PRAKTEK TEXT MINING CLASSIFICATION

NAMA : AURA NAJMA KUSTIANANDA

NIM : 164221053

MATA KULIAH : DATA MINING II

Kode ipynb dapat dilihat di: https://github.com/aura-najma/kodelaprakdatmin2

1. Impor data dan library yang diperlukan

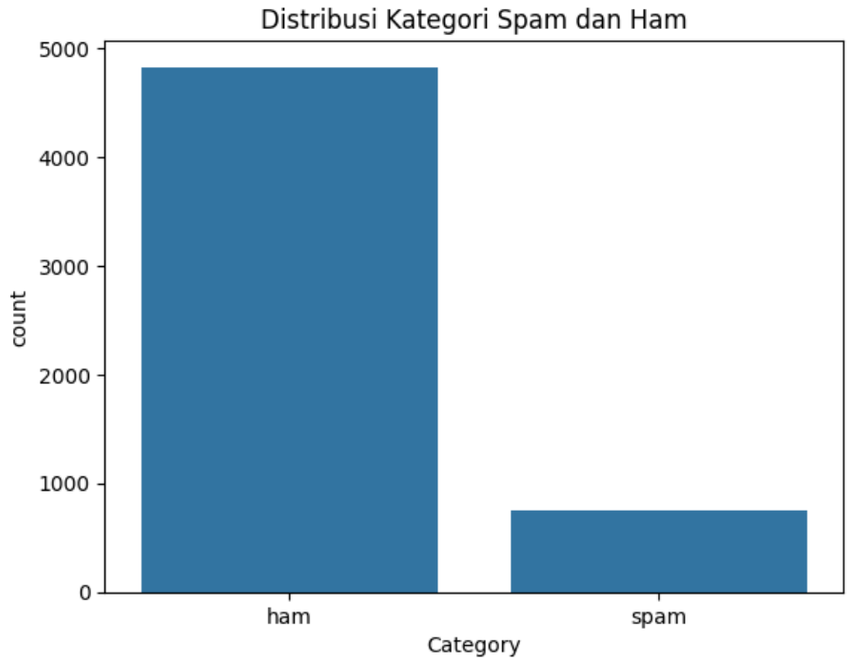
```
import numpy as np
import pandas as pd

import nltk
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory


from nltk.corpus import stopwords
from sklearn.feature_extraction.text import TfidfVectorizer
from nltk.classify.util import accuracy
from nltk.tokenize import word_tokenize
import string
import re

from collections import Counter
from wordcloud import WordCloud

from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
from sklearn.metrics import ConfusionMatrixDisplay

import matplotlib.pyplot as plt
import seaborn as sns
```

```
file_path = r'C:\Users\Acer\OneDrive\Pictures\Kuliah\Semester 5\Data Mining II - Laprak\data datmin ii\Spam_text_message.csv'

df = pd.read_csv(file_path)

df.head()
```

	Category	Message
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...

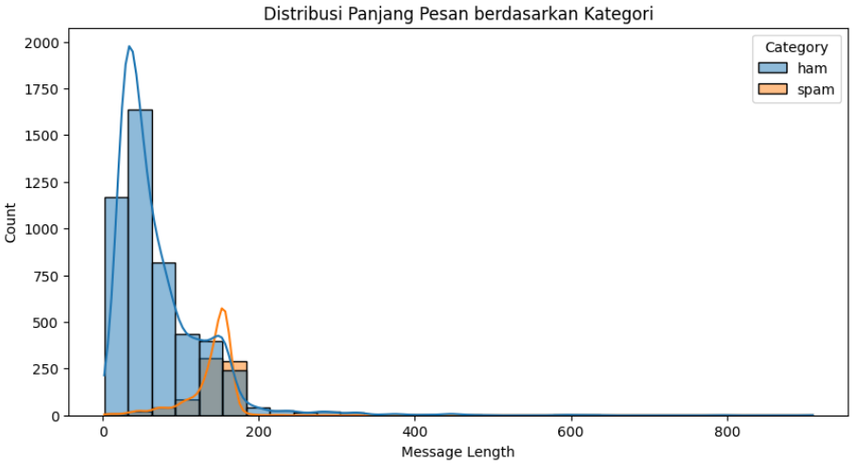
2. Melakukan exploratory data analysis

```
sns.countplot(data=df, x='Category')
plt.title('Distribusi Kategori Spam dan Ham')
plt.show()
```



Dapat dilihat bahwa terdapat ketidakseimbangan dari kategori pesan, di mana jumlah pesan ham berjumlah hampir 5000, sementara pesan spam tidak menyentuh angka 1000. Hal ini dapat menyebabkan model klasifikasi tidak bekerja dengan baik

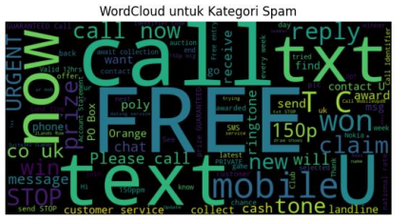
```
df['Message Length'] = df['Message'].apply(len)


plt.figure(figsize=(10, 5))
sns.histplot(data=df, x='Message Length', hue='Category', bins=30, kde=True)
plt.title('Distribusi Panjang Pesan berdasarkan Kategori')
plt.show()
```



Dapat dilihat bahwa pesan-pesan spam biasanya lebih panjang daripada pesan-pesan ham.

```
spam_messages = ' '.join(df[df['Category'] == 'spam']['Message'])
ham_messages = ' '.join(df[df['Category'] == 'ham']['Message'])

fig, ax = plt.subplots(1, 2, figsize=(15, 7))
wordcloud_spam = WordCloud(width=800, height=400, max_words=100).generate(spam_messages)
wordcloud_ham = WordCloud(width=800, height=400, max_words=100).generate(ham_messages)

ax[0].imshow(wordcloud_spam, interpolation='bilinear')
ax[0].set_title('WordCloud untuk Kategori Spam')
ax[0].axis('off')

ax[1].imshow(wordcloud_ham, interpolation='bilinear')
ax[1].set_title('WordCloud untuk Kategori Ham')
ax[1].axis('off')

plt.show()
```

WordCloud untuk Kategori Spam

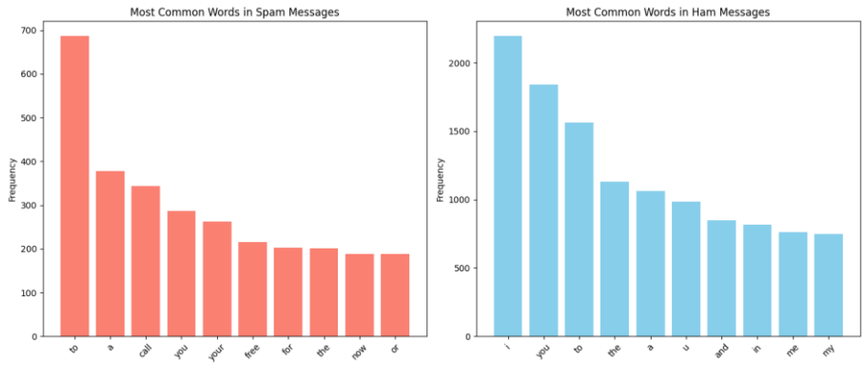


WordCloud untuk Kategori Ham

Di kategori spam, free, won, claim, text, send, merupakan kata-kata yang dominan dan memang erat kaitannya dengan karakteristik pesan spam yang biasanya berisi tawaran hadiah, imbauan untuk menghubungi suatu pihak demi hadiah, dan sebagainya. Sementara di kategori ham, kata-kata yang dominan mencakup now, ok, go, time, need, come, yang mencerminkan bahasa manusia sehari-hari..

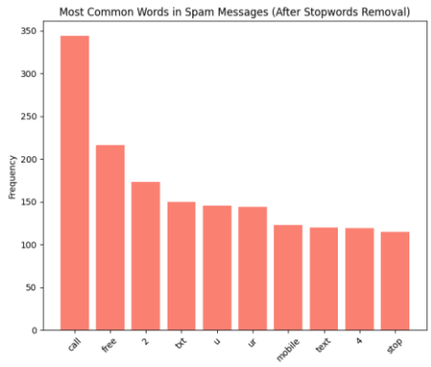
```
spam_texts, ham_texts = [], []

for i in range(len(df)):
    teks = df['Message'][i]
    teks_bersih = re.sub(r'[^\w\s]', '', teks.lower())
    if df['Category'][i] == 'spam':
        spam_texts.append(teks_bersih)
    elif df['Category'][i] == 'ham':
        ham_texts.append(teks_bersih)

spam_words = ' '.join(spam_texts).split()
ham_words = ' '.join(ham_texts).split()

common_spam_words = Counter(spam_words).most_common(10)
common_ham_words = Counter(ham_words).most_common(10)
```

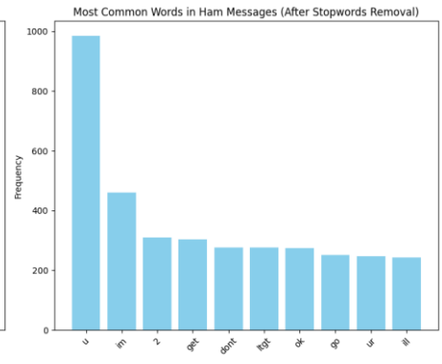
```
plt.figure(figsize=(14, 6))
plt.subplot(1, 2, 1)
spam_words, spam_counts = zip(*common_spam_words)
plt.bar(spam_words, spam_counts, color='salmon')
plt.title('Most Common Words in Spam Messages')
plt.xticks(rotation=45)
plt.ylabel('Frequency')

plt.subplot(1, 2, 2)
ham_words, ham_counts = zip(*common_ham_words)
plt.bar(ham_words, ham_counts, color='skyblue')
plt.title('Most Common Words in Ham Messages')
plt.xticks(rotation=45)
plt.ylabel('Frequency')

plt.tight_layout()
plt.show()
```



Gambar di atas menunjukkan 10 kata terbanyak di setiap kategori. Kata tersebut mayoritas tidak memiliki makna, seperti the, u, a, and, dan lainnya.

```
spam_texts_stop = [list_bersih_stop[i] for i in range(len(df)) if df['Category'][i] == 'spam']
ham_texts_stop = [list_bersih_stop[i] for i in range(len(df)) if df['Category'][i] == 'ham']

spam_words_stop = ' '.join(spam_texts_stop).split()
ham_words_stop = ' '.join(ham_texts_stop).split()

common_spam_words_stop = Counter(spam_words_stop).most_common(10)
common_ham_words_stop = Counter(ham_words_stop).most_common(10)

plt.figure(figsize=(14, 6))
plt.subplot(1, 2, 1)
spam_words, spam_counts = zip(*common_spam_words_stop)
plt.bar(spam_words, spam_counts, color='salmon')
plt.title('Most Common Words in Spam Messages (After Stopwords Removal)')
plt.xticks(rotation=45)
plt.ylabel('Frequency')

plt.subplot(1, 2, 2)
ham_words, ham_counts = zip(*common_ham_words_stop)
plt.bar(ham_words, ham_counts, color='skyblue')
plt.title('Most Common Words in Ham Messages (After Stopwords Removal)')
plt.xticks(rotation=45)
plt.ylabel('Frequency')

plt.tight_layout()
plt.show()
```





Setelah dilakukan pembersihan dengan stopwords, dapat dilihat bahwa 10 kata terbanyak di setiap kategori terlihat lebih memiliki makna, seperti ada call, free, stop di kategori spam dan dont, go, di kategori ham. Namun, perlu dicatat sebetulnya kata-kata ini juga kurang bermakna karena banyak di antara kata-kata tersebut yang merupakan singkatan/slank.

3. Lakukan preprocessing dengan pembersihan tanda baca dan stopwords

```
list_bersih = []
for i in range(len(df)):
    teks = df['Message'][i]
    teks_bersih = re.sub(r'[^\w\s]', '', teks.lower())
    list_bersih.append(teks_bersih)
    print("Teks ke", i, "sudah dibersihkan")
```

```
stop_word_set = set(stopwords.words('english'))


def remove_stopwords(text, stopwords):
    words = text.split()
    filtered_words = [word for word in words if word not in stopwords]
    return ' '.join(filtered_words)

list_bersih_stop = []

for i, teks in enumerate(list_bersih):
    teks_bersih_stop = remove_stopwords(teks, stop_word_set)
    list_bersih_stop.append(teks_bersih_stop)
    print(f"Teks ke {i} sudah dibersihkan")
```

4. Lakukan ekstraksi fitur dengan TF-IDF dan mapping variabel y

```
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(df['Message'])
print(X.shape)
y = df['Category'].map({'ham': 0, 'spam': 1})
```

5. Lakukan splitting data

```
X_train_val, X_test, y_train_val, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

X_train, X_val, y_train, y_val = train_test_split(X_train_val, y_train_val, test_size=0.2, random_state=42)
```

Data dipisah menjadi 64 persen train, 16 persen, val, dan 20 persen sisanya test.

6. Melakukan klasifikasi dengan Random Forest

    a. Model

```
rf_classifier = RandomForestClassifier(random_state=42, class_weight='balanced')
rf_classifier.fit(X_train, y_train)
```

```
RandomForestClassifier
RandomForestClassifier(class_weight='balanced', random_state=42)
```

Model klasifikasi Random Forest diatur dengan class_weight = balanced karena adanya ketimpangan kelas. Dengan class_weight yang diatur, harapannya model bisa lebih adil ke kelas ham dan spam.

    b. Akurasi dan classification report

```
y_val_pred = rf_classifier.predict(X_val)

val_accuracy = accuracy_score(y_val, y_val_pred)
print(f'Validation Accuracy: {val_accuracy * 100:.2f}%')

print("Validation Set Classification Report:")
print(classification_report(y_val, y_val_pred, target_names=['ham', 'spam']))
```

```
Validation Accuracy: 97.20%
Validation Set Classification Report:
              precision    recall  f1-score   support

         ham       0.97      1.00      0.98       772
        spam       1.00      0.79      0.88       120

    accuracy                           0.97       892
   macro avg       0.98      0.90      0.93       892
weighted avg       0.97      0.97      0.97       892
```

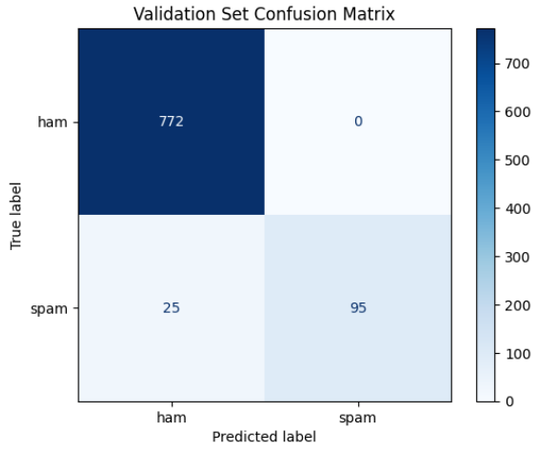
Dari data validation, akurasi yang didapatkan sebesar 97.2 persen, yang merupakan akurasi yang baik. Semua pesan yang diprediksi sebagai spam adalah spam, tetapi hanya 79 persen pesan yang benar-benar spam dilabeli sebagai spam, sementara sisanya diklasifikasikan sebagai ham. Ini menyebabkan precision ham menjadi sebesar 97 persen, karena 3 persennya kemungkinan besar adalah pesan berlabel spam yang salah diklasifikasikan. Tetapi, karena nilai recall-nya sebesar 1, artinya semua pesan ham berhasil diprediksi sebagai ham juga.

```
y_test_pred = rf_classifier.predict(X_test)

test_accuracy = accuracy_score(y_test, y_test_pred)
print(f'Test Accuracy: {test_accuracy * 100:.2f}%')

print("Test Set Classification Report:")
print(classification_report(y_test, y_test_pred, target_names=['ham', 'spam']))
```

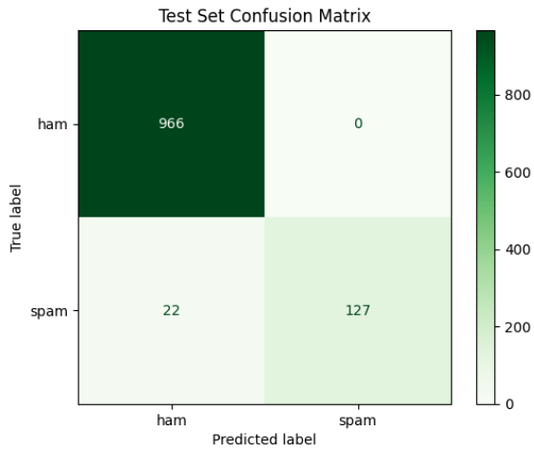
```
Test Accuracy: 98.03%
Test Set Classification Report:
              precision    recall  f1-score   support

         ham       0.98      1.00      0.99       966
        spam       1.00      0.85      0.92       149

    accuracy                           0.98      1115
   macro avg       0.99      0.93      0.95      1115
weighted avg       0.98      0.98      0.98      1115
```

Akurasi dari data test sebesar 98.03 persen menunjukkan bahwa model Random Forest merupakan model yang baik. Karena akurasinya tidak jauh berbeda dengan data validasi, ini juga menunjukkan model tidak overfitting atau underfitting. Terdapat peningkatan akurasi pada recall spam. Dari seluruh pesan berlabel spam, hanya 15 persen yang salah diklasifikasikan, mengakibatkan nilai precision ham sebesar 0.98. Kesimpulannya, model Random Forest ini adalah model yang baik untuk mengklasifikasikan pesan berlabel ham, tetapi membutuhkan peningkatan untuk pesan berlabel spam.

c. Confusion matrix



Dapat dilihat bahwa dari hasil validation, terdapat 25 pesan spam yang salah dikategorikan.



Dapat dilihat dari hasil test, bahwa hanya ada 22 pesan spam yang salah dikategorikan.

d. Analisis terkait teks yang salah diklasifikasikan

```
val_misclassified = df.iloc[y_val.index][y_val != y_val_pred]
val_misclassified['Predicted'] = y_val_pred[y_val != y_val_pred]
val_misclassified['Actual'] = y_val[y_val != y_val_pred]

print("Misclassified Instances in Validation Set:")
for idx, row in val_misclassified.iterrows():
    print(f"Index: {idx}, Message: {row['Message']}")
    print(f"Actual: {row['Actual']}, Predicted: {row['Predicted']}\n")
```

Python

```
Misclassified Instances in Validation Set:
Index: 943, Message: How about getting in touch with folks waiting for company? Just txt back your NAME and AGE to opt in! Enjoy the community (150p/SM
Actual: 1, Predicted: 0

Index: 5381, Message: You have 1 new message. Call 0207-083-6089
Actual: 1, Predicted: 0

Index: 5427, Message: Santa Calling! Would your little ones like a call from Santa Xmas eve? Call 09058094583 to book your time.
Actual: 1, Predicted: 0

Index: 5098, Message: TheMob>Hit the link to get a premium Pink Panther game, the new no. 1 from Sugababes, a crazy Zebra animation or a badass Hoody w
Actual: 1, Predicted: 0

Index: 955, Message: Filthy stories and GIRLS waiting for your
Actual: 1, Predicted: 0
```

```
test_misclassified = df.iloc[y_test.index][y_test != y_test_pred]
test_misclassified['Predicted'] = y_test_pred[y_test != y_test_pred]
test_misclassified['Actual'] = y_test[y_test != y_test_pred]

print("\nMisclassified Instances in Test Set:")
for idx, row in test_misclassified.iterrows():
    print(f"Index: {idx}, Message: {row['Message']}")
    print(f"Actual: {row['Actual']}, Predicted: {row['Predicted']}\n")
```

Python

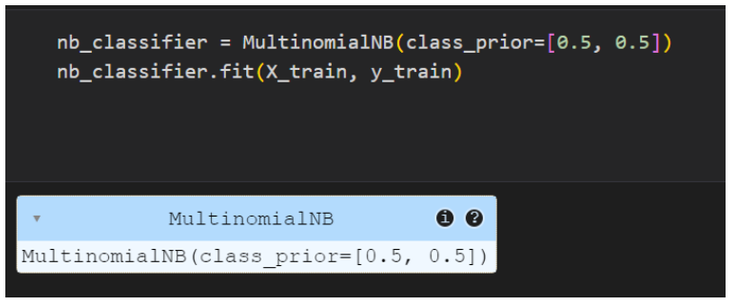
```
Misclassified Instances in Test Set:
Index: 881, Message: Reminder: You have not downloaded the content you have already paid for. Goto http://doit. mymoby. tv/ to collect your content.
Actual: 1, Predicted: 0

Index: 3864, Message: Oh my god! I've found your number again! I'm so glad, text me back xafter this msgs cst std ntwk chg £1.50
Actual: 1, Predicted: 0

Index: 2575, Message: Your next amazing xxx PICSFREE1 video will be sent to you enjoy! If one vid is not enough for 2day text back the keyword PICSFREE
Actual: 1, Predicted: 0

Index: 3548, Message: Rock yr chik. Get 100's of filthy films &XXX pics on yr phone now. rply FILTH to 69669. Saristar Ltd, E14 9YT 08701752560. 450p p
Actual: 1, Predicted: 0

Index: 2402, Message: Babe: U want me dont u baby! Im nasty and have a thing 4 filthyguys. Fancy a rude time with a sexy bitch. How about we go slo n h
Actual: 1, Predicted: 0
```

Kesalahan klasifikasi pada model Random Forest ini dapat disebabkan oleh beberapa hal, seperti pola pesan spam yang mirip dengan ham sehingga model salah memprediksi/ Contohnya adalah "Hello darling how are you today? I would

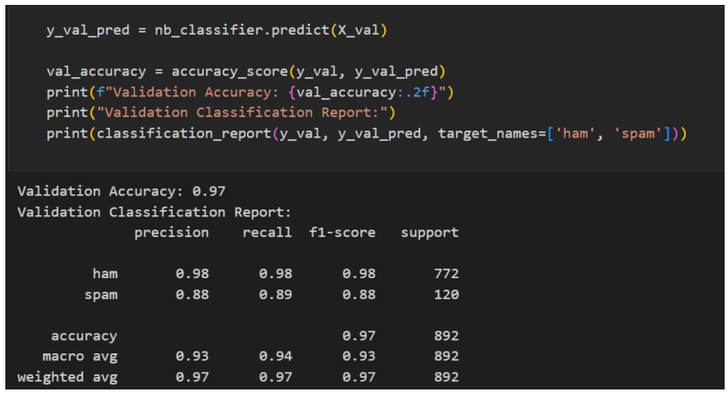
love to have a chat..." dan "Do you realize that in about 40 years, we'll have...". Alasan lain yang mungkin adalah model kurang belajar dari data berlabel spam karena adanya ketimpangan data. Contohnya, pesan seperti "Reminder: You have not downloaded the content.." atau "Xmas & New Years Eve tickets are now on sale..." dapat dianggap sebagai pesan sungguhan yang mengingatkan dan mempromosikan sesuatu. Selain itu, banyak pesan spam yang menggunakan kata-kata biasa seperti pesan berlabel ham, seperti hi, call, atau free, sehingga model pun sulit menangkap pola pesan berlabel spam.

7. Melakukan klasifikasi dengan Naive Bayes

   a. Model



   b. Akurasi dan classification report



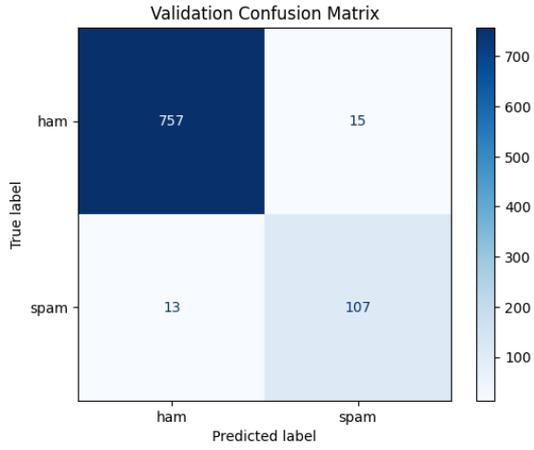
Model ini mencapai akurasi 97%, menunjukkan performa yang sangat baik dalam mengklasifikasikan pesan. Precision dan recall tinggi pada kelas "ham" (masing-masing 98%) mengindikasikan bahwa model sangat akurat dalam mengidentifikasi pesan non-spam. Untuk kelas "spam," precision dan recall masing-masing 88% dan 89%, menunjukkan model sedikit kesulitan dalam mendeteksi semua pesan spam dengan benar, tetapi tetap efektif secara keseluruhan.

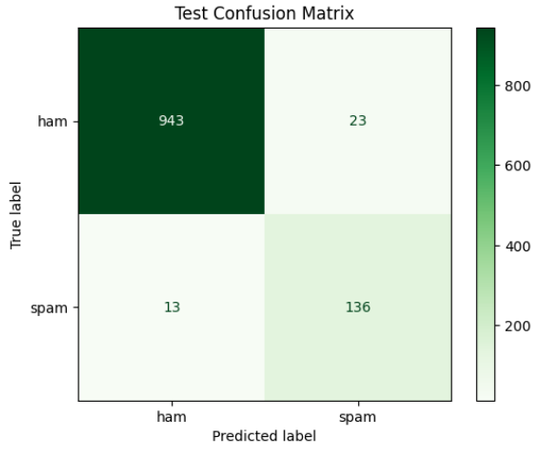
```
y_test_pred = nb_classifier.predict(X_test)

test_accuracy = accuracy_score(y_test, y_test_pred)
print(f"Test Accuracy: {test_accuracy:.2f}")
print("Test Classification Report:")
print(classification_report(y_test, y_test_pred, target_names=['ham', 'spam']))
```

```
Test Accuracy: 0.97
Test Classification Report:
              precision    recall  f1-score   support

         ham       0.99      0.98      0.98       966
        spam       0.86      0.91      0.88       149

    accuracy                           0.97      1115
   macro avg       0.92      0.94      0.93      1115
weighted avg       0.97      0.97      0.97      1115
```

Hasil dari akurasi test mencapai 97 persen, sama seperti hasil akurasi val. Ini berarti model secara konsisten sudah baik. Sementara itu, nilai precision untuk ham untuk test sedikit lebih tinggi daripada untuk val, sama seperti nilai precision untuk spam lebih rendah sedikit daripada untuk val. Hal ini menandakan model memang sedikit kurang baik dalam mengklasifikasikan pesan spam.

c. Confusion matrix





Hasil dari test menunjukkan kesalahan klasifikasi untuk pesan ham lebih banyak daripada hasil dari val.

d. Analisis terkait teks yang salah diklasifikasikan

```
val_misclassified = df.iloc[y_val.index][y_val != y_val_pred]
val_misclassified['Predicted'] = y_val_pred[y_val != y_val_pred]
val_misclassified['Actual'] = y_val[y_val != y_val_pred]

print("Misclassified Instances in Validation Set:")
for idx, row in val_misclassified.iterrows():
    print(f"Index: {idx}, Message: {row['Message']}")
    print(f"Actual: {row['Actual']}, Predicted: {row['Predicted']}\n")
```

```
Misclassified Instances in Validation Set:
Index: 3358, Message: Brainless Baby Doll..:-D;-), vehicle sariyag drive madoke barolla..
Actual: 0, Predicted: 1

Index: 4021, Message: University of southern california.
Actual: 0, Predicted: 1

Index: 943, Message: How about getting in touch with folks waiting for company? Just txt back your NAME and AGE to opt in! Enjoy the community (150p
Actual: 1, Predicted: 0

Index: 3788, Message: WHORE YOU ARE UNBELIEVABLE.
Actual: 0, Predicted: 1

Index: 1381, Message: i dnt wnt to tlk wid u
Actual: 0, Predicted: 1
```

```
test_misclassified = df.iloc[y_test.index][y_test != y_test_pred]
test_misclassified['Predicted'] = y_test_pred[y_test != y_test_pred]
test_misclassified['Actual'] = y_test[y_test != y_test_pred]

print("\nMisclassified Instances in Test Set:")
for idx, row in test_misclassified.iterrows():
    print(f"Index: {idx}, Message: {row['Message']}")
    print(f"Actual: {row['Actual']}, Predicted: {row['Predicted']}\n")
```

```
Misclassified Instances in Test Set:
Index: 4937, Message: K..k.:)congratulation ..
Actual: 0, Predicted: 1

Index: 1961, Message: Guess what! Somebody you know secretly fancies you! Wanna find out who it is? Give us a call on 09065394973 from Landline DATEBox
Actual: 1, Predicted: 0

Index: 3864, Message: Oh my god! I've found your number again! I'm so glad, text me back xafter this msgs cst std ntwk chg £1.50
Actual: 1, Predicted: 0

Index: 2575, Message: Your next amazing xxx PICSFREE1 video will be sent to you enjoy! If one vid is not enough for 2day text back the keyword PICSFREE
Actual: 1, Predicted: 0

Index: 2245, Message: No management puzzeles.
Actual: 0, Predicted: 1
```

Kesalahan klasifikasi dalam model Naive Bayes lebih beragam daripada model Random Forest, karena model ini juga salah mengklasifikasikan pesan ham, tidak hanya spam.

Untuk spam:

- Karena model Naive Bayes bekerja dengan asumsi independensi kata-kata, sehingga ketika ada kata-kata yang muncul pada pesan spam tapi sering muncul pada pesan ham, terjadi kesalahan klasifikasi.
- Struktur kalimat yang terlalu umum dan konteks yang ambigu, sehingga ketika pesan spam menggunakan kalimat seperti percakapan sehari-hari, model bisa salah mengklasifikasikan. Contohnya adalah "Hi this is Amy, we will be sending you a free phone number in a couple of days, which will give you an access to all the adult parties..." di mana bagian adult parties menunjukkan bahwa pesan ini spam, tetapi karena awalnya ini terlihat seperti pesan dari seorang teman, model salah mengklasifikasikan.
- Untuk ham:

- Kata-kata yang muncul pada pesan ham sering muncul di pesan spam, contohnya pada pesan "Brainless Baby Doll..:-D;-), vehicle sariyag drive madoke barolla..".
- Pesan yang ambigu, tidak jelas, terlalu umum dan kerap menggunakan singkatan. Contohnya pada pesan "i dnt wnt to tlk wid u", ini mungkin berarti pesan sungguhan dari seseorang yang merasa kesal pada penerima sehingga menggunakan bahasa yang disingkat-singkat, tetapi model malah mengklasifikasikannya menjadi spam.