Prediksi Keterlambatan Pengiriman Paket

Disusun untuk memenuhi tugas besar pada mata kuliah: Data Science

Dosen Pengampu: Yopi Hidayatul Akbar, S.Kom., M.T

Disusun oleh:

Agung Febrian	(220660121086)
Kemal Hapidz Prastiawan	(220660121115)
Dede Yayan Suciyana	(220660121179)

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS SEBELAS APRIL

SUMEDANG

2025

A. Ringkasan Kasus

Dalam era digital yang serba cepat, layanan ekspedisi memiliki peran vital dalam mendukung aktivitas e-commerce dan distribusi logistik. Ketepatan waktu pengiriman menjadi indikator utama dalam menilai kepuasan pelanggan terhadap layanan yang diberikan. Namun, dalam praktiknya, keterlambatan pengiriman masih kerap terjadi karena berbagai faktor, seperti kondisi operasional, karakteristik barang, hingga strategi diskon yang diterapkan.

Untuk menjawab tantangan tersebut, analisis ini diarahkan untuk mengidentifikasi faktor-faktor signifikan yang memengaruhi keterlambatan pengiriman serta membangun model prediktif yang mampu memperkirakan apakah suatu pengiriman akan terlambat atau tidak. Dengan memanfaatkan data historis pengiriman, diharapkan solusi ini dapat meningkatkan efisiensi operasional serta membantu pengambilan keputusan yang lebih tepat dalam manajemen logistik.

B. Metode Analisis

Langkah-langkah analisis yang dilakukan:

a. *Data Cleaning & Preprocessing*

- Duplikat data dihapus
- Kolom kategorial seperti: `Warehouse_block`, `Mode_of_Shipment`, dan `Product_importance diencoding` menggunakan `LabelEncoder`
- Standarisasi dilakukan menggunakan `StandardScaler`
- Target prediksi adalah `Reached.on.Time_Y.N` (0: terlambat, 1: tepat waktu)

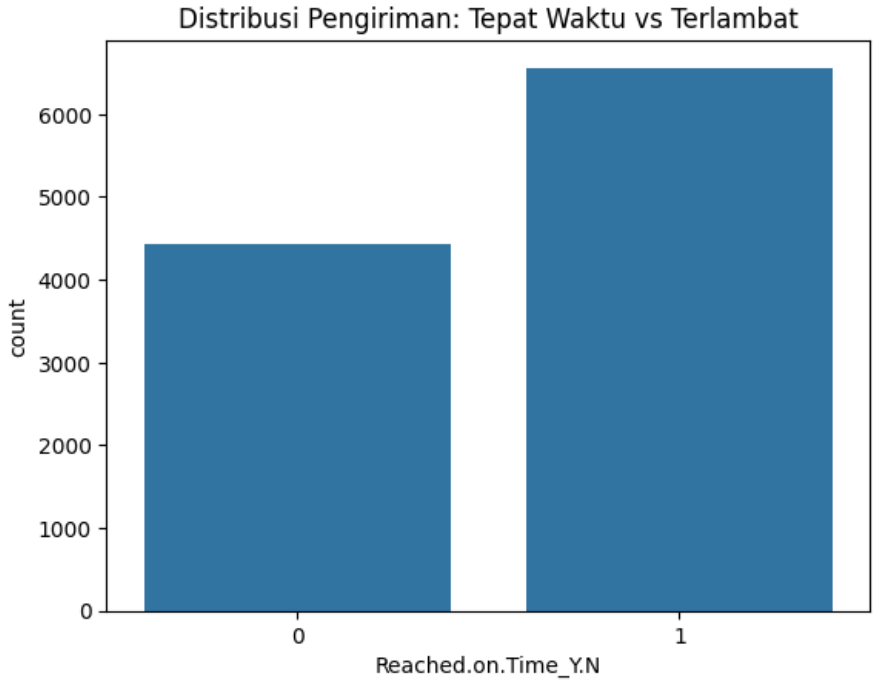
```python
# 2.1 Hapus duplikat dan cek nilai kosong
df = df.drop_duplicates()
print("Missing Values per Kolom:\n", df.isnull().sum())

# 2.2 Label Encoding kolom kategorikal
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
for col in ['Warehouse_block', 'Mode_of_Shipment', 'Product_importance', 'Gender']:
    df[col] = le.fit_transform(df[col])

# 2.3 Pisahkan fitur dan target
X = df.drop(['ID', 'Reached.on.Time_Y.N'], axis=1)
y = df['Reached.on.Time_Y.N']

# 2.4 Scaling fitur numerik
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

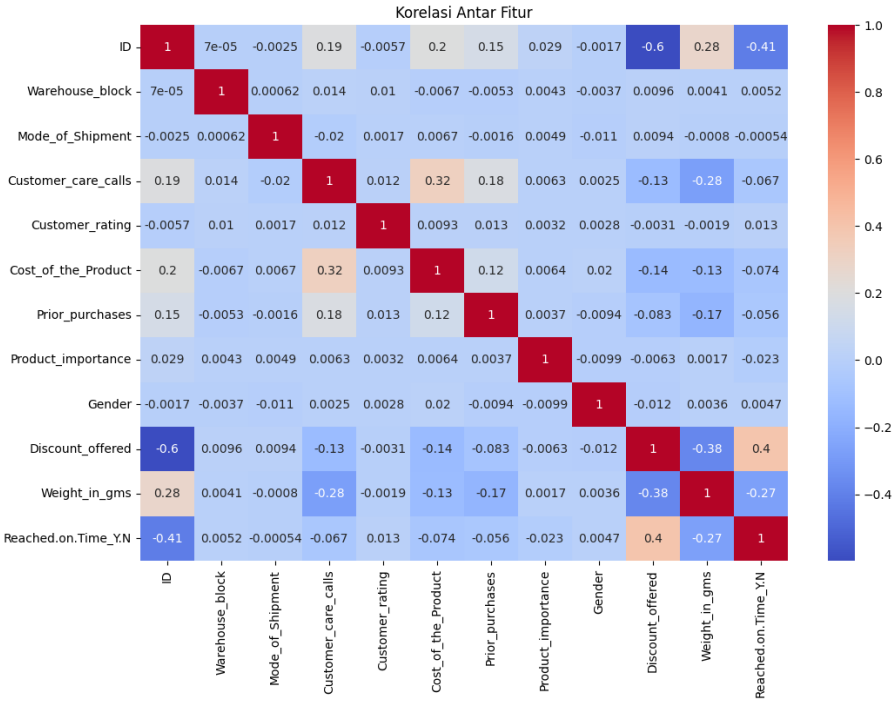
```
Missing Values per Kolom:
 ID                     0
Warehouse_block         0
Mode_of_Shipment        0
Customer_care_calls     0
Customer_rating         0
Cost_of_the_Product     0
Prior_purchases         0
Product_importance      0
Gender                  0
Discount_offered        0
Weight_in_gms           0
Reached.on.Time_Y.N     0
dtype: int64
```

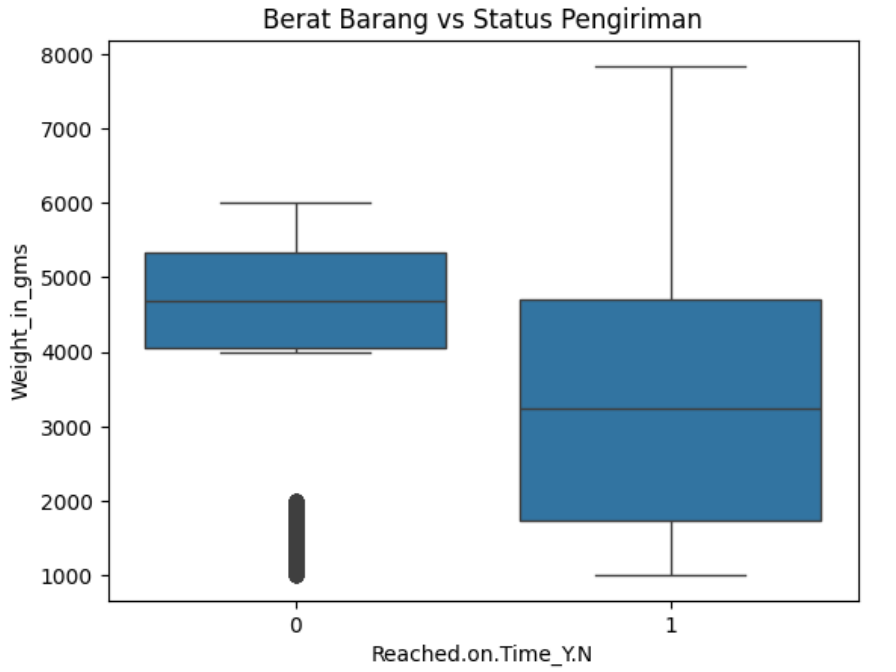
b. *Exploratory Data Analysis* (EDA)

- Visualisasi distribusi keterlambatan pengiriman
- Analisis korelasi antara fitur-fitur (diskon, berat, rating)
- *Boxplot* menunjukkan bahwa paket dengan berat tertentu lebih rentan terlambat

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

# 3.1 Visualisasi distribusi target
sns.countplot(x='Reached.on.Time_Y.N', data=df)
plt.title("Distribusi Pengiriman: Tepat Waktu vs Terlambat")
plt.show()

# 3.2 Heatmap korelasi antar fitur
plt.figure(figsize=(12, 8))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
plt.title("Korelasi Antar Fitur")
plt.show()

# 3.3 Berat barang vs keterlambatan (boxplot)
sns.boxplot(x='Reached.on.Time_Y.N', y='Weight_in_gms', data=df)
plt.title("Berat Barang vs Status Pengiriman")
plt.show()
```



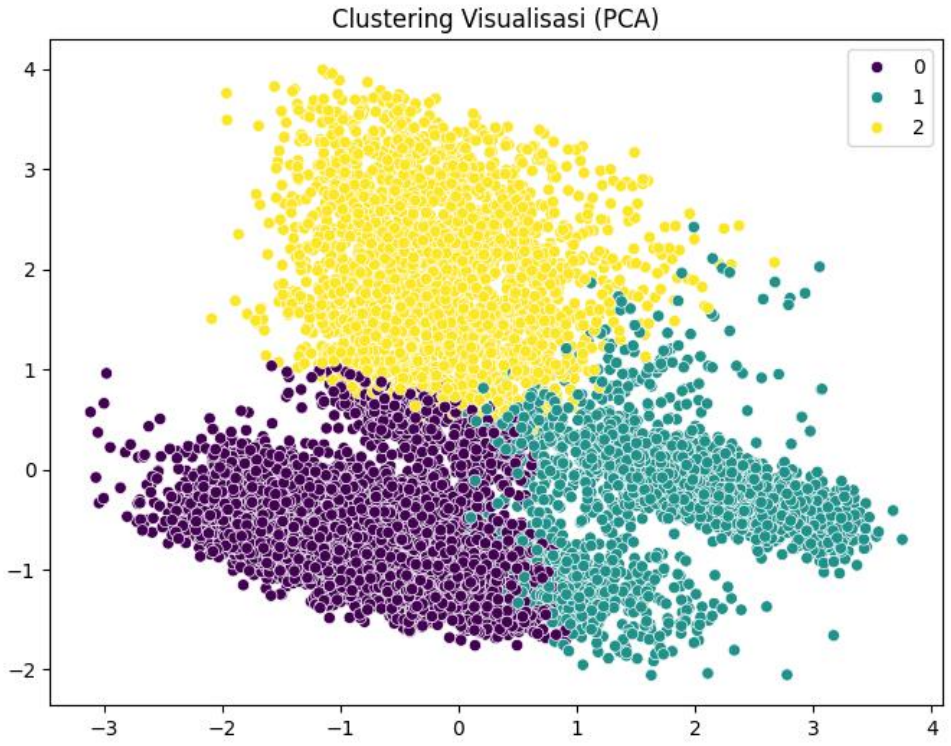




c. Segmentasi Pelanggan (*Clustering*)

- PCA digunakan untuk reduksi dimensi agar *clustering* dapat divisualisasikan
- *Clustering* dilakukan menggunakan K-Means dengan 3 klaster
- Visualisasi menunjukkan pola yang dapat dimanfaatkan untuk segmentasi operasional

```python
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA

# 4.1 Reduksi dimensi untuk visualisasi dengan PCA
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)

# 4.2 Clustering dengan KMeans
kmeans = KMeans(n_clusters=3, random_state=42)
clusters = kmeans.fit_predict(X_scaled)
df['Cluster'] = clusters

# 4.3 Visualisasi hasil clustering
plt.figure(figsize=(8, 6))
sns.scatterplot(x=X_pca[:, 0], y=X_pca[:, 1], hue=clusters, palette='viridis')
plt.title("Clustering Visualisasi (PCA)")
plt.show()
```



d. Model Prediksi (*Supervised Learning*)

- Data dibagi menjadi 80% data latih dan 20% data uji
- Model *Random Forest* digunakan karena kemampuannya menangani fitur kategorikal dan numerik
- Model dilatih untuk memprediksi apakah paket akan terlambat atau tidak

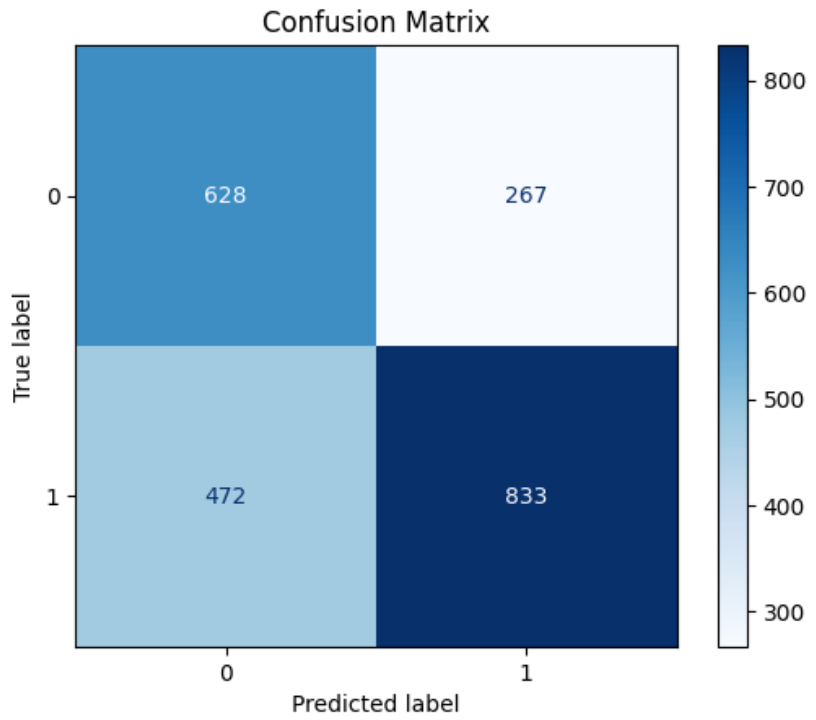
```python
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier

# 5.1 Split Data untuk Training & Testing
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

# 5.2 Latih Model Random Forest
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

e. Evaluasi Model

- Model menghasilkan metrik:
    1. *Accuracy*: (0.66)
    2. *Precision*, *Recall*, F1-Score ditampilkan menggunakan `classification_report`
- Confusion matrix divisualisasikan untuk memahami kesalahan prediksi

```python
from sklearn.metrics import classification_report, accuracy_score
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

# 6.1 Classification Report
print(classification_report(y_test, y_pred))

# 6.2 Accuracy Model
accuracy = accuracy_score(y_test, y_pred)
print(f"Akurasi Model: {accuracy:.2}")

# 6.3 Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm)
disp.plot(cmap='Blues')
plt.title("Confusion Matrix")
plt.show()
```

```
              precision    recall  f1-score   support

           0       0.57      0.70      0.63       895
           1       0.76      0.64      0.69      1305

    accuracy                           0.66      2200
   macro avg       0.66      0.67      0.66      2200
weighted avg       0.68      0.66      0.67      2200

Akurasi Model: 0.66
```



f. Visualisasi & Interpretasi

Berdasarkan hasil eksplorasi dan visualisasi data (heatmap korelasi, distribusi keterlambatan, dan proporsi moda pengiriman), diperoleh beberapa insight penting untuk mendukung strategi bisnis logistik, antara lain:

1. **Diskon Besar Meningkatkan Risiko Keterlambatan**

   Ditemukan korelasi positif antara `Discount_offered` dengan keterlambatan. Artinya, semakin besar diskon yang diberikan, semakin tinggi kemungkinan pengiriman mengalami keterlambatan.

   Strategi: Diskon besar sebaiknya tidak diberikan pada produk berat atau daerah rawan keterlambatan. Perlu kebijakan promosi berbasis kapasitas logistik.

2. **Berat Paket Mempengaruhi Ketepatan Waktu**

   Paket dengan `Weight_in_gms` yang tinggi lebih sering terlambat dibandingkan paket ringan.

   Strategi: Segmentasikan pengiriman berdasarkan berat. Produk berat dapat diprioritaskan menggunakan moda pengiriman yang lebih andal atau disiapkan lebih awal.

3. **Moda Pengiriman Dominan dan Risiko**

Moda `Ship` merupakan yang paling banyak digunakan, namun jika dikaitkan dengan tingkat keterlambatan, perlu perhatian lebih.

Strategi: Evaluasi performa mitra logistik untuk moda pengiriman dominan dan lakukan renegosiasi SLA (*Service Level Agreement*) jika diperlukan.

4. **Peluang Segmentasi Pelanggan dan Produk**

   Hasil clustering menunjukkan pola berbeda antar kelompok transaksi.

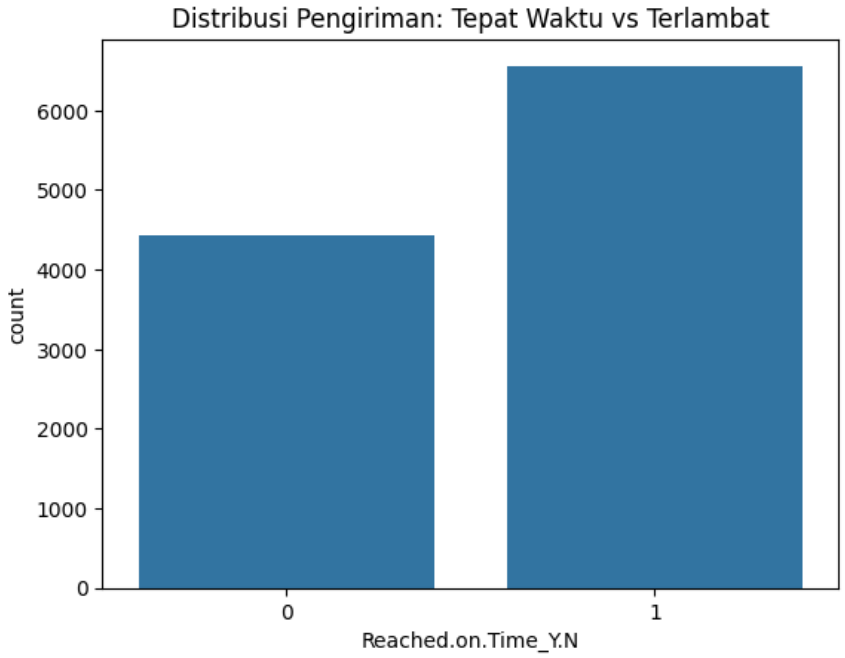
   Strategi: Terapkan kebijakan logistik atau layanan yang berbeda untuk tiap segmen, misalnya penanganan khusus untuk klaster berisiko tinggi keterlambatan.

C. Hasil Analisis dan Model

a. Eksplorasi Data

1. Distribusi Keterlambatan

- Grafik *countplot* memperlihatkan distribusi antara pengiriman yang tepat waktu (1) dan terlambat (0).
- Terlihat bahwa sebagian besar pengiriman berada di kategori tepat waktu, namun keterlambatan tetap signifikan secara jumlah.

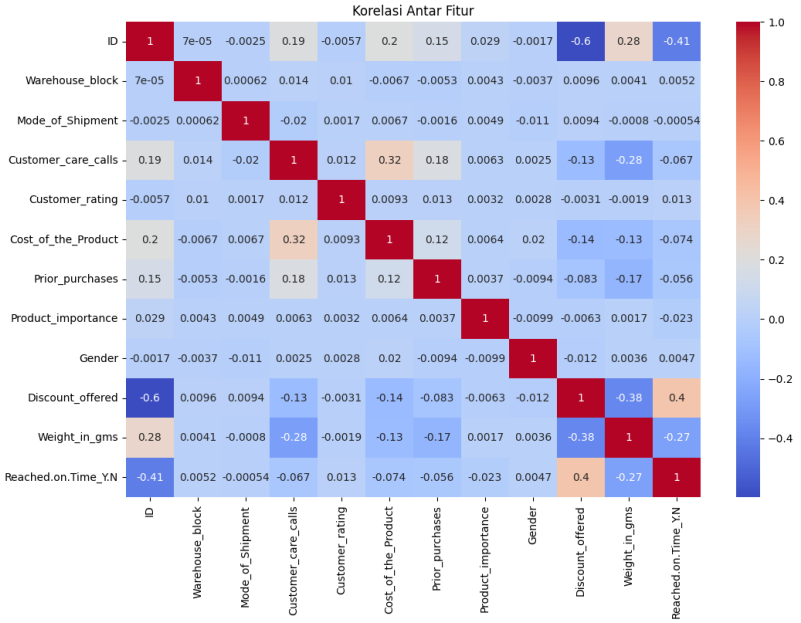


2. Korelasi Antar Fitur

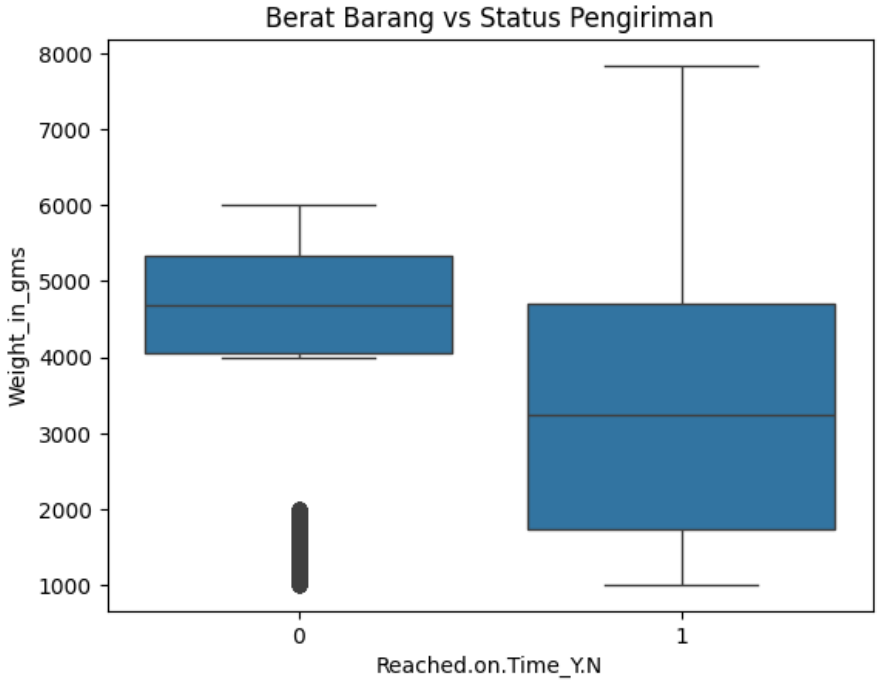
- *Heatmap* menunjukkan fitur yang paling berkorelasi dengan keterlambatan (Reached.on.Time_Y.N) adalah:
  - `Discount_offered` (positif)

- `Weight_in_gms` (positif)

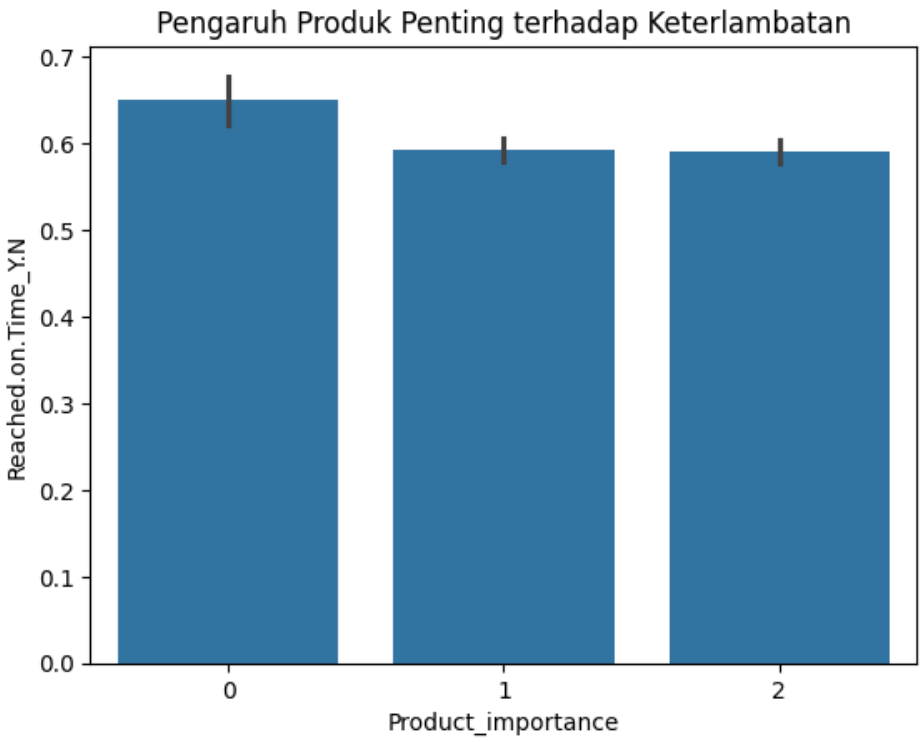
- Ini menunjukkan bahwa diskon besar dan berat paket tinggi meningkatkan risiko keterlambatan.



3. Berat Paket & Status Pengiriman

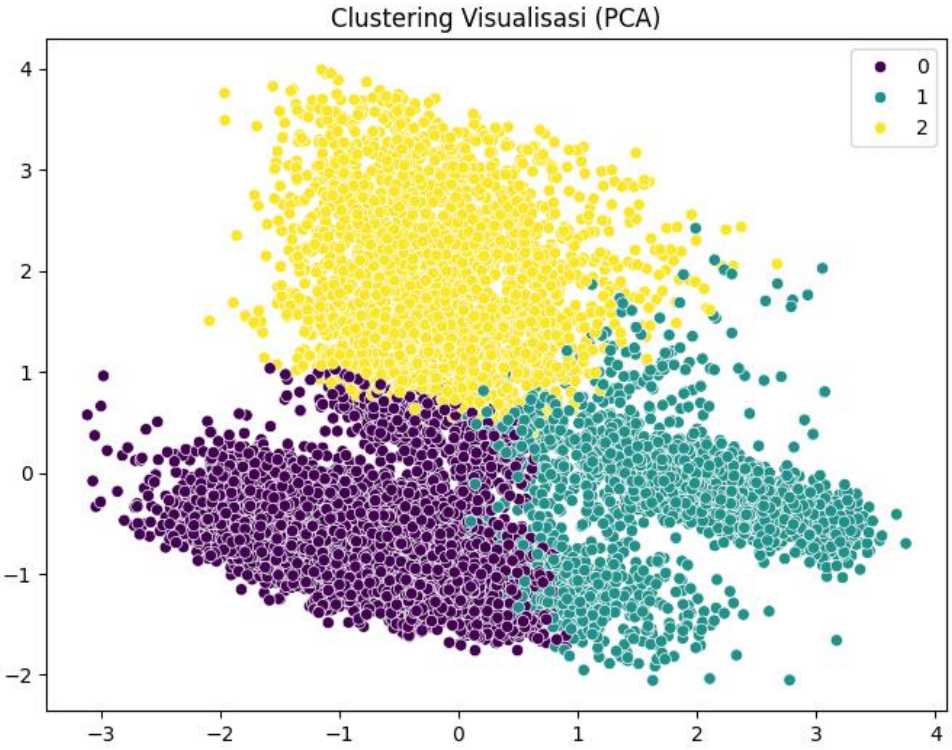


4. *Importance* Produk vs Keterlambatan



b. *Clustering* (Segmentasi Transaksi)

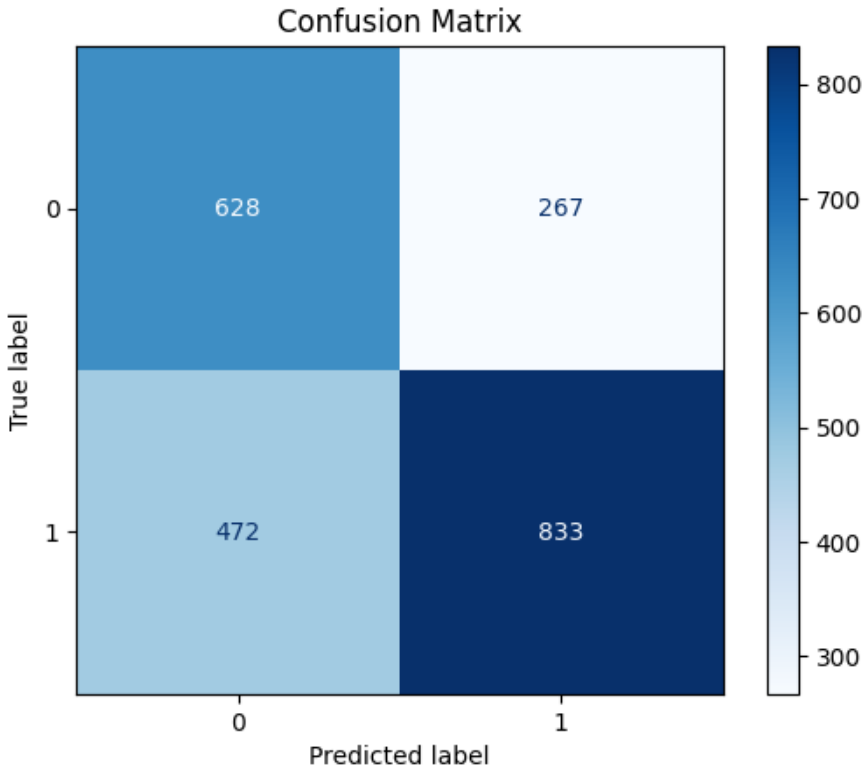
- *Clustering* dilakukan menggunakan K-Means (k = 3).
- Data direduksi menggunakan PCA agar dapat divisualisasikan dalam 2 dimensi.
- Setiap klaster merepresentasikan pola pembelian/logistik yang serupa.



c. Model Prediksi (*Random Forest*)

- Model: *Random Forest Classifier*
- Data dibagi menjadi 80% data latih dan 20% data uji.
- Fitur penting: `Discount_offered`, `Weight_in_gms`, `Product_importance`, dsb.
- Hasil prediksi dievaluasi dengan Classification Report dan Confusion Matrix.

	precision	recall	f1-score	support
0	0.57	0.70	0.63	895
1	0.76	0.64	0.69	1305
accuracy			0.66	2200
macro avg	0.66	0.67	0.66	2200
weighted avg	0.68	0.66	0.67	2200



D. Kesimpulan dan Saran

a. Kesimpulan

- Model Random Forest menghasilkan akurasi sebesar 66% atau 0,66 dengan F1-score yang cukup baik untuk kedua kelas keterlambatan dan ketepatan waktu.
- Model lebih mampu mengenali pengiriman yang terlambat daripada yang tepat waktu, ditunjukkan oleh precision dan recall yang lebih tinggi pada kelas 1.
- Fitur `Discount_offered` dan `Weight_in_gms` menunjukkan korelasi yang signifikan terhadap kemungkinan keterlambatan.
- Hasil *clustering* memetakan transaksi ke dalam beberapa kelompok yang menunjukkan pola risiko logistik berbeda, yang dapat dijadikan acuan segmentasi.

b. Saran

- Perlu peningkatan kualitas model, misalnya:
  - Menggunakan algoritma lain seperti XGBoost,
  - Melakukan hyperparameter tuning, dan
  - Mengatasi ketidakseimbangan data dengan teknik seperti SMOTE.
- Model ini sebaiknya digunakan sebagai alat bantu awal, bukan keputusan mutlak, karena masih terdapat cukup banyak *false negative*.
- Disarankan melakukan analisis lanjutan terhadap fitur waktu jika tersedia (misalnya waktu pengiriman, tanggal order) untuk meningkatkan akurasi.
- Perusahaan dapat menggunakan hasil prediksi ini untuk memfilter risiko keterlambatan lebih awal dan menyesuaikan strategi pengiriman pada produk dengan risiko tinggi.

E. *Tools* dan *Library*

1. Python (pandas, seaborn, matplotlib, scikit-learn)
2. Google Colab: Pada link berikut: colab/kelompok-14
3. Dataset: *E-Commerce Shipping* Data (Kaggle/@prachi13)

F. *Repository*

github.com/kelomppok-14