

## PENERAPAN ALGORITMA *NAIVE BAYES* UNTUK KLASIFIKASI KUALITAS PEMAIN SEPAK BOLA BERDASARKAN DATA STATISTIK PERTANDINGAN

Ivan Suryo Wicaksono<sup>1</sup>, Mustafa<sup>2</sup>

<sup>12</sup>University Islam Sultan Agung

[ivansuryo15@std.unissula.ac.id](mailto:ivansuryo15@std.unissula.ac.id), [cakmustafa@unissula.ac.id](mailto:cakmustafa@unissula.ac.id)

Received: 26-08- 2025

Revised: 18-09-2025

Approved: 25-10-2025

### ABSTRAK

*Sepak bola merupakan olahraga yang semakin berkembang tidak hanya dalam hal permainan, tetapi juga dalam pemanfaatan teknologi untuk menganalisis performa pemain. Dengan sistem berbasis web menggunakan Streamlit, aplikasi ini diharapkan dapat mempermudah manajer tim dalam melakukan seleksi dan pemantauan kualitas pemain secara efisien dan objektif. Penelitian ini bertujuan untuk mengklasifikasikan kualitas pemain sepak bola ke dalam kategori meningkat atau menurun berdasarkan data statistik pertandingan menggunakan algoritma Naïve Bayes. Data yang digunakan diperoleh dari platform Transfermarkt, mencakup variabel seperti jumlah gol, assist, kartu kuning, kartu merah, dan jumlah pertandingan. Dengan data historis sebanyak 1.343 entri(1.098 data latihan, 274 data uji), model ini menghasilkan prediksi kategori Meningkat(456) dan kategori Menurun (642), dengan akurasi sebesar 91%, presisi 91%, recall 91%, f1-score 91% yang menunjukkan efektivitas algoritma ini dalam mendukung pengambilan keputusan berbasis data bagi klub dan pelatih.*

*Kata kunci : klasifikasi, kualitas pemain, naïve bayes, sepak bola, machine learning*

### PENDAHULUAN

Sepak bola merupakan olahraga paling populer di dunia dan terus berkembang, tidak hanya dalam strategi permainan tetapi juga dalam penggunaan teknologi untuk menganalisis performa pemain. Seiring dengan meningkatnya ketersediaan data statistik pertandingan, klub dan pelatih semakin mengandalkan analisis berbasis data untuk mengukur kualitas pemain(Priyo Utomo & Indarto, 2021). Proses penilaian yang sebelumnya hanya mengandalkan observasi subjektif kini dapat diperkuat dengan metode berbasis machine learning yang memberikan hasil lebih objektif dan akurat.

Dalam dunia sepak bola profesional, berbagai aspek performa pemain, seperti jumlah gol, assist, kartu kuning, kartu merah, dan menit bermain, menjadi indikator utama dalam menentukan kontribusi seorang pemain terhadap tim. Namun, tanpa pendekatan analitis yang tepat, data ini sulit digunakan secara efektif untuk mengidentifikasi pola performa pemain. Oleh karena itu, diperlukan sistem klasifikasi yang dapat mengolah data statistik ini menjadi informasi yang berguna bagi klub, pelatih, dan manajer tim.

Machine learning menawarkan berbagai metode dalam analisis data, salah satunya adalah Naïve Bayes(Sinaga dkk., 2022), sebuah algoritma klasifikasi berbasis probabilitas yang sederhana namun efektif. Algoritma ini bekerja berdasarkan Teorema Bayes dengan asumsi independensi antar fitur. Dalam konteks sepak bola, Naïve Bayes dapat digunakan untuk mengklasifikasikan pemain berdasarkan performa mereka, membantu dalam pengambilan keputusan yang lebih baik dalam hal strategi tim maupun rekrutmen pemain.

Penggunaan Naïve Bayes dalam analisis sepak bola semakin menarik karena

algoritma ini memiliki kecepatan komputasi yang tinggi serta mampu menangani dataset dengan jumlah variabel yang relatif banyak. Dengan membangun model klasifikasi menggunakan algoritma ini, diharapkan klub dapat mengidentifikasi pemain berkualitas berdasarkan data statistik pertandingan secara otomatis. Hal ini akan mengurangi subjektivitas dalam penilaian dan meningkatkan efisiensi dalam proses seleksi pemain.

Selain itu, penelitian ini juga bertujuan untuk mengevaluasi performa algoritma Naïve Bayes dalam klasifikasi kualitas pemain sepak bola. Evaluasi dilakukan menggunakan metrik seperti akurasi, presisi, recall, dan F1-score untuk memastikan sejauh mana model yang dikembangkan mampu memberikan hasil yang valid (Erinsyah dkk., 2024). Dengan evaluasi yang tepat, model ini diharapkan dapat memberikan prediksi yang akurat dan membantu klub dalam mengambil keputusan berbasis data.

Dalam penelitian ini, data statistik pemain diperoleh dari situs transfermarkt, yang menyediakan beberapa informasi mengenai pertandingan sepak bola. Dengan menambahkan kolom "kualitas pemain" yang berisi label meningkat atau menurun, dataset yang sebelumnya hanya berisi statistik pertandingan kini memiliki target klasifikasi yang lebih jelas. Sebelumnya, dataset belum memiliki indikator langsung mengenai perubahan performa pemain, sehingga sulit untuk menentukan apakah seorang pemain mengalami peningkatan atau penurunan kualitas. Dengan adanya kolom ini, model Naïve Bayes dapat dilatih untuk memahami pola dari variabel statistik seperti jumlah gol, assist, kartu kuning, kartu merah, dan jumlah pertandingan, lalu mengklasifikasikan pemain berdasarkan tren performanya. Tujuan utama penambahan kolom ini adalah untuk menghasilkan target prediksi yang lebih konkret, sehingga klub, pelatih, atau pencari bakat dapat dengan mudah mengidentifikasi pemain yang mengalami perkembangan atau kemunduran dan mengambil keputusan yang lebih tepat dalam pengelolaan tim.

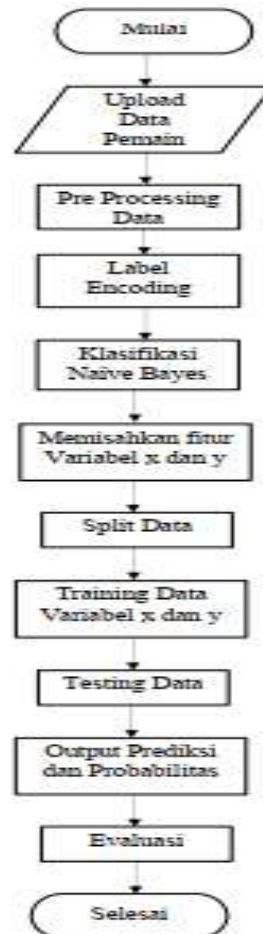
Data ini akan melalui proses preprocessing, termasuk normalisasi, sebelum digunakan untuk pelatihan model. Setelah itu, dataset akan dibagi menjadi data latih dan data uji untuk memastikan model dapat melakukan prediksi dengan baik.

Secara keseluruhan, penelitian ini bertujuan untuk membuktikan efektivitas algoritma Naïve Bayes dalam klasifikasi pemain sepak bola serta mengevaluasi tingkat akurasi. Dengan demikian, diharapkan bahwa metode ini dapat digunakan sebagai alat bantu dalam pengambilan keputusan di dunia sepak bola profesional, baik dalam aspek teknis maupun manajemen tim.

Sistem ini dirancang khusus untuk mendukung kebutuhan beberapa manajer dalam melakukan pencarian dan pemantauan pemain berdasarkan kualitas performa mereka yang mengalami peningkatan atau penurunan. Dengan memanfaatkan data historis dan analisis performa secara berkala, sistem ini mampu memberikan informasi yang relevan dan terkini mengenai tren perkembangan setiap pemain.

Hasil penelitian ini diharapkan dapat memberikan kontribusi bagi dunia sepak bola dalam mengembangkan sistem analisis berbasis kecerdasan buatan. Dengan pemanfaatan machine learning, proses evaluasi pemain dapat menjadi lebih efisien dan objektif, memungkinkan klub untuk lebih cepat dalam mengidentifikasi pemain berbakat serta menentukan strategi permainan yang optimal.

## 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 dikodekan menggunakan label encoding agar variabel kategori dapat diolah oleh algoritma. Selanjutnya data diklasifikasikan menggunakan algoritma *naïve bayes*. Pada tahap ini, dilakukan persiapan variabel  $x$ (fitur) dan variabel  $y$ (label). 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 Statistik Pemain

Penelitian ini menggunakan data statistik pemain untuk mengetahui pemain mana

yang mempunyai kualitas permainan bermain sepak bola untuk sebuah klub. Proses input data pada sistem ini adalah mengupload data pemain dengan file excel kemudian menampilkan data pemain yang sesuai dengan file excel. Pada system ini menggunakan *library* dari *pandas*, dan *numpy*.

## 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. Proses awal pada system ini dimulai dengan menghapus beberapa kolom seperti no, date, nama\_pemain club\_id, dan harga\_pemain. Setelah itu, sistem menyiapkan kolom negara, posisi, yellow\_card, red\_card, gol, assists, jumlah\_pertandingan, dan kualitas\_pemain yang digunakan sebagai fitur dalam proses klasifikasi menggunakan metode *naïve bayes*.

Berikut penjelasan penghapusan kolom disertai penomoran per kolom:

- a. Kolom no dihapus karena hanya berfungsi sebagai *indeks* atau nomor urut data dalam dataset. Karena tidak memiliki nilai informasi yang relevan terhadap variabel target (kualitas\_pemain), maka kolom ini tidak diperlukan dalam proses analisis dan klasifikasi.
- b. Kolom date di hapus karena kolom ini tidak relevan secara langsung terhadap penentuan kualitas pemain karena waktu pertandingan bukan merupakan fitur yang memengaruhi performa pemain dalam model ini. Selain itu, kolom ini bersifat *non-numeric* dan tidak kontributif secara langsung terhadap fitur statistik seperti gol atau assist.
- c. Kolom nama\_pemain dihapus dari proses pengembangan model karena tidak memiliki nilai prediktif dan hanya berfungsi sebagai identitas. Jika dimasukkan, bisa menyebabkan *overfitting* karena model bisa menghafal nama, bukan pola data statistik. Namun, nama\_pemain tetap ditampilkan kembali setelah prediksi untuk memudahkan interpretasi hasil oleh pengguna.
- d. Kolom club\_id di hapus karena merupakan identitas klub dalam bentuk *numeric* atau *string* tertentu. Informasi ini tidak digunakan secara langsung dalam klasifikasi kualitas pemain karena analisis difokuskan pada performa individu (statistik pertandingan) bukan afiliasi klubnya. Penghapusan kolom ini juga mencegah bias terhadap klub tertentu.
- e. Harga pemain di hapus karena bersifat *fluktuatif* dan ditentukan oleh banyak faktor di luar statistik pertandingan (misalnya popularitas, usia, atau kontrak). Kolom ini tidak dijadikan variabel *prediktor* karena dapat menimbulkan bias, dan tidak selalu mencerminkan kualitas performa aktual seorang pemain.

Selanjutnya data yang relevan, seperti negara, posisi, yellow\_card, red\_card, gol, assists, dan jumlah\_pertandingan, dipersiapkan untuk diproses dalam sistem. Faktor – faktor yang mempengaruhi kualitas pemain seperti yellow\_card, red\_card, gol, assists, dan jumlah\_pertandingan akan dianalisis untuk melihat pengaruhnya terhadap tingkat kualitas pemain. Setelah itu, model *naïve bayes* dilatih dengan mempertimbangkan faktor – faktor yang mempengaruhi kualitas pemain, kemudian diuji menggunakan data *testing* untuk mengevaluasi kinerja model berdasarkan *akurasi*, *confusion matrix*, dan *clasification report*.

```
[ ] # Menghapus kolom yang tidak dibutuhkan
kolom_dihapus = ['no','date ','nama_pemain','club_id','harga_pemain'] # Fixed typo: 'apperance_id' to 'appearance_id'
df_bersih = data.drop(columns=kolom_dihapus)
```

**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.

### 3. Label Encoding

*Label Encoding* merupakan teknik dalam pemrosesan data yang mengubah data kategorikal menjadi bentuk numerik dengan memberikan setiap kategori nilai bilangan bulat yang unik. (seperti nama negara, posisi pemain, atau label kualitas pemain) ke dalam bentuk angka. Setiap nilai unik pada fitur kategorikal akan diubah menjadi bilangan bulat yang berbeda. Teknik ini sangat berguna terutama ketika nilai-nilai kategorikal tidak memiliki urutan yang eksplisit, namun diperlukan format numerik untuk diproses oleh algoritma klasifikasi.

```
[ ] import pandas as pd
    from sklearn.preprocessing import LabelEncoder

[ ] # Kolom yang akan diubah dari teks ke angka
kolom_encode = ['negara', 'posisi', 'kualitas_pemain']

[ ] # Inisialisasi LabelEncoder
encoder = LabelEncoder()

[ ] # Melakukan encoding untuk setiap kolom dalam kolom_encode
for kolom in kolom_encode:
    data[kolom] = encoder.fit_transform(data[kolom])
```

**Gambar 2. 3 Label Encoding**

Gambar 2. 3 kode ini melakukan proses label encoder dengan mengganti nilai unik dalam beberapa kolom kategorikal dengan angka. Langkah – langkahnya adalah sebagai berikut :

- Menggunakan pandas untuk memanipulasi data dalam bentuk tabel (DataFrame)
- Menggunakan LabelEncoder digunakan untuk mengubah data kategori(teks) menjadi angka
- Mengubah kolom teks menjadi numerik, kolom yang dirubah adalah negara, posisi, kualitas pemain
- Membuat objek LabelEncoder yang nanti digunakan untuk proses encoding
- Melakukan loop untuk setiap kolom dalam kolom\_encode
- Fit\_transform() akan memetakan setiap nilai unik di kolom menjadi angka, lalu menggantinya di DataFrame data

Dalam *Label Encoding*, terdapat dua jenis encoding yang digunakan, yaitu *Encoding Nominal* dan *Encoding Ordinal*. Berikut adalah penjelasannya:

a. *Encoding nominal*

*Encoding nominal* adalah proses mengubah data kategorikal tanpa urutan menjadi bentuk numerik agar bisa diproses oleh algoritma machine learning. Yang merupakan encoding nominal adalah negara, dan posisi.

b. *Encoding Ordinal*

*Encoding Ordinal* adalah data kategorikal yang memiliki urutan atau tingkatan.

Yang merupakan *encoding ordinal* adalah *yellow\_card*, *red\_card*, *gol*, *assist*, *jumlah\_pertandingan*, dan *kualitas\_pemain*.

#### 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
    nbtrain = 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 *nbtrain*, yang artinya dapat digunakan untuk pelatihan dan prediksi data.

#### 5. Memuat variabel x dan y

Dalam pemodelan ini, variabel X terdiri dari negara, posisi, *yellow\_cards*, *red\_cards*, *gol*, *assists*, dan *jumlah\_pertandingan*, yang merupakan fitur-fitur input. Sementara itu, variabel Y(label) adalah *kualitas\_pemain*, yang menjadi target atau output dari model.

```
[ ] # Memisahkan fitur (X) dan target (y)
    x = data.drop(["kualitas_pemain"], axis=1)
    y = data["kualitas_pemain"]
```

**Gambar 2. 5** Memuat Variabel x

Gambar 2. 5 Kode tersebut untuk memisahkan data menjadi dua bagian, yaitu fitur (X) dan target (y). Pada proses ini, kolom "kualitas\_pemain" dihapus dari DataFrame dan disimpan dalam variabel x karena kolom tersebut merupakan target yang ingin diprediksi, bukan bagian dari fitur. Sementara itu, kolom "kualitas\_pemain" disimpan dalam variabel y sebagai target atau label. Pemisahan ini penting dalam proses machine learning agar model dapat dilatih menggunakan fitur sebagai input dan mempelajari hubungan terhadap target sebagai output.

#### 6. Split Data

Dalam pembagian data ini, proses dilakukan secara acak sebanyak 80% data digunakan untuk pelatihan, sementara 20% data digunakan untuk pengujian.

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

**Gambar 2. 6** Library Sklearn

Gambar 2. 6 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. 7 Split Data

Gambar 2. 7 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  $0,8 * 1.373 = 1.098$  data, digunakan sebagai data training untuk proses pembelajaran model.

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

### Gambar 2. 8 X train

Gambar 2. 8 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. 9 Y Train

Gambar 2. 9 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  $0,2 * 1.373 = 274$  data digunakan sebagai data `testing` untuk mengevaluasi apakah model dapat memprediksi kategori laris atau kurang laris dengan akurat.

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

### Gambar 2. 10 x Test

Gambar 2. 10 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 meningkat atau menurun. pada proses pengujian ini, sebanyak 274 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. 11 Kode Prediksi

Gambar 2. 11 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([1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1,
1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0,
0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0,
1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1,
1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0,
1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0,
1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0,
0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0,
1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0,
0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1,
1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1,
1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0])
```

Gambar 2. 12 Hasil Prediksi

Hasil prediksi ditampilkan dalam bentuk *array* yang berisi kode 0 (meningkat) dan kode 1 (menurun) dengan total sebanyak 275 data.

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

Gambar 2. 13 kode Probabilitas

Gambar 2. 13 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([[3.04652486e-004, 9.99695348e-001],
[1.00000000e+000, 1.21656846e-010],
[4.07401322e-004, 9.99592599e-001],
[3.38683166e-004, 9.99661317e-001],
[1.00000000e+000, 0.00000000e+000],
[1.26638021e-004, 9.99873362e-001],
[6.94917415e-004, 9.99305083e-001],
[1.00000000e+000, 3.05337359e-052],
[1.18195694e-002, 9.88180431e-001],
[1.00000000e+000, 1.35957762e-036],
[5.48694316e-004, 9.99451306e-001],
[3.09948149e-001, 6.90051851e-001],
[9.99999999e-001, 5.97074703e-010],
[2.70407386e-004, 9.99729593e-001],
[2.19212676e-001, 7.80787324e-001],
[1.18777582e-004, 9.99881222e-001],
[1.68184948e-004, 9.99831815e-001],
[3.03922785e-004, 9.99696077e-001],
[1.00000000e+000, 1.55085014e-142],
[1.00000000e+000, 3.61762209e-015],
[1.00000000e+000, 2.43800626e-053],
[4.45742295e-004, 9.99554258e-001],
[1.02759131e-004, 9.99897241e-001],
[7.19910209e-003, 9.92800898e-001],
[1.78571255e-004, 9.99821429e-001],
[9.80686806e-001, 1.93131939e-002],
```

Gambar 2. 14 Hasil Probabilitas

Gambar 2. 14 tersebut menunjukkan output berupa array probabilitas dari hasil prediksi model klasifikasi biner, di mana setiap baris merepresentasikan probabilitas suatu data termasuk ke dalam masing-masing kelas. Nilai pada posisi pertama menunjukkan probabilitas data termasuk ke kelas 0, sedangkan nilai pada posisi kedua menunjukkan probabilitas data termasuk ke kelas 1. Jumlah dari kedua nilai tersebut selalu mendekati 1 karena mewakili distribusi probabilitas antar kelas. Model akan memilih kelas dengan probabilitas tertinggi sebagai hasil prediksi akhir.

### 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.

1. Akurasi

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

2. Presisi

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

3. Recall

$$Recall = \frac{TP}{(TP + FN)} \quad (4)$$

4. F1-Score

$$F1 - Score = \frac{2 \times recall \times presisi}{(recall + presisi)} \quad (5)$$

```
# 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. 15 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. 16 Kode Evaluasi

Gambar 2. 16 Berdasarkan hasil evaluasi, model menghasilkan akurasi sebesar 0.91 dengan precision 0.91, recall 0.91 dan f1-score 0.91, 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. 17 Kode Laporan Klasifikasi

Gambar 2. 17 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 |
|--------------|-----------|--------|----------|---------|
| 0            | 0.83      | 0.93   | 0.88     | 100     |
| 1            | 0.96      | 0.89   | 0.92     | 175     |
| accuracy     |           |        | 0.91     | 275     |
| macro avg    | 0.89      | 0.91   | 0.90     | 275     |
| weighted avg | 0.91      | 0.91   | 0.91     | 275     |

**Gambar 2. 18** Hasil Klasifikasi Laporan

Gambar 2. 18 hasil klasifikasi menunjukkan bahwa model memiliki akurasi sebesar 91% dengan nilai precision, recall dan f1-score untuk masing – masing kelas.

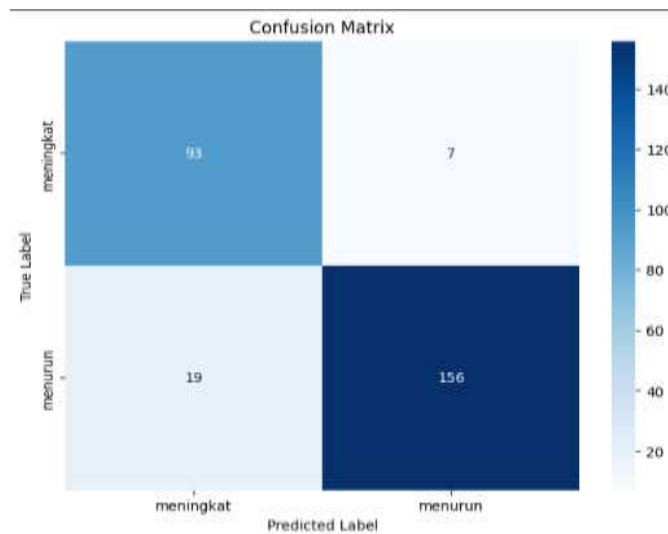
- Kelas 0 (meningkat) memiliki precision 83%, recall 93% dan f1-score 88% menunjukkan bahwa model lebih sering salah dalam mengidentifikasi kelas ini.
- Kelas 1 (menurun) memiliki precision 96%, recall 89%, dan f1-score 92%, yang berarti model lebih baik dalam mendeteksi kelas ini dibanding kelas 1.
- Macro avg dan weighted avg masing – masing bernilai sekitar 90%, 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, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=['meningkat', 'menurun'], yticklabels=['meningkat', 'menurun'])
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.title('Confusion Matrix')
plt.show()
```

**Gambar 2. 19** Kode *Confusion Matrix*

Gambar 2. 19 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. 20** Visualiasi Confusion Matrix

Gambar 2. 20 menunjukkan *confusion matrix*, yang menggambarkan performa

model klasifikasi dalam membedakan kategori meningkat dan menurun.

- 93 pemain yang sebenarnya meningkat diprediksi dengan benar sebagai meningkat (*true positive*)
- 7 pemain yang sebenarnya meningkat salah diprediksi sebagai menurun (*false negative*)
- 19 pemain yang sebenarnya menurun salah diprediksi sebagai meningkat (*false positive*)
- 156 pemain yang sebenarnya menurun diprediksi benar sebagai menurun (*true negative*)

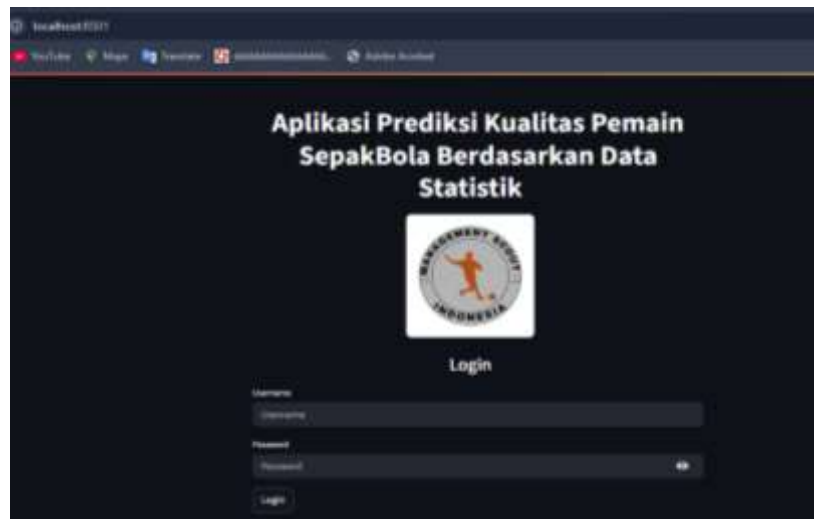
Hasil ini menunjukkan bahwa model lebih sering salah memprediksi pemain kualitas meningkat sebagai menurun yang dapat berdampak pada keputusan pembelian pemain untuk sebuah klub.

## 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*

Gambar 3. 2 setelah berhasil login, pengguna akan melihat menu utama yang berisi dashboard, prediksi dan logout untuk mengelola data serta melakukan analisis kualitas pemain sepak bola.



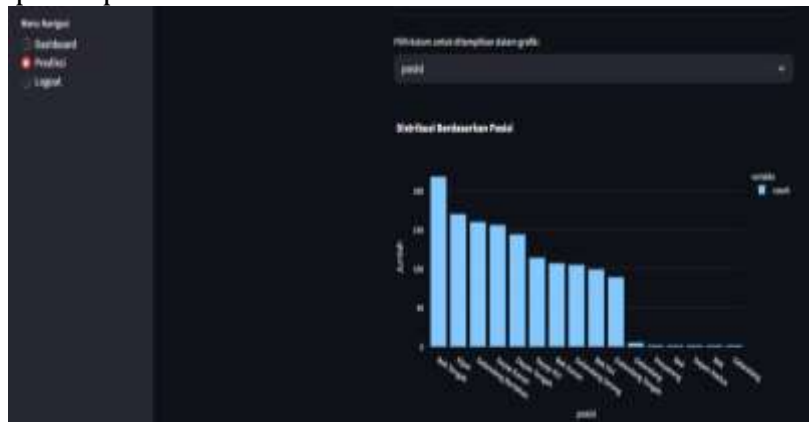
b. Data yang diunggah



Gambar 3. 6 Data Yang Diunggah

Pada gambar 3. 6 setelah data diunggah, sistem akan menampilkan data yang telah dimasukkan.

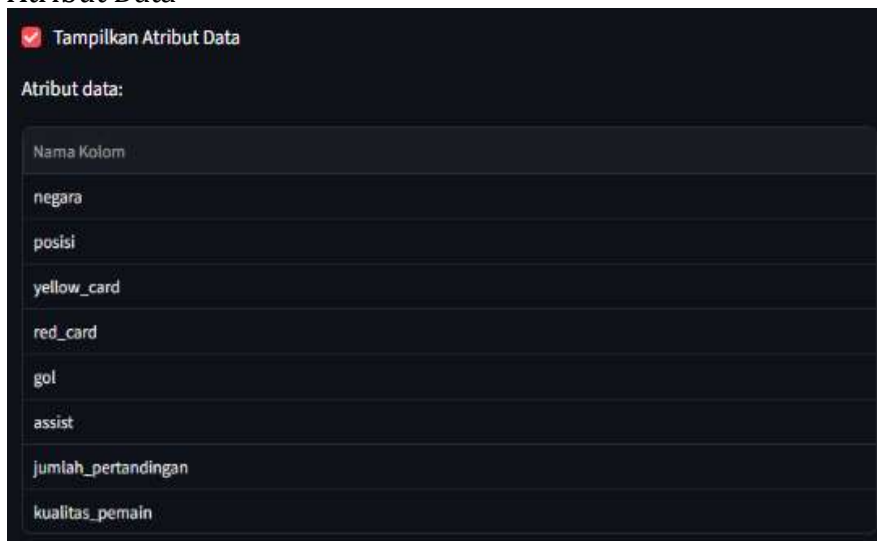
c. Grafik posisi pemain



Gambar 3. 7 Grafik posisi pemain

Gambar 3. 7 kemudian pengguna dapat memilih grafik data posisi yang ditampilkan berdasarkan kualitas pemain.

d. Atribut Data



Gambar 3. 8 Atribut Data

Pada gambar 3. 8 selanjutnya, centang atribut data pada tampilan untuk menampilkan value.

e. Menghapus Kolom

**Menghapus Kolom Tidak Diperlukan**

| negara          | posisi     | yellow_card | red_card | gol | assist | jumlah_pertandingan | kualitas_pe |
|-----------------|------------|-------------|----------|-----|--------|---------------------|-------------|
| Indonesia       | Kiper      | 2           | 1        | 0   | 0      | 24                  | Meningkat   |
| Indonesia       | Kiper      | 2           | 0        | 0   | 0      | 12                  | Meningkat   |
| Makedonia Utara | Bek Tengah | 2           | 0        | 0   | 0      | 16                  | Meningkat   |
| Serbia          | Bek Tengah | 2           | 1        | 0   | 1      | 29                  | Meningkat   |
| Indonesia       | Bek Kanan  | 6           | 1        | 0   | 0      | 29                  | Meningkat   |
| Indonesia       | Bek Tengah | 3           | 0        | 0   | 0      | 27                  | Menurun     |
| Indonesia       | Bek Tengah | 5           | 0        | 0   | 0      | 18                  | Meningkat   |
| Indonesia       | Bek Kanan  | 8           | 0        | 0   | 1      | 26                  | Meningkat   |
| Indonesia       | Bek Tengah | 2           | 0        | 0   | 0      | 18                  | Menurun     |
| Indonesia       | Bek Kanan  | 3           | 0        | 0   | 0      | 9                   | Menurun     |

**Gambar 3. 9** Menghapus Kolom

Gambar 3. 9 sistem telah menghapus beberapa kolom yaitu no, date, nama\_pemain, club\_id, dan harga\_pemain serta menampilkan kolom yang akan dianalisis dalam sistem yaitu negara, posisi, yellow\_card, red\_card, gol, assist dan jumlah\_pertandingan.

f. Label Encoding

**Setelah Label Encoding**

| negara | posisi | yellow_card | red_card | gol | assist | jumlah_pertandingan | kualitas_pemain |
|--------|--------|-------------|----------|-----|--------|---------------------|-----------------|
| 18     | 12     | 2           | 1        | 0   | 0      | 24                  | 0               |
| 18     | 12     | 2           | 0        | 0   | 0      | 12                  | 0               |
| 29     | 4      | 2           | 0        | 0   | 0      | 16                  | 0               |
| 47     | 4      | 2           | 1        | 0   | 1      | 29                  | 0               |
| 18     | 2      | 6           | 1        | 0   | 0      | 29                  | 0               |
| 18     | 4      | 3           | 0        | 0   | 0      | 27                  | 1               |
| 18     | 4      | 5           | 0        | 0   | 0      | 18                  | 0               |
| 18     | 2      | 8           | 0        | 0   | 1      | 26                  | 0               |
| 18     | 4      | 2           | 0        | 0   | 0      | 18                  | 1               |
| 18     | 2      | 3           | 0        | 0   | 0      | 9                   | 1               |

**Gambar 3. 10** Label Encoding

Gambar 3. 10 setelah data dianalisis dalam sistem, sistem melakukan proses label encoding dengan mengubah teks menjadi numerik. Dengan data dalam bentuk numerik, sistem dapat mengelolanya menggunakan algoritma naïve bayes.

g. Tahap Klasifikasi Naive Bayes

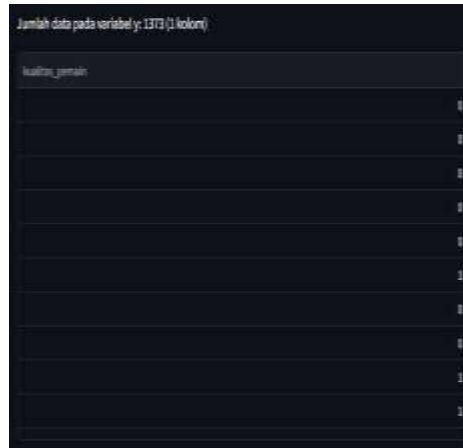
**Penerapan Klasifikasi Naive Bayes**

Jumlah data pada variabel X: 1373 (8 kolom)

| negara | posisi | yellow_card | red_card | gol | assist | jumlah_pertandingan | kualitas_pemain |
|--------|--------|-------------|----------|-----|--------|---------------------|-----------------|
| 18     | 12     | 2           | 1        | 0   | 0      | 24                  | 0               |
| 18     | 12     | 2           | 0        | 0   | 0      | 12                  | 0               |
| 29     | 4      | 2           | 0        | 0   | 0      | 16                  | 0               |
| 47     | 4      | 2           | 1        | 0   | 1      | 29                  | 0               |
| 18     | 2      | 6           | 1        | 0   | 0      | 29                  | 0               |
| 18     | 4      | 3           | 0        | 0   | 0      | 27                  | 1               |
| 18     | 4      | 5           | 0        | 0   | 0      | 18                  | 0               |
| 18     | 2      | 8           | 0        | 0   | 1      | 26                  | 0               |
| 18     | 4      | 2           | 0        | 0   | 0      | 18                  | 1               |
| 18     | 2      | 3           | 0        | 0   | 0      | 9                   | 1               |

**Gambar 3. 11** Tahap Klasifikasi Naive Bayes

Gambar 3. 11 setelah melalui proses label encoding, 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 1.373 dengan 8 kolom.



**Gambar 3. 12** Variabel y

Gambar 3. 12 kemudian, variabel y disiapkan dengan 1.373 data dalam 1 kolom, yaitu jumlah penjualan. Variabel ini digunakan untuk menentukan hasil prediksi, apakah termasuk kategori 0 (meningkat) atau 1 (menurun).

h. Pembagian Data

**Pembagian Data Training dan Testing**

Jumlah data untuk training: 1098 (80%)  
 Jumlah data untuk testing: 275 (20%)

Data Training Variabel X:

| negara | posisi | yellow_card | red_card | gol | assist | jumlah_pertandingan |
|--------|--------|-------------|----------|-----|--------|---------------------|
| 18     | 2      | 4           | 0        | 0   | 1      | 13                  |
| 8      | 4      | 4           | 0        | 1   | 0      | 16                  |
| 18     | 15     | 0           | 0        | 2   | 0      | 8                   |
| 8      | 9      | 7           | 0        | 3   | 1      | 27                  |
| 18     | 12     | 0           | 0        | 0   | 0      | 0                   |
| 18     | 14     | 0           | 0        | 3   | 2      | 15                  |
| 18     | 3      | 5           | 1        | 0   | 0      | 15                  |
| 18     | 9      | 2           | 0        | 0   | 1      | 16                  |
| 18     | 2      | 3           | 0        | 0   | 1      | 11                  |
| 18     | 9      | 5           | 0        | 0   | 0      | 21                  |

**Gambar 3. 13** Pembagian Data Training dan Testing

Gambar 3. 13 selanjutnya, dilakukan pembagian data, dimana data training terdiri dari 1.098 data (80%) dan testing sebanyak 275 data (20%). Data yang digunakan untuk training data pada variabel x berjumlah 1.098 data dengan 8 kolom.



**Gambar 3. 14** Training Variabel y

Pada gambar 3. 14 kemudian, pada variabel y terdapat 1.098 data yang digunakan dalam proses pelatihan model.

i. Testing Data

Data Testing Variabel X:

| negara | posisi | yellow_card | red_card | gol | assist | jumlah_pertandingan |
|--------|--------|-------------|----------|-----|--------|---------------------|
| 18     | 4      | 0           | 0        | 0   | 0      | 9                   |
| 18     | 10     | 6           | 0        | 0   | 2      | 17                  |
| 18     | 15     | 1           | 0        | 0   | 0      | 8                   |
| 18     | 15     | 0           | 0        | 0   | 0      | 8                   |
| 18     | 6      | 2           | 0        | 8   | 1      | 22                  |
| 18     | 10     | 0           | 0        | 0   | 0      | 1                   |
| 18     | 12     | 1           | 0        | 0   | 0      | 13                  |
| 18     | 3      | 5           | 0        | 1   | 4      | 17                  |
| 27     | 4      | 2           | 0        | 0   | 0      | 16                  |
| 29     | 4      | 4           | 1        | 2   | 2      | 30                  |

**Gambar 3. 15** Testing Data

Gambar 3. 15 setelah itu, data diuji (testing) menggunakan 1.098 data, yaitu 20% dari keseluruhan data, pada variabel x. proses ini bertujuan untuk memprediksi apakah produk tersebut termasuk dalam kategori 0 (meningkat) atau 1 (menurun).

j. Hasil Prediksi

Hasil Prediksi Kualitas Pemain Berdasarkan Data Statistik

| negara | posisi | yellow_card | red_card | gol | assist | jumlah_pertandingan | Prediksi kualitas_pemain | Ka |
|--------|--------|-------------|----------|-----|--------|---------------------|--------------------------|----|
| 18     | 4      | 0           | 0        | 0   | 0      | 9                   | 1                        | Me |
| 18     | 10     | 6           | 0        | 0   | 2      | 17                  | 0                        | Me |
| 18     | 15     | 1           | 0        | 0   | 0      | 8                   | 1                        | Me |
| 18     | 15     | 0           | 0        | 0   | 0      | 8                   | 1                        | Me |
| 18     | 6      | 2           | 0        | 8   | 1      | 22                  | 0                        | Me |
| 18     | 10     | 0           | 0        | 0   | 0      | 1                   | 1                        | Me |
| 18     | 12     | 1           | 0        | 0   | 0      | 13                  | 1                        | Me |
| 18     | 3      | 5           | 0        | 1   | 4      | 17                  | 0                        | Me |
| 27     | 4      | 2           | 0        | 0   | 0      | 16                  | 1                        | Me |
| 29     | 4      | 4           | 1        | 2   | 2      | 30                  | 0                        | Me |

**Gambar 3. 16** Hasil Prediksi

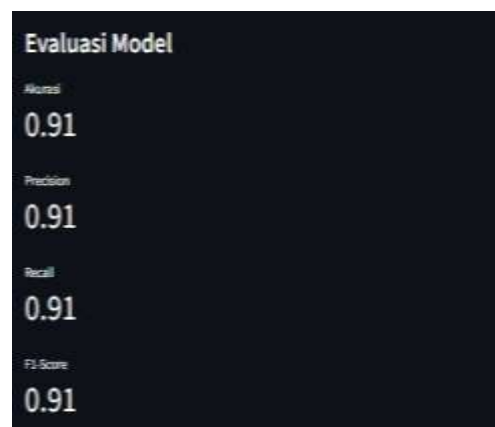
Gambar 3. 16 setelah data diuji, sistem telah menghasilkan prediksi kualitas pemain dengan 1.098 data yang diprediksi masuk ke dalam kategori 0 (menurun) dan 1 (meningkat). Jumlah produk berdasarkan prediksi dalam kategori 0 (meningkat) sebanyak 456, sedangkan dalam kategori 1 (menurun) sebanyak 642.

| Probabilitas Meningkat | Probabilitas Menurun |
|------------------------|----------------------|
| 3.0114657433e-04       | 9.9969885343e-01     |
| 9.9999999989e-01       | 1.1483176360e-10     |
| 4.0360065997e-04       | 9.9959639934e-01     |
| 3.3553822716e-04       | 9.9966446177e-01     |
| 1.0000000000e+00       | 0.0000000000e+00     |
| 1.2509591358e-04       | 9.9987490409e-01     |
| 6.8730936567e-04       | 9.9931269063e-01     |
| 1.0000000000e+00       | 2.2574250941e-52     |
| 1.1743285833e-02       | 9.8825671417e-01     |
| 1.0000000000e+00       | 1.0234922599e-36     |

Gambar 3. 17 Probabilitas Prediksi

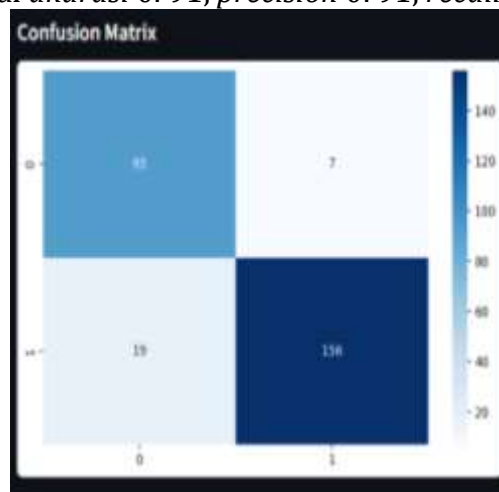
Gambar 3. 17 hasil prediksi kategori menurun dan meningkat menampilkan probabilitas untuk setiap data dari 257 data yang telah diprediksi.

k. Evaluasi Model



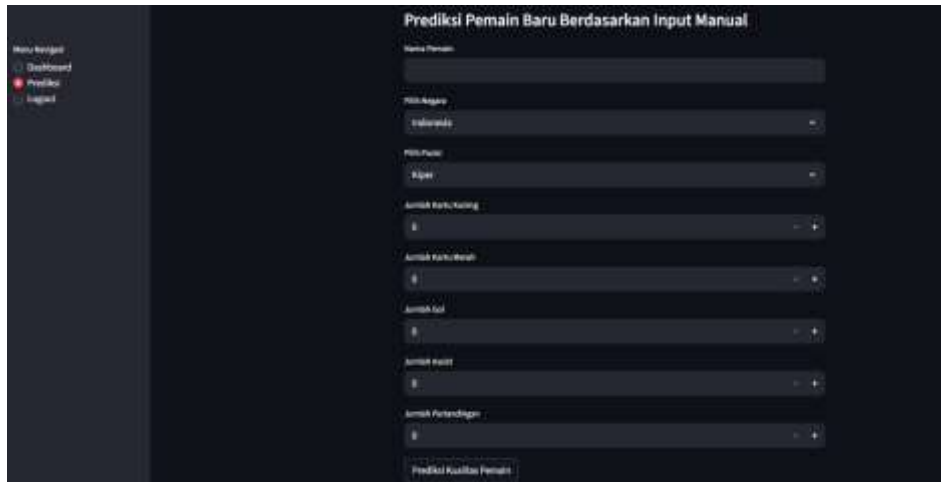
Gambar 3. 18 Hasil Evaluasi Model

Pada gambar 3. 18 evaluasi model dilakukan setelah memperoleh hasil prediksi dengan menampilkan nilai *akurasi* 0. 91, *precision* 0. 91, *recall* 0.91 dan *f1-score* 0.91 .



Gambar 3. 19 Confusion Matrix

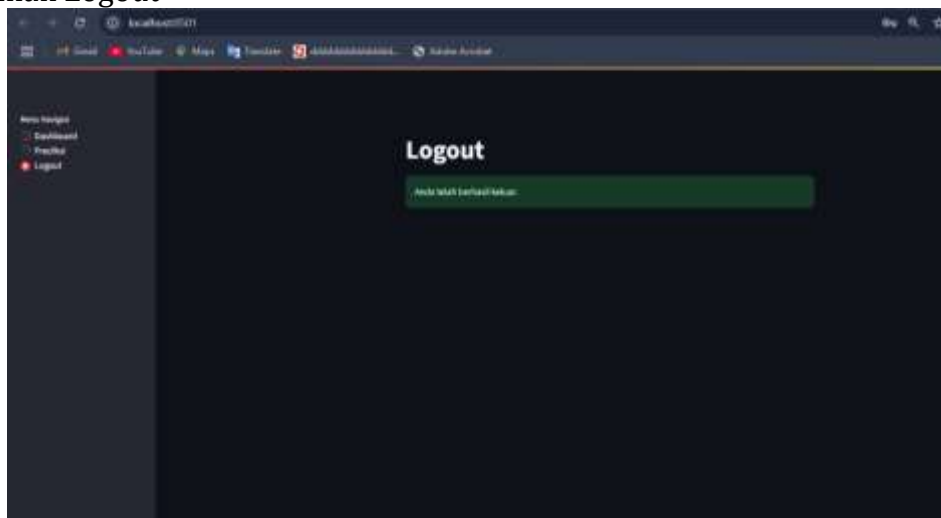
Pada gambar 3. 19 setelah itu, sistem menampilkan visualisasi *confusion matrix* untuk menganalisis performa hasil prediksi.



**Gambar 3. 20** Prediksi Pemain Baru Berdasarkan *Input* Manual

Pada gambar 3.20 fitur ini dapat melakukan prediksi kualitas pemain sepak bola dengan cara memasukkan data secara manual melalui form yang telah disediakan pada sistem. Fitur ini memudahkan manajer tim untuk mengevaluasi pemain baru sehingga dapat digunakan sebagai bahan pertimbangan dalam proses seleksi dan strategi tim.

5. Halaman Logout



**Gambar 3. 21** Logout

Pada gambar 3. 21 setelah selesai menggunakan sistem, pengguna dapat keluar melalui menu *logout*, dan sistem akan menampilkan notifikasi “Anda telah berhasil keluar”.

**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 modeEl.

**Tabel 3. 1** Confusion Matrix

| True label/ Predicted label | Meningkat | Menurun |
|-----------------------------|-----------|---------|
| Meningkat (TP+FN)           | 93        | 7       |
| Menurun (FP+TN)             | 19        | 156     |

Dimana :

TP (True Positive) = 93 (prediksi benar sebagai laris)

FN (False Negative) = 7 (seharusnya laris, tetapi diprediksi kurang laris)

FP (False Positive) = 19 (seharusnya kurang laris, tetapi diprediksi sebagai laris)

TN (True Negative) = 156 (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 :

$$akurasi = \frac{93 + 156}{93 + 156 + 7 + 19} = \frac{249}{275} = 0.9054 = 0.91$$

Dengan demikian hasil perhitungan dari akurasi adalah 0.91

### 2. Precision

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

Untuk kelas Meningkatkan dan Menurun

$$precisionmeningkat = \frac{93}{93 + 19} = \frac{93}{112} = 0.8303$$

$$precisionmenurun = \frac{156}{156 + 7} = \frac{156}{163} = 0.9570$$

Rata - rata *precision* (macro average)

$$precision = \frac{0.8303 + 0.9570}{2} = 0.91$$

Dengan demikian hasil perhitungan dari *precision* adalah 0.91

### 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

$$Recallmeningkat = \frac{93}{93 + 7} = \frac{93}{100} = 0.93$$

$$Recallmenurun = \frac{156}{156 + 19} = \frac{156}{175} = 0.8914$$

Rata - rata *recall* (marco average)

$$Recall = \frac{0.93 + 0.8914}{2} = 0.9107 = 0.91$$

Dengan demikian hasil perhitungan dari *recall* adalah 0.91

#### 4. F1-Score

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

$$\begin{aligned} f1 &= 2 * \frac{0.8303 * 0.93}{0.8303 + 0.93} = 0.8773 \\ f1 &= 2 * \frac{0.9570 * 0.8914}{0.9570 + 0.8914} = 0.9232 \\ \text{Rata - rata } f1 - \text{score (marco average)} &= \frac{0.8773 * 0.9232}{2} = 0.912 = 0.91 \end{aligned}$$

Dengan demikian hasil perhitungan dari *f1-score* adalah 0.91

### KESIMPULAN

Penelitian ini berhasil menerapkan algoritma **Naïve Bayes** dalam mengklasifikasikan kualitas pemain sepak bola berdasarkan data statistik pertandingan. Proses pengolahan data dilakukan melalui tahap preprocessing, seperti penghapusan kolom yang tidak relevan dan penerapan label encoding, sehingga data siap digunakan untuk melatih model. Dengan pendekatan ini, variabel seperti jumlah gol, assist, kartu kuning, kartu merah, dan jumlah pertandingan dapat diolah menjadi informasi yang relevan untuk mengidentifikasi pola performa pemain.

Hasil pengujian terhadap **1.343 data** pemain yang dibagi menjadi data latih dan data uji menunjukkan bahwa model mampu melakukan klasifikasi kualitas pemain ke dalam dua kategori utama, yaitu meningkat dan menurun. Model menghasilkan prediksi yang konsisten dengan pola data historis, sehingga dapat memberikan gambaran yang lebih objektif mengenai tren perkembangan performa pemain sepak bola.

Berdasarkan evaluasi, algoritma Naïve Bayes memberikan hasil yang sangat baik dengan **akurasi sebesar 91%**, serta nilai presisi, recall, dan *f1-score* yang juga mencapai 91%. Hal ini menunjukkan bahwa model yang dibangun tidak hanya mampu memberikan hasil klasifikasi dengan tingkat akurasi tinggi, tetapi juga cukup andal dalam mendeteksi kedua kategori performa pemain secara seimbang.

Selain menghasilkan model klasifikasi, penelitian ini juga berhasil mengimplementasikan sistem berbasis web menggunakan Streamlit. Sistem ini memungkinkan pengguna, khususnya pelatih dan manajer tim, untuk melakukan seleksi dan pemantauan kualitas pemain secara lebih efisien dan objektif. Dengan demikian, penelitian ini membuktikan bahwa algoritma Naïve Bayes dapat menjadi solusi praktis untuk mendukung pengambilan keputusan berbasis data dalam dunia sepak bola profesional.

## **DAFTAR PUSTAKA**

- Erinsyah, M. F., Sasmito, G. W., Wibowo, D. S., & Bakti, V. K. (2024). Sistem Evaluasi Pada Aplikasi Akademik Menggunakan Metode Skala Likert Dan Algoritma Naïve Bayes. *Komputa : Jurnal Ilmiah Komputer Dan Informatika*, 13(1), 74–82.  
<https://doi.org/10.34010/komputa.v13i1.10940>
- Priyo Utomo, N., & Indarto, P. (2021). Analisis Keterampilan Teknik Dasar Passing dalam Sepak Bola. *Jurnal Porkes*, 4(2), 87–94. <https://doi.org/10.29408/porkes.v4i2.4578>
- Sinaga, S., Sembiring, R. W., & Sumarno, S. (2022). Penerapan Algoritma Naive Bayes untuk Klasifikasi Prediksi Penerimaan Siswa Baru. *Journal of Machine ...*, 1(1), 55–64.  
<https://journal.fkpt.org/index.php/malda/article/view/162%0Ahttps://journal.fkpt.org/index.php/malda/article/download/162/115>
- Astuti, Y., Wulandari, I. R., Putra, A. R., & Kharomadhona, N. (2022). Naïve Bayes untuk Prediksi Tingkat Pemahaman Kuliah Online Terhadap Mata Kuliah Algoritma Struktur Data. *Jurnal Edukasi Dan Penelitian Informatika (JEPIN)*, 8(1), 28.  
<https://doi.org/10.26418/jp.v8i1.48848>
- Damanik, A. R., Sumijan, S., & Nurcahyo, G. W. (2021). Prediksi Tingkat Kepuasan dalam Pembelajaran Daring Menggunakan Algoritma Naïve Bayes. *Jurnal Sistim Informasi Dan Teknologi*, 3, 88–94. <https://doi.org/10.37034/jsisfotek.v3i3.49>
- Dang Kurniawan, D. (2025). METODE NAÏVE BAYES UNTUK KLASIFIKASI SENTIMEN TWEET PEMAIN NATURALISASI TIM NASIONAL SENIOR SEPAK BOLA INDONESIA. In *Jurnal Mahasiswa Teknik Informatika* (Vol. 9, Issue 2).
- Dania, S., Ishak, R., Kom, M., & Dalai, H. (2024). Penerapan Algoritma Naive Bayes Classifier Untuk Klasifikasi Judul Skripsi Berdasarkan Konsentrasi. *Jurnal Ilmiah Ilmu Komputer Banthayo Lo Komputer*, 3(1), 15–22.
- Herdian, C., Kamila, A., & Agung Musa Budidarma, I. G. (2024). Studi Kasus Feature Engineering Untuk Data Teks: Perbandingan Label Encoding dan One-Hot Encoding Pada Metode Linear Regresi. *Technologia : Jurnal Ilmiah*, 15(1), 93.  
<https://doi.org/10.31602/tji.v15i1.13457>
- Humam, M. N. (2023). Perbandingan Kinerja CNN dan Naïve Bayes pada Analisis Sentimen Performa Manchester United di Twitter. *Journal of Information Engineering and Educational Technology*, 7(2), 83–91. <https://doi.org/10.26740/jieet.v7n2.p83-91>
- Laia, M. P., Hafidz, R. M., & Hestiani, M. D. (2023). Proses Pengumpulan Data Dan Keakuratan Aplikasi Transfermarkt Sebagai Acuan Dari Harga Transfer Pemain Sepakbola Dunia. *OKTAL : Jurnal Ilmu Komputer Dan Science*, 2(5), 1526–1533.
- Material, P., Sandy, B., & Sitohang, S. (2022). Penerapan Algoritma Naive Bayes Dalam Memprediksi. *Jurnal Comasie*, 4(2), 848–858.
- Metode, P., Algoritma, C., & Naive, D. A. N. (2024). Perbandingan metode algoritma c4.5 dan naive bayes untuk memprediksi penjualan kosmetik pada toko jelita 1,2. 7(2), 220–225.
- Musrifin, A. Y., & Bausad, A. A. (2020). Analisis Unsur Kondisi Fisik Pemain Sepak Bola Mataram Soccer Akademi Ntb. *Jurnal Ilmiah Mandala Education*, 6(1), 113–119.  
<https://doi.org/10.58258/jime.v6i1.1116>
- Pradana, D., Luthfi Alghifari, M., Farhan Juna, M., & Palaguna, D. (2022). Klasifikasi Penyakit Jantung Menggunakan Metode Artificial Neural Network. *Indonesian Journal of Data and Science*, 3(2), 55–60. <https://doi.org/10.56705/ijodas.v3i2.35>
- Prasetyo, A. B., & Laksana, T. G. (2022). Optimasi Algoritma K-Nearest Neighbors dengan Teknik Cross Validation Dengan Streamlit (Studi Data: Penyakit Diabetes). *Journal of Applied Informatics and Computing (JAIC)*, 6(2), 194. <http://jurnal.polibatam.ac.id/index.php/JAIC>
- Pratama, F. F., & Nurhasanah, Y. I. (2020). Penggunaan Metode Profile Matching Dan Naïve Bayes Untuk Menentukan Starting Eleven Pada Sepak Bola. *Jurnal Tekno Insentif*, 14(2), 59–68.  
<https://doi.org/10.36787/jti.v14i2.268>
- Putra, I. N. T. A., Kartini, K. S., Suyitno, Y. K., Sugiarta, I. M., & Puspita, N. K. E. (2023). Penerapan Library Tensorflow, Cvzone, dan Numpy pada Sistem Deteksi Bahasa Isyarat Secara Real Time. *Jurnal Krisnadana*, 2(3), 412–423. <https://doi.org/10.58982/krisnadana.v2i3.335>
- Sartika, D., Andreswari, D., & Anggriani, K. (2023). Penentuan Posisi Ideal Pemain Dalam Cabang

- Olahraga Sepak Bola Dengan Menggunakan Pendekatan Dua Metode Naïve Bayes & Profile Matching. *Jurnal Rekursif / ISSN 2303-0755*, 4(3), 311-324.
- Suryanto, T. L. M., Nuryananda, P. F., & Wibowo, N. C. (2023). Mangunjaya Watch: Sekolah Sepakbola Berbasis Teknologi dan Analisis Data Digital. *I-Com: Indonesian Community Journal*, 3(4), 1571-1582. <https://doi.org/10.33379/icom.v3i4.3262>
- Wahyudin, Y., & Rahayu, D. N. (2020). Analisis Metode Pengembangan Sistem Informasi Berbasis Website: A Literatur Review. *Jurnal Interkom: Jurnal Publikasi Ilmiah Bidang Teknologi Informasi Dan Komunikasi*, 15(3), 26-40. <https://doi.org/10.35969/interkom.v15i3.74>