0.4688	0.5312
0.4656	0.5344
0.4972	0.5028
0.4914	0.5086

Gambar 3. 23 Probabilitas Prediksi

Hasil prediksi kategori laris dan kurang laris menampilkan probabilitas untuk setiap data dari 324 data yang telah diprediksi.

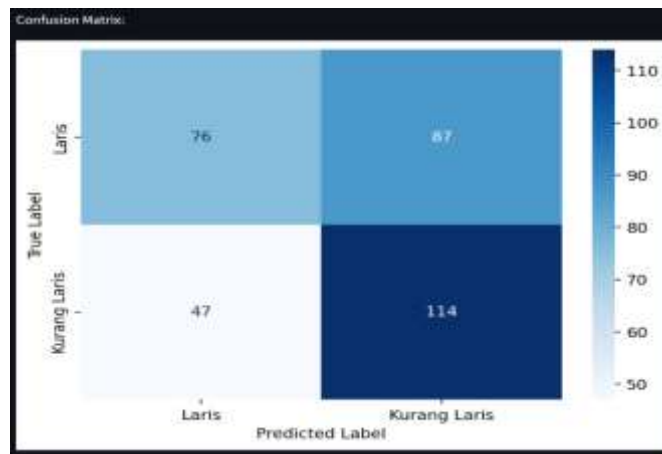
Evaluasi model



Metric	Value
Akurasi	0.59
Precision	0.59
Recall	0.59
F1-Score	0.58

Gambar 3. 24 Evaluasi Model

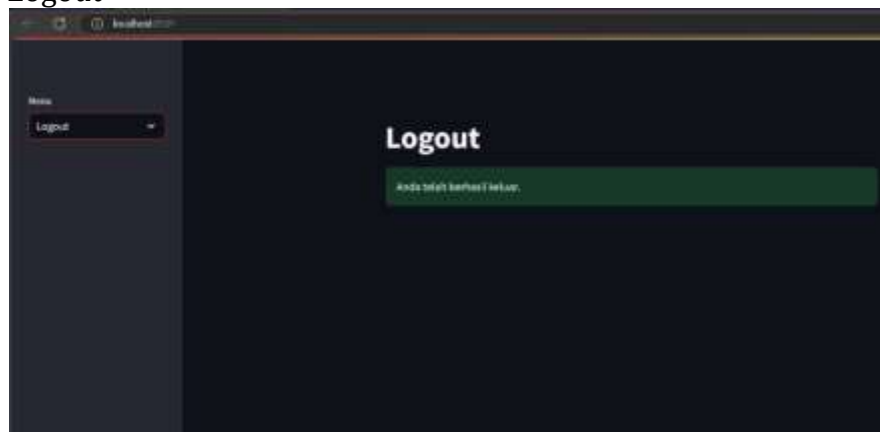
Evaluasi model dilakukan setelah memperoleh hasil prediksi dengan menampilkan nilai akurasi 0.59, precision 0.59, recall 0.59 dan f1-score 0.58.



Gambar 3. 25 Confusion Matrix

Setelah itu, sistem menampilkan visualisasi confusion matrix untuk menganalisis performa hasil prediksi.

5. Halaman Logout



Gambar 3. 26 Logout

Setelah selesai menggunakan sistem, pengguna dapat keluar melalui menu logout, dan sistem akan menampilkan notifikasi “Anda telah berhasil keluar”.

3. 1 Evaluasi model

Setelah proses prediksi selesai, hasil prediksi dibandingkan dengan label asli (y_{test}) untuk mengevaluasi performa model. Evaluasi ini dilakukan dengan menghitung jumlah prediksi yang benar dan salah, kemudian ditampilkan dalam bentuk tabel untuk menghitung akurasi model.

Tabel 3. 1 Confusion Matrix

True label/ Predicted label	Laris	Kurang Laris
Laris (TP+FN)	76	87
Kurang Laris (FP+TN)	47	144

Dimana :

TP (True Positive) = 76 (prediksi benar sebagai laris)

FN (False Negative) = 87 (seharusnya laris, tetapi diprediksi kurang laris)

FP (False Positive) = 47 (seharusnya kurang laris, tetapi diprediksi sebagai laris)

TN (True Negative) = 114 (prediksi benar sebagai kurang laris)

Berikut perhitungan dari akurasi, presisi, *recall* dan *f1 - score* :

1. Accuracy

Perhitungan akurasi dilakukan dengan membandingkan jumlah prediksi yang benar dengan total data uji, yang dinyatakan dalam persentase menggunakan rumus berikut :

$$akurasi = \frac{76 + 144}{76 + 144 + 87 + 47} = \frac{190}{324} = 0.5864 = 0.59$$

Dengan demikian hasil perhitungan dari akurasi adalah 0.59

2. Precision

Precision dihitung dengan membagi jumlah prediksi benar pada kelas positif dengan total prediksi untuk kelas tersebut, menggunakan rumus berikut:

Untuk kelas laris dan kurang laris

$$precision_{laris} = \frac{76}{76 + 47} = \frac{76}{123} = 0.6179$$
$$precision_{kuranglaris} = \frac{114}{114 + 87} = \frac{114}{201} = 0.5672$$

Rata - rata *precision* (macro average)

$$precision = \frac{0.6179 + 0.5672}{2} = 0.5926 = 0.59$$

Dengan demikian hasil perhitungan dari *precision* adalah 0.59

3. Recall

Recall dihitung dengan membagi jumlah prediksi benar pada kelas positif dengan total data aktual untuk kelas tersebut, menggunakan rumus berikut:

Untuk kelas laris dan kurang laris

$$Recall_{laris} = \frac{76}{76 + 87} = \frac{76}{163} = 0.4663$$
$$Recall_{kuranglaris} = \frac{114}{114 + 47} = \frac{114}{161} = 0.7081$$

Rata - rata recall (marco average)

$$Recall = \frac{0.4663 + 0.7081}{2} = 0.5872 = 0.59$$

Dengan demikian hasil perhitungan dari *recall* adalah 0.59

4. F1-Score

F1-score dihitung sebagai keseimbangan antara *precision* dan *recall* dengan menggunakan rata-rata, sesuai rumus berikut:

$$f1 = 2 * \frac{0.6179 * 0.4663}{0.6179 + 0.4663} = 0.5325$$
$$f1 = 2 * \frac{0.5672 * 0.7081}{0.5672 + 0.7081} = 0.6298$$

Rata - rata *f1 - score* (*marco average*)

$$= \frac{0.5235 * 0.6298}{2} = 0.5812 = 0.58$$

Dengan demikian hasil perhitungan dari *f1-score* adalah 0.58

KESIMPULAN

Hasil penelitian ini menunjukkan bahwa prediksi penjualan pada toko snack id menggunakan metode *naïve bayes* belum mencapai tingkat akurasi yang diharapkan. model prediksi telah diimplementasikan dalam sistem berbasis *web* menggunakan *Streamlit* untuk memudahkan prediksi penjualan secara interaktif. Studi kasus pada toko snack id menunjukkan bahwa penerapan metode *naïve bayes* dalam memprediksi penjualan masih memiliki keterbatasan, sehingga perlu dilakukan perbaikan atau pengembangan untuk meningkatkan performa model. Tujuan dari implementasi metode *naïve bayes* dalam penelitian ini adalah untuk membangun sistem yang mampu memprediksi kategori penjualan produk seperti laris atau kurang laris berdasarkan pola *historis* penjualan yang dianalisis dari variabel - variabel seperti nama produk, rasa, jenis, kemasan, bulan dan tahun. Berdasarkan data *historis* sebanyak 1.620 *entri*, model *naïve bayes* dilatih menggunakan 1.296 data latih dan diuji pada 324 data uji. Hasil pengujian menunjukkan bahwa model berhasil mengklasifikasikan 123 data sebagai kategori laris dan 201 data kategori kurang laris. Hasil penelitian ini menunjukkan performa evaluasi model dengan menghasilkan akurasi sebesar 0,59, presisi 0,59, *recall* 0,59 dan *f1-score* 0,58. hasil prediksi ini masih kurang atau belum maksimal dalam akurasi. Faktor - faktor yang mempengaruhi prediksi menurun :

1. Ketidakseimbangan data terjadi pada jumlah sampel dalam satu kategori jauh lebih sedikit dibandingkan dengan kategori lainnya, sehingga cenderung lebih akurat dalam mengenali kategori yang dominan tetapi mengalami kesulitan dalam memprediksi kategori yang jarang muncul.
2. *Naïve bayes* memiliki keterbatasan karena mengasumsikan bahwa setiap fitur tidak saling bergantung, padahal dalam kenyataannya banyak fitur yang memiliki hubungan erat, sehingga ada fitur yang berkolerasi kuat, model dapat menghasilkan prediksi yang akurat.
3. Model yang menggunakan parameter *default* tanpa optimasi tidak sesuai dengan dataset, sehingga perlu dilakukan *hyperparameter tuning* menggunakan *gridserachcv* atau *randomizedserachcv* untuk meningkatkan performa.

DAFTAR PUSTAKA

- [1] Sulastri and Y. S. Nugroho, “Penerapan Data Mining Untuk Prediksi Rating Penjualan Buku Menggunakan Metode Naive Bayes,” *Duta.com*, vol. 12, no. 2, pp. 57–72, 2017.
- [2] N. Indah Kusuma Wardhani, I. Hartami S, and W. Dwi Puspitasari, “Sistem Forecasting Penjualan Beras Dengan Menerapkan Metode Trend Moment,” *JATI (Jurnal Mhs. Tek. Inform.,* vol. 6, no. 2, pp. 901–907, 2022, doi: 10.36040/jati.v6i2.5780.
- [3] F. B. Putra, M. T. Chulkamdi, and F. Febrinita, “IMPLEMENTASI DATA MINING UNTUK MEMPREDIKSI DATA STOK FUKUBI OUTFIT MENGGUNAKAN METODE K- NEAREST NEIGHBOR,” vol. 8, no. 5, pp. 9907–9914, 2024.
- [4] B. Salsabila and I. Cholissodin, “Prediksi Permintaan Keripik Buah dengan Metode Jaringan Syaraf Tiruan Backpropagation (Studi Kasus: CV. Arjuna 999),” vol. 4, no. 6, pp. 1667–1674, 2020, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [5] M. Juanda Saputra and M. Izman Herdiansyah, “Penerapan Naive Bayes Dalam Memprediksi Penjualan Dan Persediaan Kain Jumputan Pada Toko Batiq Colet Tuan Kentang Palembang,” *J. Mantik*, vol. 6, no. 2, pp. 2502–2507, 2022.
- [6] N. M. A. Mahar, Vihi Atina, and Nugroho Arif Sudiby, “Pemodelan Prediksi Kelulusan Mahasiswa Dengan Metode Naive Bayes Di Uniba,” *J. Manaj. Inform. dan Sist. Inf.*, vol. 6, no. 2, pp. 148–158, 2023, doi: 10.36595/misi.v6i2.875.
- [7] M. F. Rozi, R. Siregar, and N. I. Syahputri, “Penerapan Data Mining Menggunakan Metode Naive Bayes Untuk Klasifikasi Data Penentuan Hasil Penjualan Dalam Strategi Pemasaran,” vol. 2, no. 2, pp. 444–454, 2023.
- [8] F. Anggraini, R. Rustam, S. Rahmatullah, and S. Supriyanto, “PENERAPAN DATA MINING JUMLAH PENJUALAN SEPEDA MOTOR MENGGUNAKAN METODE K-MEANS CLUSTERING (Studi Kasus : PT. Tunas Dwipa Matra Way Kanan),” *J. Inform.*, vol. 24, no. 1, pp. 42–51, 2024, doi: 10.30873/ji.v24i1.3685.
- [9] F. Harahap, W. Fahrozi, R. Adawiyah, E. T. Siregar, and A. Y. N. Harahap, “Implementasi Data Mining dalam Memprediksi Produk AC Terlaris untuk Meningkatkan Penjualan Menggunakan Metode Naive Bayes,” *J. Unitek*, vol. 16, no. 1, pp. 41–51, 2023, doi: 10.52072/unitek.v16i1.541.
- [10] Juwita, M. Safii, and B. Efendi Damanik, “Algoritma Naive Bayes Untuk Memprediksi Penjualan Pada Toko VJCakes Pematang Siantar,” *J. Mach. Learn. Artif. Intell.*, vol. 1, no. 4, pp. 337–346, 2022, doi: 10.55123/jomlai.v1i4.1674.
- [11] I. Pramana, I. W. Sudiarsa, and ..., “Penerapan Algoritma Naive Bayes Untuk Prediksi Penjualan Produk Terlaris Pada CV Akusara Jaya Abadi,” *JATISI (Jurnal Tek. ...)*, vol. 10, no. 4, pp. 518–534, 2023, [Online]. Available: <https://jurnal.mdp.ac.id/index.php/jatisi/article/view/6498%0Ahttps://jurnal.mdp.ac.id/index.php/jatisi/article/download/6498/1694>
- [12] H. Hajaroh, T. Suprapti, and R. Narasati, “Implementasi Algoritma Naive Bayes Untuk Analisis Sentimen Ulasan Produk Makanan Dan Minuman Di Tokopedia,” *JATI (Jurnal Mhs. Tek. Inform.,* vol. 8, no. 1, pp. 111–118, 2024, doi: 10.36040/jati.v8i1.8237.
- [13] M. Gulo, Sumarno, and F. Anggraini, “Penerapan Algoritma Naive Bayes dalam Memprediksi Harga Penjualan Makanan Ringan Application of the Naive Bayes Algorithm in Predicting Sales Prices for Snacks,” *JOMLAI J. Mach. Learn. Artif. Intell. Vol.*, vol. 2, no. 2, pp. 125–132, 2023, doi: 10.55123/jomlai.v2i2.2444.
- [14] N. Hadi and J. Benedict, “Implementasi Machine Learning Untuk Prediksi Harga

- Rumah Menggunakan Algoritma Random Forest,” *Comput. J. Comput. Sci. Inf. Syst.*, vol. 8, no. 1, pp. 50–61, 2024, doi: 10.24912/computatio.v8i1.15173.
- [15] Zulkifli, Asmawati.S, and Arnita Irianti, “Penerapan Algoritma Naive Bayes dalam Memprediksi Persediaan Bahan Mebel (Studi Kasus Mebel Usaha Bersama Palipi Soreang),” *J. Comput. Inf. Syst. (J-CIS)*, vol. 5, no. 1, pp. 57–64, 2022, doi: 10.31605/jcis.v5i1.1360.
- [16] P. W. P. Putra, D. Irawan, and ..., “Sistem Prediksi Persediaan Barang Di Mini Market Mars Menggunakan Algoritma Naïve Bayes Classifier Dengan Pemrograman ...,” *JUTIM (Jurnal Tek. ...)*, vol. 7, no. 2, pp. 148–159, 2022, [Online]. Available: <https://jurnal.univbinainsan.ac.id/index.php/jutim/article/view/2101%0Ahttps://jurnal.univbinainsan.ac.id/index.php/jutim/article/download/2101/1077>
- [17] L. Febriyanti and H. Zakaria, “Implementasi Data Mining Untuk Memprediksi Produktivitas Pada Tanaman Kacang Tanah Menggunakan Metode Naive Bayes (Studi Kasus : Perkebunan Kacang Tanah Di Kota Bogor),” *Log. J. Ilmu Komput. dan Pendidik.*, vol. 1, no. 2, pp. 105–118, 2023, [Online]. Available: <https://journal.mediapublikasi.id/index.php/logic>
- [18] I. Yulian, D. S. Anggraeni, and Q. Aini, “Penerapan Metode Trend Moment Dalam Forecasting Penjualan Produk CV. Rabbani Asyisa,” *J. Teknol. dan Sist. Inf.*, vol. 6, no. 2, pp. 193–200, 2020.
- [19] D. Science, I. Technology, and D. Analytics, “Journal of Dinda,” *Journal.Ittelkom-Pwt.Ac.Id/Index.Php/Dinda*, vol. 3, no. 2, pp. 65–70, 2023.
- [20] C. Wulandari, F. Rizki, and A. Lestari, “Prototype Sistem Informasi Publik dan Prediksi Produksi Karet Menggunakan Metode Naive Bayes Berbasis Website,” *BEES Bull. Electr. Electron. Eng.*, vol. 3, no. 1, pp. 57–65, 2022, doi: 10.47065/bees.v3i1.1532.