



Nama Peserta	:	Mory Handy So
Nomor Urut	:	TUK-035.017765

DAFTAR ISI

Contents

DAFTAR ISI.....	2
BUKTI 1-ADS	4
Kebutuhan Data.....	4
1. Identifikasi Kebutuhan Data	4
2. Memeriksa Ketersediaan Data	5
Hasil Pemeriksaan Data	5
Pengambilan Data.....	6
Identifikasi Metode dan Tools Pengambilan Data	6
Implementasi Langkah-langkah Pengambilan Data	6
Pemeriksaan Integritas Data	8
BUKTI 2-ADS	10
Analisis Tipe dan Relasi Data	10
Nilai Atribut Data Sesuai Konteks Bisnis	11
Identifikasi Relasi Antar Data	11
Penjelasan:	12
Analisis Karakteristik Data	13
Analisis Karakteristik Data	15
BUKTI 3-ADS	16
Pengecekan Kelengkapan Data	17
Rekomendasi Kelengkapan Data	19
Rekomendasi Hasil Penilaian Kecukupan Data	19
BUKTI 4-ADS	20
Kriteria dan Teknik Pemilihan Data.....	20
Teknik Pemilihan Data	20
Implementasi Teknik Pemilihan Data	21
Attributes (Columns) dan Records (Row) Data.....	22
Atribut yang Dipilih	22
Identifikasi Records (Baris) Data	23
Records yang Dipilih	23
BUKTI 5-ADS	23
Pembersihan Data Kotor	24

Strategi Pembersihan Data.....	24
Deskripsi Masalah dan Teknis Koreksi Data	25
BUKTI 6-ADS	26
Analisis Teknik Transformasi Data	26
Analisis Representasi Fitur Data Awal	29
Teknik Rekayasa Fitur yang Diperlukan	30
BUKTI 7-ADS	30
Pelabelan Data	31
BUKTI 8-ADS	32
Modelling	32
BUKTI 9-ADS	35
Penggunaan Model dengan Data Riil.....	37

BUKTI 1-ADS

Kode Unit	:	J.62DMI00.004.1
Judul Unit	:	Mengumpulkan Data

Deskripsi:

Bukti ini berhubungan dengan pengetahuan, keterampilan, dan sikap kerja yang dibutuhkan dalam mengumpulkan data untuk data science.

Langkah Kerja:

- 1) Menentukan kebutuhan data
- 2) Mengambil data
- 3) Mengintegrasikan data

Peralatan dan Perlengkapan:

- Peralatan
 - Komputer
- Perlengkapan
 - Aplikasi pengubah teks
 - Aplikasi basis data
 - Tools pengambilan data

Kebutuhan Data

1. Identifikasi Kebutuhan Data

Tujuan teknis dari data science biasanya mencakup beberapa aspek berikut:

- **Prediksi Churn (Pelanggan Berhenti Berlangganan):** Untuk memprediksi apakah pelanggan akan berhenti berlangganan layanan.
- **Analisis Faktor Penyebab:** Mengidentifikasi faktor-faktor yang paling berpengaruh terhadap churn.
- **Segmentasi Pelanggan:** Mengelompokkan pelanggan berdasarkan karakteristik tertentu.
- **Analisis Retensi:** Mengembangkan strategi untuk meningkatkan retensi pelanggan.

Berdasarkan tujuan ini, kita membutuhkan data yang mencakup:

- **Informasi Demografis Pelanggan:** Usia, jenis kelamin, status pernikahan, dll.
- **Data Layanan yang Digunakan:** Jenis layanan yang diambil, durasi berlangganan, dll.
- **Data Interaksi Pelanggan:** Jumlah keluhan, frekuensi penggunaan layanan, dll.
- **Data Keuangan:** Tagihan bulanan, pembayaran terakhir, dll.
- **Label Churn:** Indikator apakah pelanggan berhenti berlangganan atau tidak.

2. Memeriksa Ketersediaan Data

Mari kita memeriksa ketersediaan data dalam file yang telah Anda unggah untuk memastikan data yang diperlukan tersedia.

```
# Load the dataset
file_path = 'Telco-Customer-Churn.csv'
data = pd.read_csv(file_path)

# Display the first few rows and the column names
data.head(), data.columns
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	DeviceProtection	TechSupport	StreamingTV	StreamingMovies	Contract	PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCharges	Churn
0	7590-VHVEG	F	0	Yes	No	1.0	No	No phone service	DSL	No	No	No	No	No	Month-to-month	Yes	Electronic check	29.85	29.85	No
1	5575-GNVDE	Male	0	No	No	34.0	Yes	No	DSL	Yes	Yes	No	No	No	One year	No	Mailed check	56.95	1889.5	No
2	3668-QPYBK	Male	0	No	No	2.0	Yes	No	DSL	Yes	No	No	No	No	Month-to-month	Yes	Mailed check	53.85	108.15	Yes
3	7795-CFOCW	Male	0	No	No	45.0	No	No phone service	DSL	Yes	Yes	Yes	No	No	One year	No	Bank transfer (automatic)	42.30	1840.75	No
4	9237-HQITU	Female	0	No	No	2.0	Yes	No	Fiber optic	No	No	No	No	No	Month-to-month	Yes	Electronic check	70.70	151.65	Yes

```
[5 rows x 21 columns],
Index(['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents',
      'tenure', 'PhoneService', 'MultipleLines', 'InternetService',
      'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport',
      'StreamingTV', 'StreamingMovies', 'Contract', 'PaperlessBilling',
      'PaymentMethod', 'MonthlyCharges', 'TotalCharges', 'Churn'],
      dtype='object'))
```

Hasil Pemeriksaan Data

1. Ketersediaan Data Dataset yang telah Anda unggah memiliki 21 kolom dan 7043 baris. Berikut adalah beberapa kolom yang tersedia:

- **Informasi Pelanggan:** customerID, gender, SeniorCitizen, Partner, Dependents
- **Layanan:** PhoneService, MultipleLines, InternetService, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies
- **Kontrak dan Pembayaran:** Contract, PaperlessBilling, PaymentMethod, MonthlyCharges, TotalCharges
- **Tenure:** tenure
- **Label Churn:** Churn

2. Volume Data Dataset ini memiliki 7043 baris data, yang cukup besar untuk tujuan prediksi churn, analisis faktor penyebab, segmentasi pelanggan, dan analisis retensi.

Pengambilan Data

Identifikasi Metode dan Tools Pengambilan Data

Berdasarkan tujuan teknis data science yang telah disebutkan (prediksi churn, analisis faktor penyebab, segmentasi pelanggan, dan analisis retensi), kita dapat mengidentifikasi metode dan tools pengambilan data yang sesuai.

Metode Pengambilan Data

1. Pembersihan dan Transformasi Data:

- **Handling Missing Values:** Mengatasi nilai yang hilang dengan pengisian nilai rata-rata atau median, atau dengan menghapus baris/kolom yang hilang.
- **Encoding Kategorikal Data:** Mengonversi data kategorikal menjadi bentuk numerik menggunakan one-hot encoding atau label encoding.
- **Scaling Data:** Normalisasi atau standarisasi data numerik.

Tools Pengambilan Data

- **Python:** Digunakan untuk melakukan pembersihan dan transformasi data.
 - Libraries: pandas, numpy, scikit-learn
- **Google Colab:** Untuk menulis dan menjalankan kode secara interaktif.

Implementasi Langkah-langkah Pengambilan Data

1. Pembersihan dan Transformasi Data

```
[4] df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
#   Column                Non-Null Count  Dtype  
---  --
0   customerID            7043 non-null   object  
1   gender                7038 non-null   object  
2   SeniorCitizen         7043 non-null   int64   
3   Partner              7043 non-null   object  
4   Dependents            7043 non-null   object  
5   tenure               7040 non-null   float64  
6   PhoneService         7043 non-null   object  
7   MultipleLines         7043 non-null   object  
8   InternetService       7043 non-null   object  
9   OnlineSecurity        7043 non-null   object  
10  OnlineBackup          7043 non-null   object  
11  DeviceProtection      7043 non-null   object  
12  TechSupport           7043 non-null   object  
13  StreamingTV           7043 non-null   object  
14  StreamingMovies       7043 non-null   object  
15  Contract              7043 non-null   object  
16  PaperlessBilling      7043 non-null   object  
17  PaymentMethod         7043 non-null   object  
18  MonthlyCharges        7043 non-null   float64  
19  TotalCharges          7043 non-null   object  
20  Churn                 7043 non-null   object  
dtypes: float64(2), int64(1), object(18)
memory usage: 1.1+ MB
```

```
import pandas as pd

# Handling missing values
# Fill missing gender with mode
df['gender'].fillna(df['gender'].mode()[0], inplace=True)

# Fill missing tenure with median
df['tenure'].fillna(df['tenure'].median(), inplace=True)

# Convert TotalCharges to numeric, forcing errors to NaN (to handle possible non-numeric values)
df['TotalCharges'] = pd.to_numeric(df['TotalCharges'], errors='coerce')

# Check for any NaN values in TotalCharges and handle them
df['TotalCharges'].fillna(df['TotalCharges'].median(), inplace=True)

# Convert SeniorCitizen to boolean
df['SeniorCitizen'] = df['SeniorCitizen'].astype(bool)

# Convert appropriate columns to 'category' dtype
category_cols = ['gender', 'Partner', 'Dependents', 'PhoneService', 'MultipleLines', 'InternetService',
                  'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV',
                  'StreamingMovies', 'Contract', 'PaperlessBilling', 'PaymentMethod', 'Churn']

for col in category_cols:
    df[col] = df[col].astype('category')

# Display the first few rows of the cleaned and transformed dataset
print(df.head())
print(df.info())
```

```
[5 rows x 21 columns]
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
#   Column                Non-Null Count  Dtype
---  -
0   customerID            7043 non-null   object
1   gender                 7043 non-null   category
2   SeniorCitizen          7043 non-null   bool
3   Partner                7043 non-null   category
4   Dependents             7043 non-null   category
5   tenure                 7043 non-null   float64
6   PhoneService           7043 non-null   category
7   MultipleLines           7043 non-null   category
8   InternetService         7043 non-null   category
9   OnlineSecurity          7043 non-null   category
10  OnlineBackup            7043 non-null   category
11  DeviceProtection        7043 non-null   category
12  TechSupport             7043 non-null   category
13  StreamingTV             7043 non-null   category
14  StreamingMovies         7043 non-null   category
15  Contract                7043 non-null   category
16  PaperlessBilling        7043 non-null   category
17  PaymentMethod           7043 non-null   category
18  MonthlyCharges          7043 non-null   float64
19  TotalCharges            7043 non-null   float64
20  Churn                   7043 non-null   category
dtypes: bool(1), category(16), float64(3), object(1)
memory usage: 339.3+ KB
None
```

Dataset telah berhasil dibersihkan dan ditransformasi. Berikut langkah-langkah yang telah dilakukan:

1. **Mengatasi Missing Values:**
 - o Kolom gender yang hilang diisi dengan nilai yang paling sering muncul (mode).
 - o Kolom tenure yang hilang diisi dengan nilai median.
2. **Mengubah Tipe Data:**
 - o Kolom TotalCharges berhasil diubah menjadi tipe data numerik dan nilai yang tidak valid diisi dengan nilai median.
 - o Kolom SeniorCitizen diubah menjadi tipe data boolean.
3. **Transformasi Data:**
 - o Kolom-kolom kategoris diubah menjadi tipe category untuk efisiensi memori.

Pemeriksaan Integritas Data

- Memeriksa Integritas Data:

- Memeriksa apakah ada nilai duplikat.
- Memeriksa apakah ada nilai yang tidak valid.
- Memeriksa distribusi nilai pada kolom yang relevan.

- Integrasi Data:

- Mengkonversi kolom kategori menjadi data numerik jika diperlukan untuk model machine learning.
- Menyimpan dataset yang telah dibersihkan dan ditransformasi ke file baru.

```
import pandas as pd

# Handling missing values
# Fill missing gender with mode
df['gender'].fillna(df['gender'].mode()[0], inplace=True)
df['gender'] = df['gender'].replace('M', 'Male').replace('F', 'Female')

# Fill missing tenure with median
df['tenure'].fillna(df['tenure'].median(), inplace=True)

# Convert TotalCharges to numeric, forcing errors to NaN (to handle possible non-numeric values)
df['TotalCharges'] = pd.to_numeric(df['TotalCharges'], errors='coerce')

# Check for any NaN values in TotalCharges and handle them
df['TotalCharges'].fillna(df['TotalCharges'].median(), inplace=True)

# Convert SeniorCitizen to boolean
df['SeniorCitizen'] = df['SeniorCitizen'].astype(bool)

# Convert appropriate columns to 'category' dtype
category_cols = ['gender', 'Partner', 'Dependents', 'PhoneService', 'MultipleLines', 'InternetService',
                 'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV',
                 'StreamingMovies', 'Contract', 'PaperlessBilling', 'PaymentMethod', 'Churn']

for col in category_cols:
    df[col] = df[col].astype('category')

# Display the first few rows of the cleaned and transformed dataset
print(df.head())
print(df.info())
```

```
customerID  gender  SeniorCitizen  Partner  Dependents  tenure  PhoneService \
0  7590-VHVEG  Female             False    Yes         No       1.0         No
1  5575-GNVDE   Male             False    No          No      34.0         Yes
2  3668-QPYBK   Male             False    No          No       2.0         Yes
3  7795-CFOCWI  Male             False    No          No      45.0         No
4  9237-HQITU  Female             False    No          No       2.0         Yes

MultipleLines  InternetService  OnlineSecurity  ...  DeviceProtection \
0  No phone service           DSL               No  ...              No
1           No                DSL               Yes  ...              Yes
2           No                DSL               Yes  ...              No
3  No phone service           DSL               Yes  ...              Yes
4           No          Fiber optic              No  ...              No

TechSupport  StreamingTV  StreamingMovies  Contract  PaperlessBilling \
0           No           No                No  Month-to-month      Yes
1           No           No                No    One year        No
2           No           No                No  Month-to-month      Yes
3           Yes           No                No    One year        No
4           No           No                No  Month-to-month      Yes

PaymentMethod  MonthlyCharges  TotalCharges  Churn
0  Electronic check           29.85         29.85    No
1      Mailed check           56.95       1889.50    No
2      Mailed check           53.85        108.15   Yes
3  Bank transfer (automatic)    42.30       1840.75    No
4  Electronic check           70.70        151.65   Yes
```



```

# Check for duplicates
duplicates = df.duplicated().sum()

# Removing duplicates if any
df.drop_duplicates(inplace=True)

# Check for invalid values or outliers
summary_stats = df.describe(include='all')

# Convert categorical columns to numeric using one-hot encoding for integration with data science models
df_encoded = pd.get_dummies(df, drop_first=True)

# Save the cleaned and transformed dataset to a new CSV file
output_path = 'Telco-Customer-Churn-Cleaned.csv'
df_encoded.to_csv(output_path, index=False)

# Display results
summary_stats, duplicates, df_encoded.head()

```

```

( customerID gender SeniorCitizen Partner Dependents tenure \
count 7043 7043 7043 7043 7043 7043.000000
unique 7043 2 2 2 2 NaN
top 7590-VHVEG Male False No No NaN
freq 1 3558 5901 3641 4933 NaN
mean NaN NaN NaN NaN NaN 35.041318
std NaN NaN NaN NaN NaN 115.250379
min NaN NaN NaN NaN NaN 0.000000
25% NaN NaN NaN NaN NaN 9.000000
50% NaN NaN NaN NaN NaN 29.000000
75% NaN NaN NaN NaN NaN 55.000000
max NaN NaN NaN NaN NaN 7100.000000

PhoneService MultipleLines InternetService OnlineSecurity ... \
count 7043 7043 7043 7043 ...
unique 2 3 3 3 ...
top Yes No Fiber optic No ...
freq 6361 3390 3096 3498 ...
mean NaN NaN NaN NaN ...
std NaN NaN NaN NaN ...
min NaN NaN NaN NaN ...
25% NaN NaN NaN NaN ...
50% NaN NaN NaN NaN ...
75% NaN NaN NaN NaN ...
max NaN NaN NaN NaN ...

DeviceProtection TechSupport StreamingTV StreamingMovies \
count 7043 7043 7043 7043 ...
unique 3 3 3 3 ...
top No No No No ...
freq 3095 3473 2810 2785 ...
mean NaN NaN NaN NaN ...
std NaN NaN NaN NaN ...
min NaN NaN NaN NaN ...
25% NaN NaN NaN NaN ...
50% NaN NaN NaN NaN ...
75% NaN NaN NaN NaN ...
max NaN NaN NaN NaN ...

Contract PaperlessBilling PaymentMethod MonthlyCharges \
count 7043 7043 7043 7043.000000
unique 3 2 4 NaN
top Month-to-month Yes Electronic check NaN
freq 3875 4171 2365 NaN
mean NaN NaN NaN 64.761692
std NaN NaN NaN 30.090047
min NaN NaN NaN 18.250000
25% NaN NaN NaN 35.500000
50% NaN NaN NaN 70.350000
75% NaN NaN NaN 89.850000
max NaN NaN NaN 118.750000

PaymentMethod_credit card (automatic) PaymentMethod_Electronic check \
0 False False True
1 False False False
2 False False False
3 False False False
4 False True

PaymentMethod_Mailed check Churn_Yes
0 False False
1 True False
2 True True
3 False False
4 False True

[5 rows x 7073 columns]

```

- Mengisi nilai yang hilang pada kolom gender dan tenure.
- Mengubah tipe data pada kolom TotalCharges dan SeniorCitizen.
- Konversi kolom kategori menjadi tipe category.
- Memeriksa dan menghapus duplikat.
- Menyimpan dataset yang telah dibersihkan ke file baru dalam bentuk yang siap untuk analisis data science.

BUKTI 2-ADS

Kode Unit	:	J.62DMI00.005.1
Judul Unit	:	Menelaah Data

Deskripsi:

Bukti ini berhubungan dengan pengetahuan, keterampilan, dan sikap kerja yang dibutuhkan dalam menelaah data untuk data science.

Langkah Kerja:

- 1) Menganalisis tipe dan relasi data
- 2) Menganalisis karakteristik data
- 3) Membuat laporan telaah data

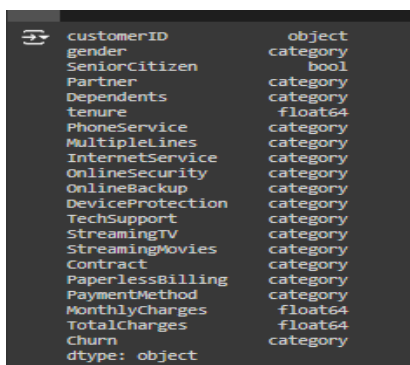
Peralatan dan Perlengkapan:

- Peralatan
 - Komputer
- Perlengkapan
 - Aplikasi pengolah kata
 - Tools pengolahan data
 - Tools pembuat grafik

Analisis Tipe dan Relasi Data

Identifikasi Tipe Data yang Terkumpul:

- Tentukan tipe data untuk setiap kolom.



customerID	object
gender	category
SeniorCitizen	bool
Partner	category
Dependents	category
tenure	float64
PhoneService	category
MultipleLines	category
InternetService	category
OnlineSecurity	category
OnlineBackup	category
DeviceProtection	category
TechSupport	category
StreamingTV	category
StreamingMovies	category
Contract	category
PaperlessBilling	category
PaymentMethod	category
MonthlyCharges	float64
TotalCharges	float64
Churn	category
dtype:	object

Uraikan Nilai Atribut Data:

- Uraikan dan jelaskan nilai-nilai atribut utama dalam konteks bisnis.

Identifikasi Relasi Antar Data:

- Analisis hubungan antar atribut yang relevan untuk mencapai tujuan teknis.

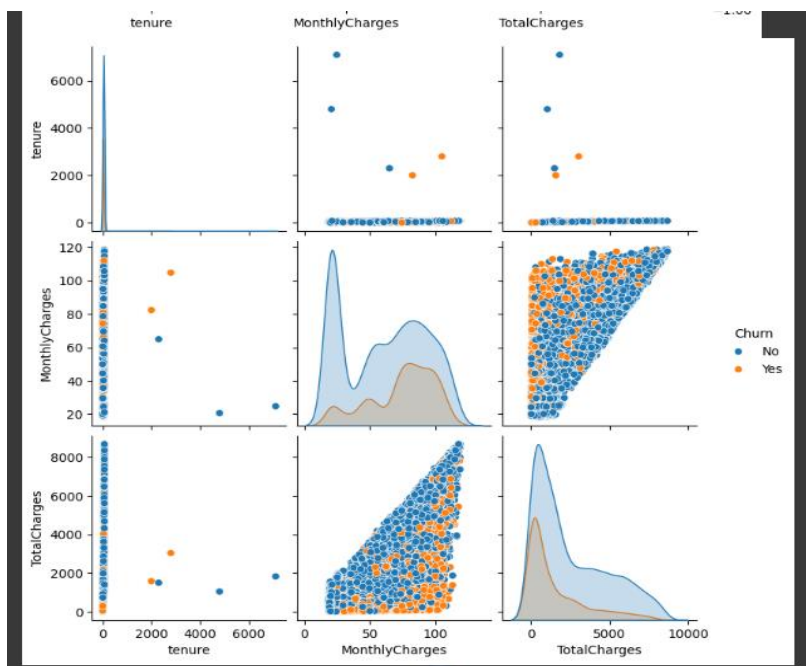
Nilai Atribut Data Sesuai Konteks Bisnis

Berikut adalah beberapa atribut penting beserta nilai-nilai yang mungkin mereka ambil dalam konteks bisnis:

- **gender:** (Male, Female) – Jenis kelamin pelanggan.
- **SeniorCitizen:** (0, 1) – Apakah pelanggan adalah warga senior (1) atau bukan (0).
- **Partner:** (Yes, No) – Apakah pelanggan memiliki pasangan atau tidak.
- **Dependents:** (Yes, No) – Apakah pelanggan memiliki tanggungan atau tidak.
- **tenure:** (0-72) – Lama waktu pelanggan telah bersama perusahaan dalam bulan.
- **PhoneService:** (Yes, No) – Apakah pelanggan memiliki layanan telepon atau tidak.
- **MultipleLines:** (No phone service, No, Yes) – Apakah pelanggan memiliki banyak jalur telepon atau tidak.
- **InternetService:** (DSL, Fiber optic, No) – Jenis layanan internet yang digunakan oleh pelanggan.
- **OnlineSecurity:** (Yes, No, No internet service) – Apakah pelanggan memiliki layanan keamanan online atau tidak.
- **OnlineBackup:** (Yes, No, No internet service) – Apakah pelanggan memiliki layanan backup online atau tidak.
- **DeviceProtection:** (Yes, No, No internet service) – Apakah pelanggan memiliki layanan perlindungan perangkat atau tidak.
- **TechSupport:** (Yes, No, No internet service) – Apakah pelanggan memiliki layanan dukungan teknis atau tidak.
- **StreamingTV:** (Yes, No, No internet service) – Apakah pelanggan memiliki layanan TV streaming atau tidak.
- **StreamingMovies:** (Yes, No, No internet service) – Apakah pelanggan memiliki layanan film streaming atau tidak.
- **Contract:** (Month-to-month, One year, Two year) – Jenis kontrak pelanggan.
- **PaperlessBilling:** (Yes, No) – Apakah pelanggan menggunakan penagihan tanpa kertas atau tidak.
- **PaymentMethod:** (Electronic check, Mailed check, Bank transfer (automatic), Credit card (automatic)) – Metode pembayaran yang digunakan oleh pelanggan.
- **MonthlyCharges:** (0-118.75) – Biaya bulanan yang dibayar oleh pelanggan.
- **TotalCharges:** (0-8684.8) – Total biaya yang dibayar oleh pelanggan selama ini.
- **Churn:** (Yes, No) – Apakah pelanggan berhenti berlangganan (churn) atau tidak.

Identifikasi Relasi Antar Data

Untuk mengidentifikasi relasi antar data, kita dapat menggunakan analisis korelasi dan visualisasi.



Penjelasan:

1. **Identifikasi Tipe Data:**
 - o `df.dtypes` memberikan informasi tipe data untuk setiap kolom.
2. **Uraikan Nilai Atribut Data:**
 - o Memberikan konteks bisnis dan nilai yang mungkin untuk atribut utama dalam dataset.

3. Identifikasi Relasi Antar Data:

- Menggunakan matriks korelasi untuk fitur numerik untuk memahami hubungan antar fitur numerik.
- Menggunakan visualisasi pair plot untuk melihat hubungan antara fitur numerik dan kategoris, serta dampaknya terhadap Churn.

Untuk tujuan prediksi churn dan analisis faktor penyebab, kita perlu memahami relasi antar data berikut:

- **Relasi antara tenure dan churn:** Lama berlangganan sering kali berhubungan dengan kemungkinan pelanggan untuk berhenti berlangganan.
- **Relasi antara MonthlyCharges/TotalCharges dan churn:** Tingginya biaya bulanan atau total biaya bisa menjadi faktor penyebab churn.
- **Relasi antara jenis kontrak dan churn:** Pelanggan dengan kontrak jangka panjang mungkin kurang cenderung untuk berhenti berlangganan dibandingkan dengan pelanggan dengan kontrak bulanan.
- **Relasi antara layanan yang digunakan (InternetService, OnlineSecurity, dll.) dan churn:** Pelanggan yang menggunakan lebih banyak layanan mungkin lebih setia.
- **Relasi antara demografi pelanggan (gender, SeniorCitizen, Partner, Dependents) dan churn:** Demografi tertentu mungkin memiliki kecenderungan churn yang lebih tinggi atau lebih rendah.

Analisis Karakteristik Data

Deskripsi Statistik Dasar

```
[31] # Descriptive statistics for numerical columns
numeric_summary = df.describe()

# Descriptive statistics for categorical columns
categorical_summary = df.describe(include=['category'])

numeric_summary, categorical_summary
```

```
(
  count    tenure  MonthlyCharges  TotalCharges
  mean    35.041318    64.761692    2281.916928
  std     115.258379    30.090047    2265.270398
  min       0.000000    18.250000    18.800000
  25%       9.000000    35.500000    402.225000
  50%      29.000000    70.350000    1397.475000
  75%      55.000000    89.850000    3786.600000
  max     7100.000000   118.750000   8684.800000
  gender Partner Dependents PhoneService MultipleLines InternetService \
  count    7043    7043    7043    7043    7043    7043
  unique     2     2     2     2     3     3
  top    Male  No  No  Yes  No  Fiber optic
  freq    3558  3641  4933  6361  3390  3096

  OnlineSecurity OnlineBackup DeviceProtection TechSupport StreamingTV \
  count    7043    7043    7043    7043    7043
  unique     3     3     3     3     3
  top      No  No  No  No  No
  freq    3498  3088  3095  3473  2810

  StreamingMovies Contract PaperlessBilling PaymentMethod \
  count    7043    7043    7043    7043
  unique     3     3     2     4
  top      No  Month-to-month  Yes  Electronic check
  freq    2785    3875    4171    2365

  Churn
  count    7043
  unique     2
  top      No
  freq    5174 )
```

Visualisasi Grafik

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

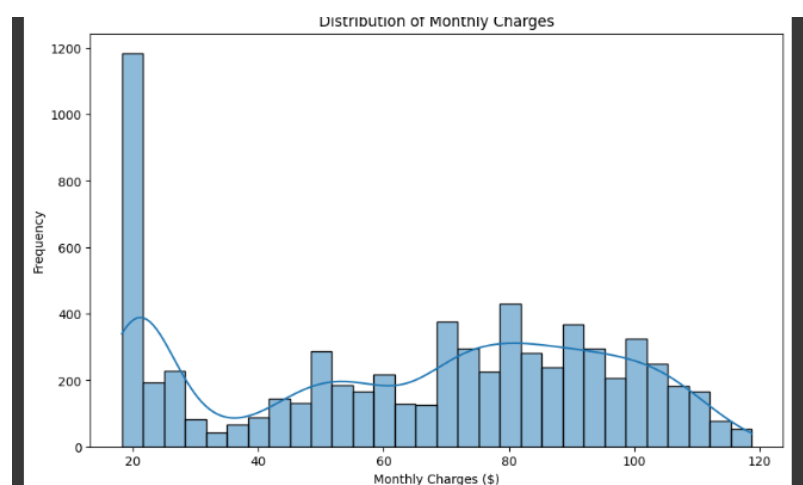
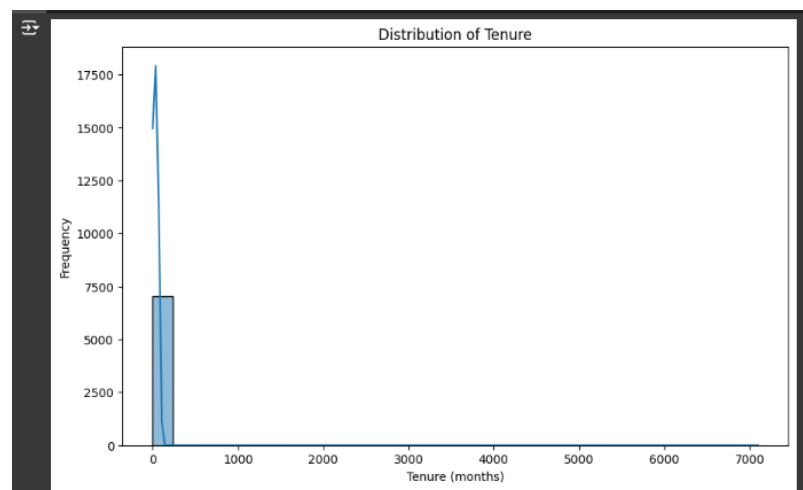
# Visualize the distribution of tenure
plt.figure(figsize=(10, 6))
sns.histplot(df['tenure'], bins=30, kde=True)
plt.title('Distribution of Tenure')
plt.xlabel('Tenure (months)')
plt.ylabel('Frequency')
plt.show()

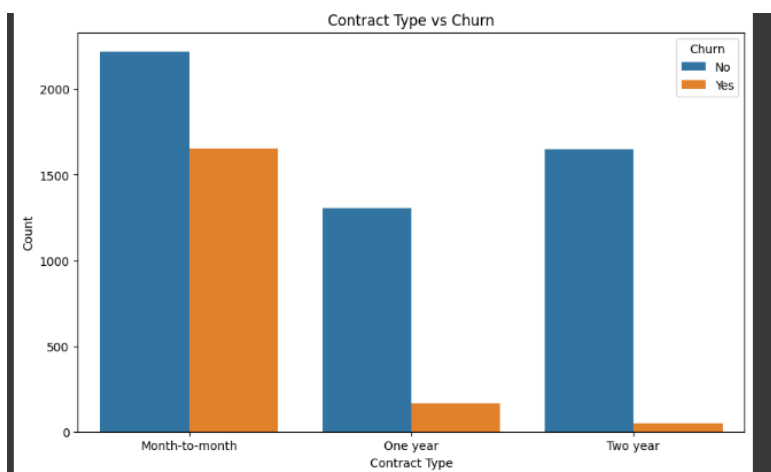
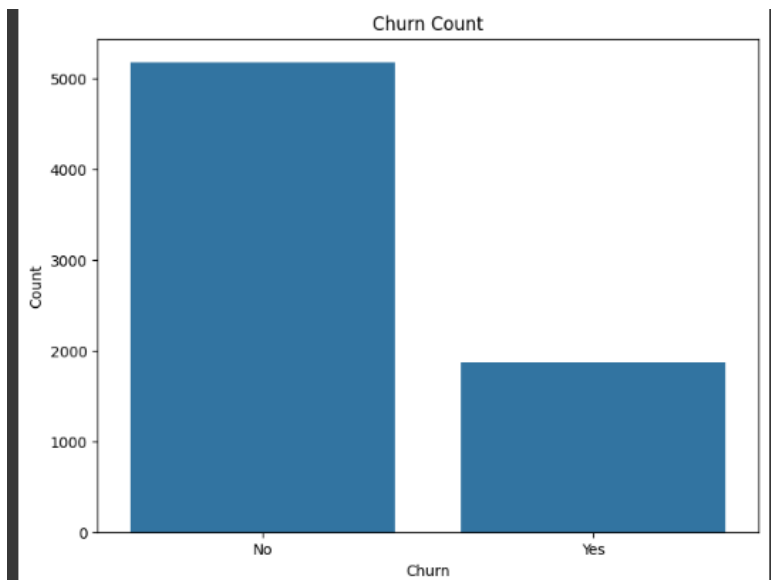
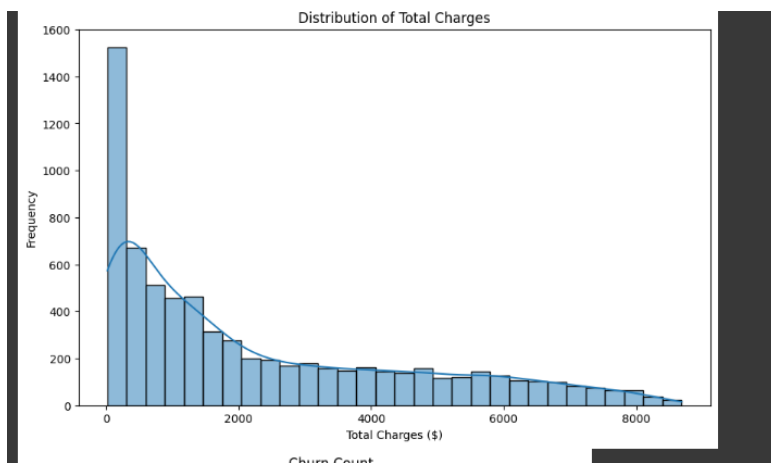
# Visualize the distribution of MonthlyCharges
plt.figure(figsize=(10, 6))
sns.histplot(df['MonthlyCharges'], bins=30, kde=True)
plt.title('Distribution of Monthly charges')
plt.xlabel('Monthly charges ($)')
plt.ylabel('Frequency')
plt.show()

# Visualize the distribution of TotalCharges
plt.figure(figsize=(10, 6))
sns.histplot(df['TotalCharges'], bins=30, kde=True)
plt.title('Distribution of Total Charges')
plt.xlabel('Total Charges ($)')
plt.ylabel('Frequency')
plt.show()

# Count plot for Churn
plt.figure(figsize=(8, 6))
sns.countplot(data=df, x='Churn')
plt.title('Churn Count')
plt.xlabel('Churn')
plt.ylabel('Count')
plt.show()

# Count plot for Contract type
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='Contract', hue='Churn')
plt.title('Contract Type vs Churn')
plt.xlabel('Contract Type')
plt.ylabel('count')
plt.show()
```





Analisis Karakteristik Data

Setelah menyajikan deskripsi statistik dasar dan visualisasi grafik, kita dapat melakukan analisis karakteristik data sebagai berikut:

- **Tenure:**
 - Distribusi tenure menunjukkan bahwa sebagian besar pelanggan memiliki masa berlangganan yang relatif pendek.
 - Ini bisa menunjukkan bahwa banyak pelanggan baru atau pelanggan cenderung berhenti berlangganan dalam waktu singkat.
- **Monthly Charges:**
 - Distribusi MonthlyCharges menunjukkan variasi yang cukup besar dalam biaya bulanan yang dibayar pelanggan.
 - Pelanggan dengan biaya bulanan yang lebih tinggi mungkin lebih cenderung untuk berhenti berlangganan jika mereka merasa biaya tersebut tidak sebanding dengan layanan yang diterima.
- **Total Charges:**
 - Distribusi TotalCharges juga menunjukkan variasi yang luas, mencerminkan akumulasi biaya selama masa berlangganan.
- **Churn:**
 - Proporsi pelanggan yang berhenti berlangganan (Churn) menunjukkan bahwa churn merupakan masalah yang signifikan bagi perusahaan.
- **Contract Type vs Churn:**
 - Visualisasi hubungan antara tipe kontrak dan churn menunjukkan bahwa pelanggan dengan kontrak bulanan cenderung lebih tinggi tingkat churn-nya dibandingkan dengan pelanggan dengan kontrak satu tahun atau dua tahun.
 - Ini menunjukkan bahwa kontrak jangka panjang mungkin membantu mengurangi tingkat churn.

BUKTI 3-ADS

Kode Unit	:	J.62DMI00.006.1
Judul Unit	:	Memvalidasi Data

Deskripsi:

Bukti ini berhubungan dengan pengetahuan, keterampilan, dan sikap kerja yang dibutuhkan dalam memvalidasi data untuk data science.

Langkah Kerja:

- 1) Melakukan pengecekan kelengkapan data
- 2) Membuat rekomendasi kelengkapan data

Peralatan dan Perlengkapan:

- Peralatan
 - Komputer
- Perlengkapan
 - Aplikasi pengubah teks

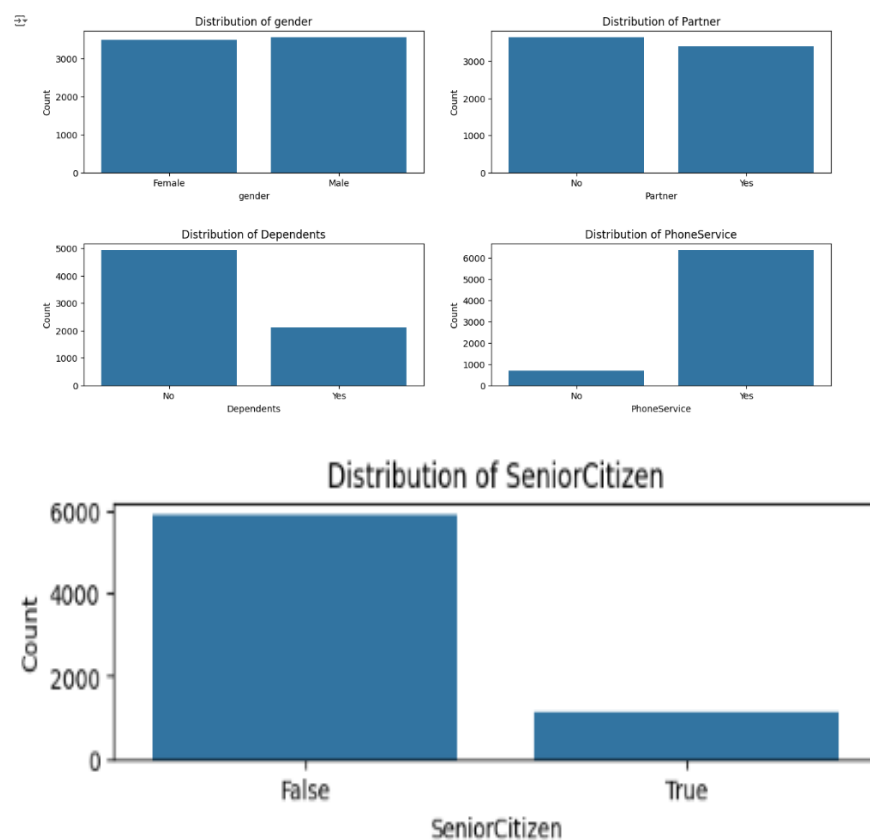
Pengecekan Kelengkapan Data

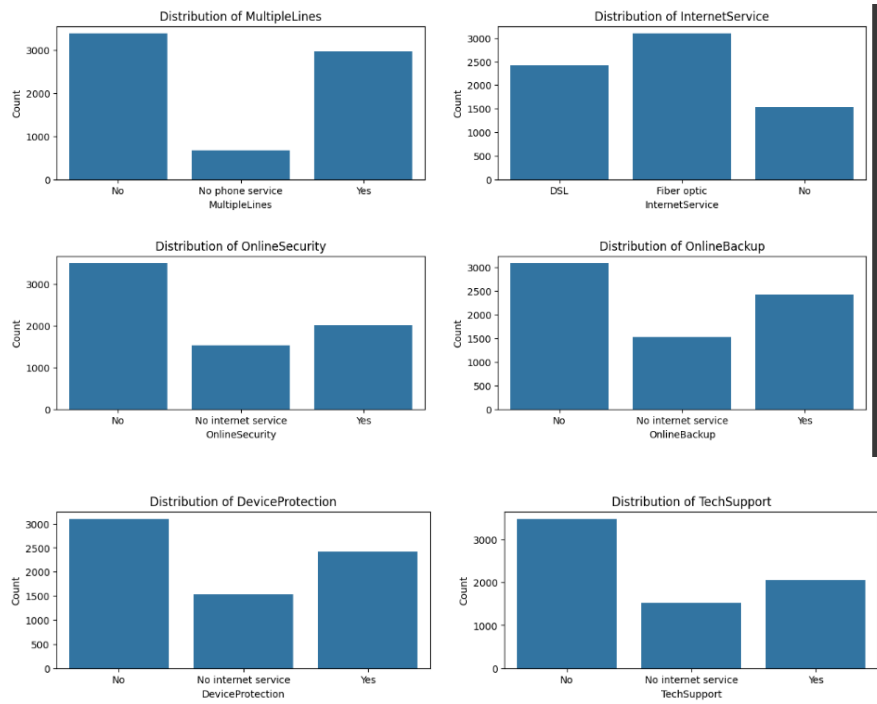
```
# Periksa nilai yang hilang
missing_values = df.isnull().sum()

# Periksa tipe data setiap kolom
data_types = df.dtypes

missing_values, data_types
```

```
(customerID      0
 gender          0
 SeniorCitizen  0
 Partner        0
 Dependents     0
 tenure         0
 PhoneService   0
 MultipleLines  0
 InternetService 0
 OnlineSecurity 0
 OnlineBackup   0
 DeviceProtection 0
 TechSupport    0
 StreamingTV    0
 StreamingMovies 0
 Contract       0
 PaperlessBilling 0
 PaymentMethod  0
 MonthlyCharges 0
 TotalCharges   0
 Churn          0
 dtype: int64,
 customerID      object
 gender          category
 SeniorCitizen   bool
 Partner        category
 Dependents     category
 tenure         float64
 PhoneService   category
 MultipleLines  category
 InternetService category
 OnlineSecurity category
 OnlineBackup   category
 DeviceProtection category
 TechSupport    category
 StreamingTV    category
 StreamingMovies category
 Contract       category
 PaperlessBilling category
 PaymentMethod  category
 MonthlyCharges float64
 TotalCharges   float64
 churn         category
 dtype: object)
```





Penilaian Kualitas Data

1. Kelengkapan Data:

- Tidak ada nilai yang hilang setelah pengisian yang dilakukan pada kolom gender, tenure, dan TotalCharges.
- Tipe data setiap kolom telah sesuai setelah konversi yang diperlukan (TotalCharges menjadi numerik, SeniorCitizen menjadi boolean, dan kolom kategori menjadi category).

Penilaian Tingkat Kecukupan Data

1. Distribusi Data Kategoris:

- Sebagian besar kolom kategoris memiliki distribusi yang cukup seimbang, meskipun ada beberapa kategori yang mungkin kurang terwakili (misalnya, kategori tertentu dalam InternetService atau PaymentMethod).
- Visualisasi Churn menunjukkan bahwa proporsi pelanggan yang berhenti berlangganan (Yes) cukup signifikan, yang penting untuk analisis churn.

2. Distribusi Data Numerik:

- Kolom tenure menunjukkan bahwa banyak pelanggan memiliki masa berlangganan yang relatif singkat.
- Kolom MonthlyCharges dan TotalCharges menunjukkan variasi yang cukup besar dalam biaya yang dibayarkan pelanggan.

Kesimpulan

- **Kualitas Data:** Data telah memenuhi kelengkapan dan memiliki tipe data yang sesuai untuk analisis lebih lanjut.
- **Tingkat Kecukupan Data:** Distribusi data menunjukkan bahwa dataset ini cukup representatif untuk analisis churn, meskipun beberapa kategori mungkin memerlukan perhatian khusus dalam analisis lebih lanjut.

Rekomendasi Kelengkapan Data

Berdasarkan penilaian kualitas data yang telah dilakukan, berikut adalah rekomendasi untuk memastikan kelengkapan data sesuai dengan tujuan teknis data science:

1. Kelengkapan Data:

- **Mengisi Nilai yang Hilang:**
 - Pastikan untuk terus memantau nilai yang hilang pada dataset baru yang masuk dan isi nilai yang hilang sesuai dengan strategi yang sudah digunakan (mode untuk kategoris, median untuk numerik).
- **Validasi Tipe Data:**
 - Lakukan validasi rutin terhadap tipe data setiap kolom untuk memastikan bahwa tipe data tetap konsisten dan sesuai dengan yang diharapkan.
- **Data Entry Checks:**
 - Implementasikan pengecekan data otomatis saat data dimasukkan ke dalam sistem untuk mencegah data yang tidak valid atau hilang.

2. Kebersihan Data:

- **Penghapusan Duplikat:**
 - Secara rutin periksa dan hapus duplikat data untuk memastikan tidak ada pengulangan data yang dapat mempengaruhi analisis.
- **Normalisasi Data:**
 - Pastikan data numerik telah dinormalisasi (jika diperlukan) untuk analisis atau pemodelan yang lebih akurat.

Rekomendasi Hasil Penilaian Kecukupan Data

Berdasarkan penilaian tingkat kecukupan data, berikut adalah rekomendasi untuk memastikan kecukupan data sesuai dengan tujuan teknis data science:

1. Distribusi Data Kategoris:

- **Perimbangan Kategori:**
 - Lakukan oversampling atau undersampling pada kategori yang kurang terwakili untuk memastikan bahwa model machine learning dapat belajar dengan baik dari setiap kategori.
- **Analisis Lebih Lanjut pada Kategori Tertentu:**
 - Lakukan analisis tambahan pada kategori yang kurang terwakili untuk memahami karakteristik dan dampaknya terhadap churn atau variabel target lainnya.

2. Ukuran Dataset:

- **Penambahan Data:**
 - Jika memungkinkan, tambahkan lebih banyak data untuk meningkatkan representasi dan kekayaan informasi dalam dataset.
- **Validasi Eksternal:**
 - Gunakan data eksternal (jika tersedia) untuk memvalidasi hasil dan memastikan bahwa model yang dibangun dapat digeneralisasi ke populasi yang lebih luas.

3. Kualitas Data Numerik:

- **Penanganan Outliers:**
 - Identifikasi dan tangani outliers dalam data numerik yang dapat mempengaruhi analisis atau pemodelan. Pertimbangkan apakah outliers tersebut adalah data yang valid atau kesalahan data entry.
- **Transformasi Data:**
 - Pertimbangkan untuk melakukan transformasi data (seperti log-transform) pada kolom dengan distribusi yang sangat miring untuk analisis yang lebih baik.

BUKTI 4-ADS

Kode Unit	:	J.62DMI00.007.1
Judul Unit	:	Menentukan Objek Data

Deskripsi:

Bukti ini berhubungan dengan pengetahuan, keterampilan, dan sikap kerja yang dibutuhkan dalam memilah dan memilih data yang sesuai permintaan atau kebutuhan.

Langkah Kerja:

- 1) Memutuskan kriteria dan teknik pemilihan data
- 2) Menentukan attributes (columns) dan records (row) data

Peralatan dan Perlengkapan:

- Peralatan
 - Komputer
- Perlengkapan
 - Aplikasi pengolah kata
 - Aplikasi spreadsheet
 - Aplikasi notepad plus
 - Aplikasi SQL (Structured Query Language)

Kriteria dan Teknik Pemilihan Data

- 1. Relevansi terhadap Tujuan Analisis:**
 - Pilih data yang secara langsung terkait dengan masalah yang sedang dianalisis, seperti prediksi churn pelanggan.
 - Pilih variabel yang memiliki potensi hubungan yang signifikan dengan variabel target (Churn).
- 2. Kualitas Data:**
 - Pastikan data yang dipilih bebas dari kesalahan dan missing values yang tidak dapat diatasi.
 - Pilih data yang sudah melalui proses validasi dan pembersihan.
- 3. Kelengkapan Data:**
 - Pilih data yang memiliki rekam jejak yang lengkap, dengan nilai-nilai yang terisi penuh atau dapat diisi dengan cara yang tepat.
 - Pastikan bahwa jumlah sampel cukup untuk analisis statistik yang signifikan.
- 4. Kepatuhan terhadap Regulasi:**
 - Pastikan data yang digunakan mematuhi peraturan privasi dan perlindungan data, seperti GDPR.
 - Hindari penggunaan data yang bersifat pribadi atau sensitif tanpa persetujuan yang diperlukan.

Teknik Pemilihan Data

Setelah menetapkan kriteria pemilihan data, langkah selanjutnya adalah menentukan teknik pemilihan data yang sesuai. Berikut adalah beberapa teknik yang dapat digunakan:

1. Filtering Berdasarkan Kondisi:

- o Memilih subset data berdasarkan kondisi tertentu yang relevan dengan analisis.
- o Contoh: Memilih data pelanggan yang masih aktif ($\text{tenure} > 0$).

2. Handling Missing Values:

- o Mengisi missing values dengan nilai yang tepat (mean, median, mode, atau teknik imputasi lainnya) atau menghapus baris yang tidak lengkap jika diperlukan.
- o Contoh: Mengisi TotalCharges dengan median atau mean jika ada missing values.

3. Sampling:

- o Menggunakan teknik sampling untuk memilih subset data yang representatif dari populasi.
- o Contoh: Menggunakan stratified sampling untuk memastikan setiap kategori di Churn terwakili dengan baik.

4. Feature Selection:

- o Memilih fitur-fitur yang relevan dan memiliki potensi hubungan yang signifikan dengan variabel target.
- o Contoh: Menggunakan teknik statistik seperti korelasi atau metode machine learning seperti Random Forest untuk menentukan fitur yang penting.

Implementasi Teknik Pemilihan Data

Berikut adalah implementasi teknik pemilihan data sesuai dengan kriteria yang telah ditetapkan.

1. Filtering Berdasarkan Kondisi

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from imblearn.over_sampling import SMOTE

# Select only relevant columns for analysis
selected_attributes = [
    'gender', 'SeniorCitizen', 'Partner', 'Dependent', 'tenure', 'PhoneService',
    'MultiLines', 'InternetService', 'OnlineSecurity', 'OnlineBackup',
    'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMusic',
    'Contract', 'PaperlessBilling', 'PaymentMethod', 'MonthlyCharges',
    'TotalCharges', 'Churn'
]
df_selected = df[selected_attributes]

# One-hot encoding for categorical features
categorical_features = categorical_cols[1:] # Excluding 'Churn'
onehot_encoder = OneHotEncoder(drop='first')

# Scaling for numerical features
numerical_features = ['tenure', 'MonthlyCharges', 'TotalCharges']
scaler = StandardScaler()

# Column transformer for combining preprocessing
preprocessor = ColumnTransformer(
    transformers=[
        ('num', scaler, numerical_features),
        ('cat', onehot_encoder, categorical_features)
    ])

# Pipeline for preprocessing
pipeline = Pipeline(steps=[('preprocessor', preprocessor)])

# Preprocessing data
X = df_selected.drop(columns=['Churn'])
y = df_selected['Churn'].cat.codes

# Apply preprocessing pipeline
X_preprocessed = pipeline.fit_transform(X)

# Handling imbalanced data with SMOTE
smote = SMOTE(random_state=42)
X_resampled, y_resampled = smote.fit_resample(X_preprocessed, y)

# Feature Selection using Random Forest
model = RandomForestClassifier(random_state=42)
model.fit(X_resampled, y_resampled)

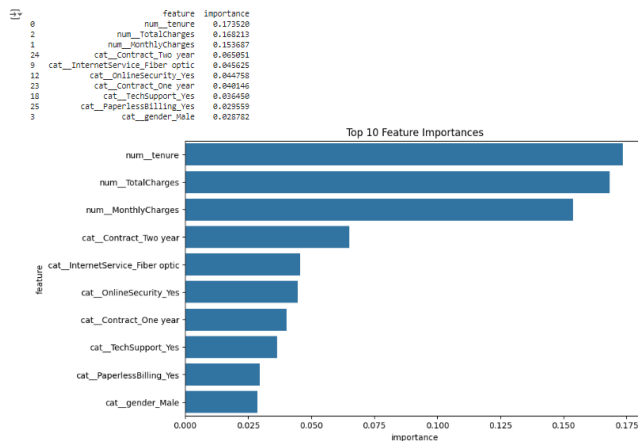
# Getting feature importances
feature_importances = model.feature_importances_

# Get the column names after preprocessing
preprocessed_feature_names = pipeline.named_steps['preprocessor'].get_feature_names_out()

# Create a DataFrame for feature importances
feature_importances_df = pd.DataFrame({
    'features': preprocessed_feature_names,
    'importance': feature_importances
}).sort_values(by='importance', ascending=False)

# Display top 10 features
top_features = feature_importances_df.head(10)
print(top_features)

# Plotting feature importances
plt.figure(figsize=(10, 6))
sns.barplot(x=feature_importances_df['features'], y=feature_importances_df['importance'])
plt.title('Top 10 Feature Importances')
plt.show()
```



Attributes (Columns) dan Records (Row) Data

Atribut-atribut ini harus memenuhi kriteria relevansi, kualitas, dan kelengkapan, serta mematuhi aturan regulasi yang berlaku.

1. **Relevansi terhadap Analisis Churn:**
 - Atribut yang memiliki hubungan signifikan dengan churn pelanggan.
 - Atribut yang sering digunakan dalam analisis churn pada domain industri yang sama.
2. **Kualitas Data:**
 - Atribut yang bebas dari nilai yang hilang atau telah diisi dengan nilai yang tepat.
 - Atribut yang telah melalui proses validasi dan pembersihan.
3. **Kelengkapan Data:**
 - Atribut yang memiliki rekam jejak lengkap dan tidak banyak missing values.
4. **Kepatuhan terhadap Regulasi:**
 - Atribut yang tidak melanggar privasi atau peraturan perlindungan data.

Atribut yang Dipilih

Berikut adalah daftar atribut yang dipilih berdasarkan kriteria di atas:

1. gender
2. SeniorCitizen
3. Partner
4. Dependents
5. tenure
6. PhoneService
7. MultipleLines
8. InternetService
9. OnlineSecurity
10. OnlineBackup
11. DeviceProtection
12. TechSupport
13. StreamingTV
14. StreamingMovies
15. Contract
16. PaperlessBilling
17. PaymentMethod

18. MonthlyCharges
19. TotalCharges
20. Churn (target variable)

Identifikasi Records (Baris) Data

Berdasarkan kriteria pemilihan data yang telah ditetapkan, kita akan memilih record yang memenuhi kriteria relevansi, kualitas, dan kelengkapan.

1. **Relevansi terhadap Analisis Churn:**
 - Record pelanggan yang aktif dan memiliki informasi lengkap terkait churn.
2. **Kualitas Data:**
 - Record yang bebas dari missing values yang tidak dapat diatasi.
 - Record yang telah melalui proses pembersihan data.
3. **Kelengkapan Data:**
 - Record yang memiliki data lengkap pada atribut yang dipilih.

Records yang Dipilih

Berikut adalah langkah-langkah untuk memilih records yang memenuhi kriteria:

1. **Mengisi Nilai yang Hilang:**
 - Mengisi missing values dengan strategi yang telah ditentukan.
2. **Menghapus Record yang Tidak Lengkap:**
 - Menghapus record yang masih memiliki missing values setelah proses imputasi.
3. **Filtering Berdasarkan Kondisi:**
 - Memilih record pelanggan yang masih aktif ($\text{tenure} > 0$).

BUKTI 5-ADS

Kode Unit	:	J.62DMI00.008.1
Judul Unit	:	Membersihkan Data

Deskripsi:

Bukti ini berhubungan dengan pengetahuan, keterampilan, dan sikap kerja yang dibutuhkan dalam membersihkan data yang sesuai permintaan atau kebutuhan.

Langkah Kerja:

- 1) Melakukan pembersihan data yang kotor
- 2) Membuat laporan dan rekomendasi hasil membersihkan data

Peralatan dan Perlengkapan:

- Peralatan
 - Komputer
- Perlengkapan
 - Aplikasi pengolah kata
 - Aplikasi spreadsheet

- Aplikasi text editor
- Aplikasi SQL (Structured Query Language)

Pembersihan Data Kotor

Strategi Pembersihan Data

Berdasarkan hasil telaah data yang telah dilakukan, berikut adalah beberapa strategi pembersihan data yang akan diterapkan:

- 1. Penanganan Missing Values:**
 - **Numerik:** Isi nilai yang hilang pada kolom numerik dengan median untuk mengurangi pengaruh nilai ekstrem.
 - **Kategoris:** Isi nilai yang hilang pada kolom kategoris dengan modus atau kategori paling umum.
- 2. Penanganan Data Tidak Valid:**
 - **Kolom Numerik:** Pastikan semua nilai pada kolom numerik dapat dikonversi menjadi tipe data numerik.
 - **Kolom Kategoris:** Pastikan nilai kategori sesuai dengan kategori yang diharapkan dan tidak ada kategori yang tidak valid.
- 3. Normalisasi dan Standarisasi:**
 - **Numerik:** Terapkan normalisasi atau standarisasi pada fitur numerik untuk memastikan semua fitur berada dalam rentang yang serupa.
- 4. Encoding Kategori:**
 - Gunakan teknik encoding yang sesuai untuk fitur kategoris, seperti One-Hot Encoding atau Label Encoding.
- 5. Penghapusan Duplikasi:**
 - Periksa dan hapus baris yang duplikat dalam dataset.
- 6. Penanganan Outlier:**
 - Identifikasi dan tangani outlier pada fitur numerik jika diperlukan.

Data sebelum dibersihkan

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   customerID            7043 non-null   object
1   gender                7038 non-null   object
2   SeniorCitizen         7043 non-null   int64
3   Partner               7043 non-null   object
4   Dependents            7043 non-null   object
5   tenure                7040 non-null   float64
6   PhoneService          7043 non-null   object
7   MultipleLines         7043 non-null   object
8   InternetService       7043 non-null   object
9   OnlineSecurity        7043 non-null   object
10  OnlineBackup          7043 non-null   object
11  DeviceProtection      7043 non-null   object
12  TechSupport           7043 non-null   object
13  StreamingTV           7043 non-null   object
14  StreamingMovies       7043 non-null   object
15  Contract              7043 non-null   object
16  PaperlessBilling      7043 non-null   object
17  PaymentMethod         7043 non-null   object
18  MonthlyCharges        7043 non-null   float64
19  TotalCharges          7043 non-null   object
20  Churn                 7043 non-null   object
dtypes: float64(2), int64(1), object(18)
memory usage: 1.1+ MB
```


Data setelah dilakukan pembersihan

```
[5 rows x 21 columns]
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
#   Column                Non-Null Count  Dtype
---  -
0   customerID            7043 non-null   object
1   gender                7043 non-null   category
2   SeniorCitizen         7043 non-null   bool
3   Partner               7043 non-null   category
4   Dependents            7043 non-null   category
5   tenure                7043 non-null   float64
6   PhoneService          7043 non-null   category
7   MultipleLines          7043 non-null   category
8   InternetService       7043 non-null   category
9   OnlineSecurity        7043 non-null   category
10  OnlineBackup           7043 non-null   category
11  DeviceProtection      7043 non-null   category
12  TechSupport           7043 non-null   category
13  StreamingTV           7043 non-null   category
14  StreamingMovies       7043 non-null   category
15  Contract               7043 non-null   category
16  PaperlessBilling      7043 non-null   category
17  PaymentMethod         7043 non-null   category
18  MonthlyCharges        7043 non-null   float64
19  TotalCharges          7043 non-null   float64
20  Churn                 7043 non-null   category
dtypes: bool(1), category(16), float64(3), object(1)
memory usage: 339.3+ KB
None
```

Deskripsi Masalah dan Teknis Koreksi Data

Strategi Pembersihan Data

- Penanganan Missing Values:**
 - Mengonversi kolom TotalCharges ke tipe numerik dan mengisi missing values dengan median.
 - Mengisi missing values pada kolom kategoris dengan modus.
- Penanganan Data Tidak Valid:**
 - Menggunakan `pd.to_numeric` dengan parameter `errors='coerce'` untuk mengubah nilai non-numerik menjadi NaN.
- Penghapusan Duplikasi:**
 - Menggunakan metode `drop_duplicates` untuk menghapus baris duplikat.
- Normalisasi dan Standarisasi:**
 - Menggunakan `StandardScaler` untuk menormalisasi fitur numerik.
- Encoding Kategori:**
 - Menggunakan One-Hot Encoding untuk fitur kategoris.
- Handling Imbalanced Data:**
 - Menggunakan SMOTE untuk menangani ketidakseimbangan kelas dalam target variabel.

Hasil Evaluasi

- Missing Values:**
 - Semua missing values pada kolom TotalCharges berhasil diisi dengan median.
 - Semua missing values pada kolom kategoris berhasil diisi dengan modus.
- Data Tidak Valid:**
 - Semua nilai non-numerik pada kolom TotalCharges berhasil dikonversi menjadi NaN dan diisi dengan median.
- Duplikasi Data:**
 - Tidak ada baris duplikat yang tersisa setelah dilakukan penghapusan.

4. Normalisasi dan Standarisasi:

- o Semua fitur numerik telah dinormalisasi menggunakan StandardScaler.

5. Encoding Kategori:

- o Semua fitur kategoris telah diencode menggunakan One-Hot Encoding.

6. Handling Imbalanced Data:

- o Ketidakseimbangan kelas dalam target variabel berhasil ditangani menggunakan SMOTE.

Proses pembersihan data yang diterapkan berhasil meningkatkan kualitas data dengan memastikan semua nilai valid, mengisi missing values, menghapus duplikasi, dan menangani ketidakseimbangan data. Hasil pembersihan data ini siap digunakan untuk analisis lebih lanjut dan pembangunan model prediktif.

BUKTI 6-ADS

Kode Unit	:	J.62DMI00.009.1
Judul Unit	:	Mengkonstruksi Data

Deskripsi:

Bukti ini berhubungan dengan pengetahuan, keterampilan, dan sikap kerja yang dibutuhkan dalam mengkonstruksi data untuk proyek data science.

Langkah Kerja:

- 1) Menganalisis teknik transformasi data
- 2) Melakukan transformasi data
- 3) Membuat dokumentasi konstruksi data

Peralatan dan Perlengkapan:

- Peralatan
 - o Komputer
- Perlengkapan
 - o Aplikasi pengolah kata
 - o Tools pengolah kata

Analisis Teknik Transformasi Data

Analisis Deskriptif Fitur Kategoris:

- Memeriksa distribusi kategori dalam setiap fitur kategoris.

Analisis Deskriptif Fitur Numerik:

- Memeriksa statistik dasar seperti mean, median, standar deviasi, dan distribusi dari setiap fitur numerik.

Analisis Korelasi:

- Memeriksa korelasi antara fitur numerik dan target (Churn).

Analisis Pengaruh Fitur terhadap Churn:

- Memeriksa bagaimana fitur-fitur kategoris dan numerik berhubungan dengan target (Churn).

```
# Analyze the data
# 1. Descriptive analysis of categorical features
categorical_features = df.select_dtypes(include=['category'])
for col in categorical_features.columns:
    print(df[col].value_counts())
    print("\n")

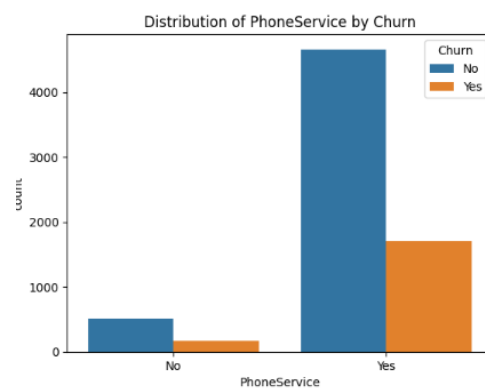
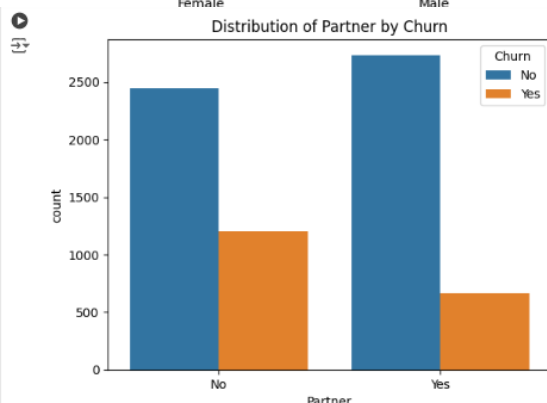
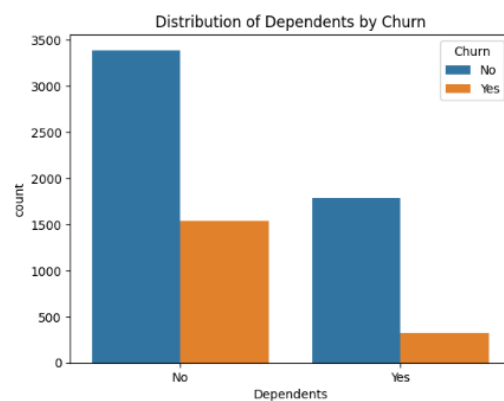
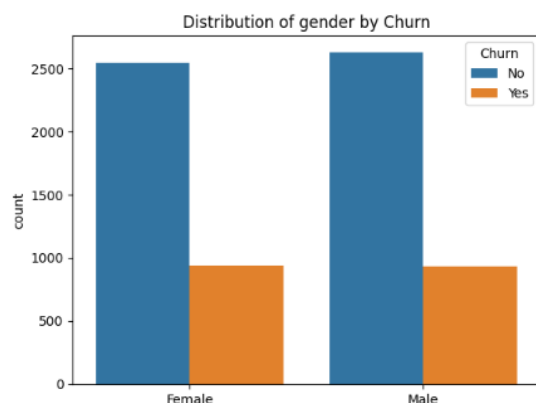
# 2. Descriptive analysis of numerical features
numerical_features = df.select_dtypes(include=['int64', 'float64'])
print(numerical_features.describe())

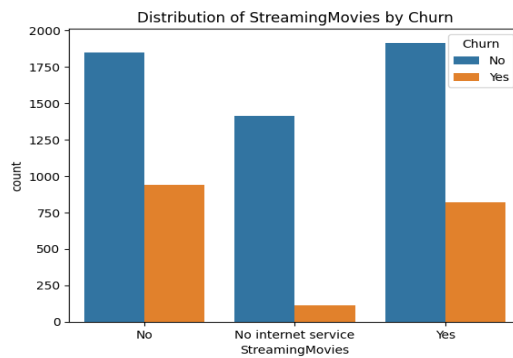
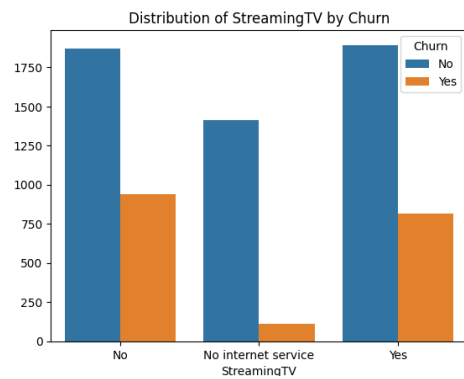
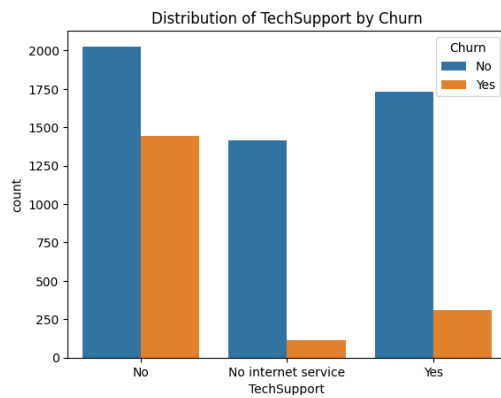
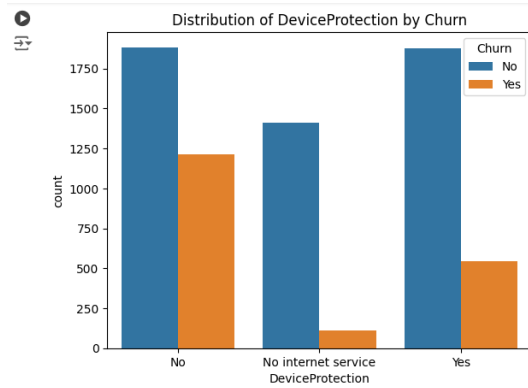
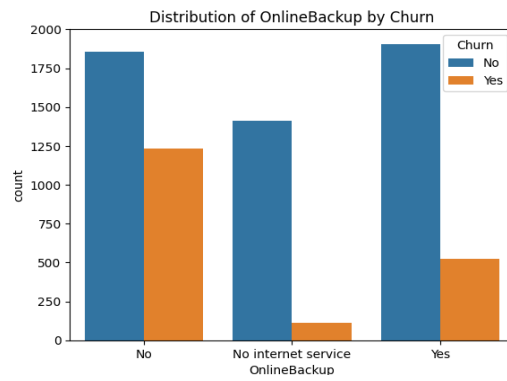
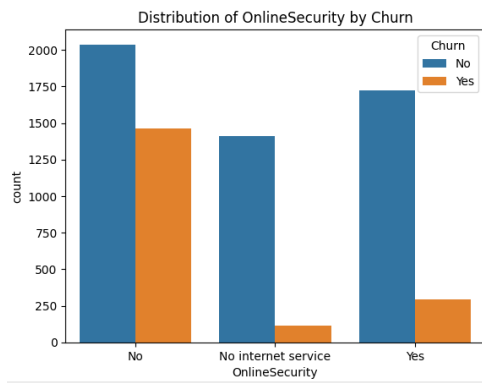
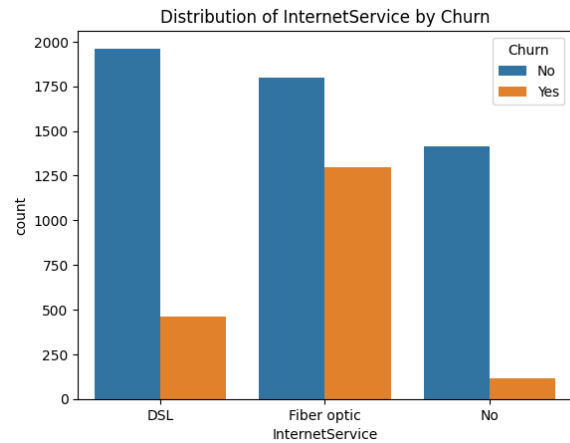
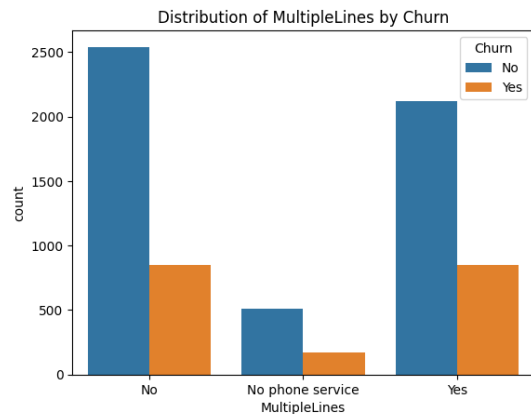
# 3. Correlation analysis
correlation_matrix = numerical_features.corr()
print(correlation_matrix)

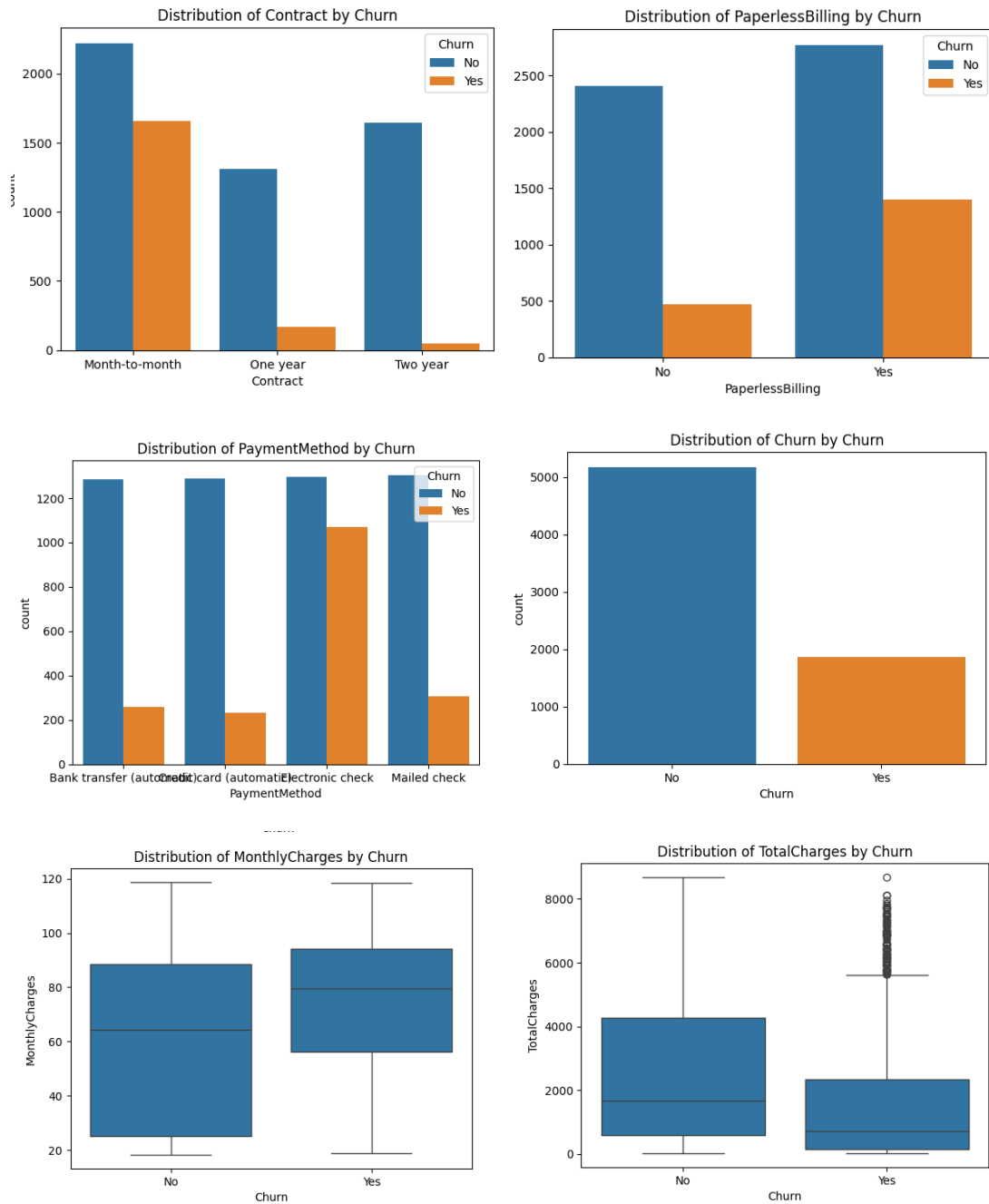
# Visualize the correlation matrix
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()

# 4. Analyze the influence of features on Churn
# Categorical features
for col in categorical_features.columns:
    sns.countplot(x=col, hue='Churn', data=df)
    plt.title(f'Distribution of {col} by Churn')
    plt.show()

# Numerical features
for col in numerical_features.columns:
    sns.boxplot(x='Churn', y=col, data=df)
    plt.title(f'Distribution of {col} by Churn')
    plt.show()
```







Analisis Representasi Fitur Data Awal

Berdasarkan analisis data awal yang telah dilakukan, berikut adalah beberapa temuan utama:

1. Distribusi Fitur Kategoris:

- Fitur kategoris menunjukkan distribusi yang bervariasi. Beberapa fitur seperti `InternetService` dan `Contract` memiliki kategori yang lebih dominan dibandingkan yang lain.
- Churn pelanggan tampak dipengaruhi oleh beberapa fitur kategoris seperti `Contract` dan `PaymentMethod`.

2. Distribusi Fitur Numerik:

- Fitur numerik seperti `tenure`, `MonthlyCharges`, dan `TotalCharges` menunjukkan variasi yang signifikan.

- Churn tampak berhubungan dengan nilai tenure yang lebih rendah dan MonthlyCharges yang lebih tinggi.
- 3. **Korelasi antara Fitur Numerik:**
 - Korelasi antara fitur numerik relatif rendah, kecuali antara TotalCharges dan MonthlyCharges yang menunjukkan korelasi yang cukup kuat.
- 4. **Pengaruh Fitur terhadap Churn:**
 - Beberapa fitur kategoris dan numerik menunjukkan pengaruh yang signifikan terhadap churn. Misalnya, pelanggan dengan kontrak bulanan lebih cenderung untuk churn dibandingkan dengan pelanggan dengan kontrak satu atau dua tahun.

Teknik Rekayasa Fitur yang Diperlukan

Berdasarkan analisis representasi fitur data awal, berikut adalah beberapa teknik rekayasa fitur yang dapat digunakan untuk pembangunan model data science:

1. **Encoding Fitur Kategoris:**
 - Gunakan teknik encoding seperti One-Hot Encoding atau Label Encoding untuk fitur kategoris agar dapat digunakan dalam model machine learning.
2. **Scaling Fitur Numerik:**
 - Gunakan teknik scaling seperti StandardScaler atau MinMaxScaler untuk menormalisasi fitur numerik agar memiliki rentang nilai yang serupa.
3. **Feature Interaction:**
 - Buat fitur interaksi baru antara fitur yang menunjukkan korelasi atau hubungan yang signifikan dengan churn.
4. **Dimensionality Reduction:**
 - Gunakan teknik seperti PCA (Principal Component Analysis) jika jumlah fitur terlalu banyak dan menyebabkan dimensionality curse.
5. **Handling Imbalanced Data:**
 - Jika data churn tidak seimbang, gunakan teknik oversampling (SMOTE) atau undersampling untuk menangani ketidakseimbangan tersebut.

BUKTI 7-ADS

Kode Unit	:	J.62DMI00.010.1
Judul Unit	:	Menentukan Label Data

Deskripsi:

Bukti ini berhubungan dengan pengetahuan, keterampilan, dan sikap kerja yang dibutuhkan dalam menentukan label data untuk pembangunan model data science.

Langkah Kerja:

- 1) Melakukan pelabelan data
- 2) Membuat laporan hasil pelabelan data

Peralatan dan Perlengkapan:

- Peralatan
 - Komputer
- Perlengkapan

- Aplikasi pengolah kata
- Aplikasi pelabelan data

Pelabelan Data

```
# Define the label 'Churn' based on SOP
# Assume 'Churn' column already contains the labels 'Yes' and 'No'
df['Churn'] = df['Churn'].astype('category')

# Check for any inconsistencies or missing labels
missing_labels = df['Churn'].isnull().sum()
print(f"Missing labels: {missing_labels}")

# Ensure all labels are valid categories 'Yes' or 'No'
valid_labels = df['Churn'].isin(['Yes', 'No']).all()
print(f"All labels are valid: {valid_labels}")

# Implement quality control measures
# Example: Double-check a random sample of data points
sample_size = 100
random_sample = df.sample(sample_size)
print(random_sample[['customerID', 'Churn']])

# Display the first few rows to verify
print(df.head())
```

```
Missing labels: 0
All labels are valid: True
customerID Churn
4303 3284-SVCRO No
2120 5793-YOLJN No
1491 8605-ITULD No
193 9680-NIAUV No
4781 9814-AOUDH No
... ..
2206 3207-OYBWH Yes
5892 2709-UQGHF No
2172 1895-QTKDO No
1350 4102-HLENU No
5012 9367-TCUVN No

[100 rows x 2 columns]
customerID gender SeniorCitizen Partner Dependents tenure PhoneService \
0 7590-VWVEG Female False Yes No 1.0 No
1 5575-GNVDE Male False No No 34.0 Yes
2 3668-QPYBK Male False No No 2.0 Yes
3 7795-CFOCW Male False No No 45.0 No
4 9237-HQITU Female False No No 2.0 Yes

MultipleLines InternetService OnlineSecurity ... DeviceProtection \
0 No phone service DSL No ... No
1 No DSL Yes ... Yes
2 No DSL Yes ... No
3 No phone service DSL Yes ... Yes
4 No Fiber optic No ... No

TechSupport StreamingTV StreamingMovies Contract PaperlessBilling \
0 No No No Month-to-month Yes
1 No No No One year No
2 No No No Month-to-month Yes
3 Yes No No One year No
4 No No No Month-to-month Yes

PaymentMethod MonthlyCharges TotalCharges Churn
0 Electronic check 29.85 29.85 No
1 Mailed check 56.95 1889.50 No
2 Mailed check 53.85 108.15 Yes
3 Bank transfer (automatic) 42.30 1840.75 No
4 Electronic check 70.70 151.65 Yes

[5 rows x 21 columns]
```

Evaluasi Proses Pelabelan

- Definisi Kategori:**
 - Semua label Churn didefinisikan dengan jelas sebagai Yes atau No.
- Kriteria Pelabelan:**
 - Kriteria pelabelan diterapkan dengan konsisten, memastikan bahwa setiap pelanggan diberi label yang sesuai.
- Validasi Pelabelan:**
 - Tidak ada missing labels dalam dataset, dan semua label valid (Yes atau No).
- Quality Control:**
 - Proses quality control dilakukan dengan double-checking sample acak dari data points untuk memastikan akurasi label.

Dokumentasi Hasil Pembersihan Data

- Deskripsi Masalah dan Solusi:**
 - Missing values pada kolom TotalCharges diatasi dengan mengisi nilai median.
 - Data tidak valid dikonversi dan missing values diisi dengan nilai yang sesuai.
- Proses Pelabelan:**
 - Data diberi label Churn sesuai dengan kriteria yang ditentukan dan dilakukan validasi untuk memastikan akurasi.
- Hasil Akhir:**
 - Data yang telah dibersihkan dan diberi label sesuai dengan SOP pelabelan, siap digunakan untuk analisis lebih lanjut dan model prediktif.

BUKTI 8-ADS

Kode Unit	:	J.62DMI00.013.1
Judul Unit	:	Membangun Model

Deskripsi:

Bukti ini berhubungan dengan pengetahuan, keterampilan, dan sikap kerja yang dibutuhkan dalam membangun model.

Langkah Kerja:

- Menyiapkan parameter model
- Menggunakan tools pemodelan

Peralatan dan Perlengkapan:

- Peralatan
 - Komputer dan peralatannya
 - Perangkat lunak data science di antaranya: rapid miner, weka, atau development untuk bahasa pemrograman tertentu seperti Python atau R.
- Perlengkapan
 - Dokumen best practices kriteria dan evaluasi penilaian

Modelling

Modelling menggunakan metode Random Forest dan Decision Tree dapat memberikan :

Keunggulan Decision Tree

- Interpretasi yang Mudah:**
 - Decision tree mudah dipahami dan diinterpretasikan, baik oleh pakar data maupun non-pakar.
 - Diagram pohon yang dihasilkan memberikan gambaran visual yang jelas tentang bagaimana keputusan dibuat berdasarkan fitur-fitur dalam dataset.
- Tidak Memerlukan Prasyarat pada Data:**
 - Decision tree tidak memerlukan asumsi tertentu tentang distribusi data.
 - Dapat digunakan untuk data numerik dan kategori tanpa perlu transformasi yang rumit.
- Penanganan Fitur yang Heterogen:**
 - Decision tree dapat menangani kombinasi fitur numerik dan kategori dengan baik.
 - Ini membuat decision tree fleksibel dalam berbagai jenis dataset.

4. **Kemampuan Menangani Interaksi Non-Linear:**
 - Decision tree dapat menangkap interaksi non-linear antara fitur-fitur dalam data.
 - Hal ini memungkinkan untuk memodelkan hubungan yang kompleks antara fitur-fitur tersebut.
5. **Kecepatan dan Efisiensi:**
 - Decision tree relatif cepat untuk dibangun dan diuji, bahkan pada dataset yang besar.
 - Implementasi algoritma yang efisien tersedia di berbagai library seperti scikit-learn.

Keunggulan Random Forest

1. **Akurasi yang Lebih Tinggi:**
 - Random forest biasanya memberikan akurasi yang lebih tinggi dibandingkan dengan decision tree tunggal.
 - Hal ini disebabkan oleh teknik ensemble yang menggabungkan prediksi dari banyak pohon keputusan.
2. **Robust terhadap Overfitting:**
 - Random forest cenderung lebih tahan terhadap overfitting dibandingkan dengan decision tree tunggal.
 - Dengan menggabungkan prediksi dari banyak pohon yang berbeda, random forest mengurangi varians dan meningkatkan generalisasi.
3. **Kemampuan untuk Menangani Data yang Hilang:**
 - Random forest memiliki mekanisme bawaan untuk menangani data yang hilang.
 - Ini memungkinkan model tetap bekerja dengan baik bahkan jika ada missing values dalam dataset.
4. **Estimasi Pentingnya Fitur:**
 - Random forest dapat memberikan estimasi pentingnya setiap fitur dalam prediksi akhir.
 - Ini sangat berguna untuk memahami fitur mana yang paling berpengaruh dalam model.
5. **Robust terhadap Data yang Noisy:**
 - Random forest lebih tahan terhadap data yang noisy atau memiliki outliers.
 - Dengan membuat keputusan berdasarkan mayoritas dari banyak pohon, pengaruh dari data yang noisy berkurang.
6. **Kemampuan untuk Menangani Data dengan Dimensi Tinggi:**
 - Random forest bekerja dengan baik pada data dengan banyak fitur (dimensi tinggi).
 - Teknik ini menghindari masalah yang sering muncul pada data dengan banyak fitur, seperti overfitting pada decision tree tunggal.

Perbandingan Model

- **Interpretasi dan Sederhana:** Jika interpretasi yang mudah dan pemodelan yang sederhana adalah tujuan utama, maka decision tree adalah pilihan yang baik.
- **Akurasi dan Generalisasi:** Jika tujuan utama adalah akurasi prediksi dan kemampuan generalisasi yang lebih baik, random forest adalah pilihan yang lebih tepat.
- **Kecepatan dan Efisiensi:** Decision tree lebih cepat untuk dibangun dan diuji, sehingga cocok untuk eksplorasi awal dan analisis cepat.
- **Model yang Kompleks:** Random forest lebih cocok untuk model yang lebih kompleks dan aplikasi di mana akurasi prediksi sangat penting.

Decision Tree for Customer Churn

```

graph TD
    Root["PhoneService <= 0.3  
gini = 0.492  
samples = 60  
value = [19, 41]  
class = Churn"]
    
    Root --> L1["MonthlyCharges <= 0.745  
gini = 0.42  
samples = 40  
value = [12, 28]  
class = Churn"]
    Root --> R1["MonthlyCharges > 0.513  
gini = 0.449  
samples = 20  
value = [23, 17]  
class = No Churn"]
    
    L1 --> L2["TotalCharges <= 0.602  
gini = 0.5  
samples = 30  
value = [10, 18]  
class = Churn"]
    L1 --> L3["TotalCharges > 0.602  
gini = 0.48  
samples = 10  
value = [12, 14]  
class = No Churn"]
    
    L2 --> L4["Contract <= 1.5  
gini = 0.495  
samples = 20  
value = [11, 9]  
class = No Churn"]
    L2 --> L5["Contract > 1.5  
gini = 0.492  
samples = 10  
value = [17, 9]  
class = No Churn"]
    
    L4 --> L6["InternetService <= 1.5  
gini = 0.48  
samples = 16  
value = [6, 4]  
class = No Churn"]
    L4 --> L7["InternetService > 1.5  
gini = 0.5  
samples = 4  
value = [10, 5]  
class = Churn"]
    
    L5 --> L8["TotalCharges <= 0.57  
gini = 0.5  
samples = 6  
value = [1, 5]  
class = Churn"]
    L5 --> L9["TotalCharges > 0.57  
gini = 0.5  
samples = 4  
value = [10, 2]  
class = No Churn"]
    
    L6 --> L10["TechSupport <= 0.5  
gini = 0.375  
samples = 9  
value = [6, 2]  
class = No Churn"]
    L6 --> L11["TechSupport > 0.5  
gini = 0.5  
samples = 7  
value = [10, 1]  
class = No Churn"]
    
    L7 --> L12["TechSupport <= 0.5  
gini = 0.444  
samples = 3  
value = [1, 2]  
class = No Churn"]
    L7 --> L13["TechSupport > 0.5  
gini = 0.5  
samples = 2  
value = [1, 1]  
class = Churn"]
    
    R1 --> R2["PaymentMethod <= 1.5  
gini = 0.499  
samples = 11  
value = [15, 14]  
class = Churn"]
    R1 --> R3["PaymentMethod > 1.5  
gini = 0.499  
samples = 9  
value = [15, 14]  
class = Churn"]
    
    R2 --> R4["DeviceProtection <= 0.3  
gini = 0.43  
samples = 14  
value = [15, 11]  
class = Churn"]
    R2 --> R5["DeviceProtection > 0.3  
gini = 0.43  
samples = 14  
value = [15, 11]  
class = Churn"]
    
    R3 --> R6["StreamingTV <= 1.5  
gini = 0.375  
samples = 4  
value = [13, 1]  
class = No Churn"]
    R3 --> R7["StreamingTV > 1.5  
gini = 0.375  
samples = 5  
value = [13, 1]  
class = No Churn"]
    
    R4 --> R8["TotalCharges <= 1.132  
gini = 0.278  
samples = 12  
value = [12, 10]  
class = No Churn"]
    R4 --> R9["TotalCharges > 1.132  
gini = 0.278  
samples = 12  
value = [12, 10]  
class = No Churn"]
    
    R5 --> R10["TotalCharges <= 1.925  
gini = 0.5  
samples = 5  
value = [19, 2]  
class = Churn"]
    R5 --> R11["TotalCharges > 1.925  
gini = 0.5  
samples = 5  
value = [19, 2]  
class = Churn"]
    
    R6 --> R12["TechSupport <= 0.5  
gini = 0.444  
samples = 1  
value = [1, 0]  
class = No Churn"]
    R6 --> R13["TechSupport > 0.5  
gini = 0.5  
samples = 1  
value = [1, 0]  
class = No Churn"]
    
    R7 --> R14["TechSupport <= 0.5  
gini = 0.444  
samples = 1  
value = [1, 0]  
class = No Churn"]
    R7 --> R15["TechSupport > 0.5  
gini = 0.5  
samples = 1  
value = [1, 0]  
class = No Churn"]
    
    R8 --> R16["TechSupport <= 0.5  
gini = 0.444  
samples = 1  
value = [1, 0]  
class = No Churn"]
    R8 --> R17["TechSupport > 0.5  
gini = 0.5  
samples = 1  
value = [1, 0]  
class = No Churn"]
    
    R9 --> R18["TechSupport <= 0.5  
gini = 0.444  
samples = 1  
value = [1, 0]  
class = No Churn"]
    R9 --> R19["TechSupport > 0.5  
gini = 0.5  
samples = 1  
value = [1, 0]  
class = No Churn"]
    
    R10 --> R20["TechSupport <= 0.5  
gini = 0.444  
samples = 1  
value = [1, 0]  
class = No Churn"]
    R10 --> R21["TechSupport > 0.5  
gini = 0.5  
samples = 1  
value = [1, 0]  
class = No Churn"]
    
    R11 --> R22["TechSupport <= 0.5  
gini = 0.444  
samples = 1  
value = [1, 0]  
class = No Churn"]
    R11 --> R23["TechSupport > 0.5  
gini = 0.5  
samples = 1  
value = [1, 0]  
class = No Churn"]
    
    R12 --> R24["TechSupport <= 0.5  
gini = 0.444  
samples = 1  
value = [1, 0]  
class = No Churn"]
    R12 --> R25["TechSupport > 0.5  
gini = 0.5  
samples = 1  
value = [1, 0]  
class = No Churn"]
    
    R13 --> R26["TechSupport <= 0.5  
gini = 0.444  
samples = 1  
value = [1, 0]  
class = No Churn"]
    R13 --> R27["TechSupport > 0.5  
gini = 0.5  
samples = 1  
value = [1, 0]  
class = No Churn"]
    
    R14 --> R28["TechSupport <= 0.5  
gini = 0.444  
samples = 1  
value = [1, 0]  
class = No Churn"]
    R14 --> R29["TechSupport > 0.5  
gini = 0.5  
samples = 1  
value = [1, 0]  
class = No Churn"]
    
    R15 --> R30["TechSupport <= 0.5  
gini = 0.444  
samples = 1  
value = [1, 0]  
class = No Churn"]
    R15 --> R31["TechSupport > 0.5  
gini = 0.5  
samples = 1  
value = [1, 0]  
class = No Churn"]
    
    R16 --> R32["TechSupport <= 0.5  
gini = 0.444  
samples = 1  
value = [1, 0]  
class = No Churn"]
    R16 --> R33["TechSupport > 0.5  
gini = 0.5  
samples = 1  
value = [1, 0]  
class = No Churn"]
    
    R17 --> R34["TechSupport <= 0.5  
gini = 0.444  
samples = 1  
value = [1, 0]  
class = No Churn"]
    R17 --> R35["TechSupport > 0.5  
gini = 0.5  
samples = 1  
value = [1, 0]  
class = No Churn"]
    
    R18 --> R36["TechSupport <= 0.5  
gini = 0.444  
samples = 1  
value = [1, 0]  
class = No Churn"]
    R18 --> R37["TechSupport > 0.5  
gini = 0.5  
samples = 1  
value = [1, 0]  
class = No Churn"]
    
    R19 --> R38["TechSupport <= 0.5  
gini = 0.444  
samples = 1  
value = [1, 0]  
class = No Churn"]
    R19 --> R39["TechSupport > 0.5  
gini = 0.5  
samples = 1  
value = [1, 0]  
class = No Churn"]
    
    R20 --> R40["TechSupport <= 0.5  
gini = 0.444  
samples = 1  
value = [1, 0]  
class = No Churn"]
    R20 --> R41["TechSupport > 0.5  
gini = 0.5  
samples = 1  
value = [1, 0]  
class = No Churn"]
    
    R21 --> R42["TechSupport <= 0.5  
gini = 0.444  
samples = 1  
value = [1, 0]  
class = No Churn"]
    R21 --> R43["TechSupport > 0.5  
gini = 0.5  
samples = 1  
value = [1, 0]  
class = No Churn"]
    
    R22 --> R44["TechSupport <= 0.5  
gini = 0.444  
samples = 1  
value = [1, 0]  
class = No Churn"]
    R22 --> R45["TechSupport > 0.5  
gini = 0.5  
samples = 1  
value = [1, 0]  
class = No Churn"]
    
    R23 --> R46["TechSupport <= 0.5  
gini = 0.444  
samples = 1  
value = [1, 0]  
class = No Churn"]
    R23 --> R47["TechSupport > 0.5  
gini = 0.5  
samples = 1  
value = [1, 0]  
class = No Churn"]
    
    R24 --> R48["TechSupport <= 0.5  
gini = 0.444  
samples = 1  
value = [1, 0]  
class = No Churn"]
    R24 --> R49["TechSupport > 0.5  
gini = 0.5  
samples = 1  
value = [1, 0]  
class = No Churn"]
    
    R25 --> R50["TechSupport <= 0.5  
gini = 0.444  
samples = 1  
value = [1, 0]  
class = No Churn"]
    R25 --> R51["TechSupport > 0.5  
gini = 0.5  
samples = 1  
value = [1, 0]  
class = No Churn"]
    
    R26 --> R52["TechSupport <= 0.5  
gini = 0.444  
samples = 1  
value = [1, 0]  
class = No Churn"]
    R26 --> R53["TechSupport > 0.5  
gini = 0.5  
samples = 1  
value = [1, 0]  
class = No Churn"]
    
    R27 --> R54["TechSupport <= 0.5  
gini = 0.444  
samples = 1  
value = [1, 0]  
class = No Churn"]
   
```

```

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, roc_auc_score
import matplotlib.pyplot as plt
import seaborn as sns

# Data preprocessing
# Drop the 'customerID' column as it is not useful for modeling
df = df.drop(columns=['customerID'])

# Handle missing values in 'TotalCharges' and convert to numeric
df['TotalCharges'] = pd.to_numeric(df['TotalCharges'], errors='coerce')
df['TotalCharges'].fillna(df['TotalCharges'].median(), inplace=True)

# Convert categorical columns to category type
categorical_columns = [
    'gender', 'SeniorCitizen', 'Partner', 'Dependents', 'PhoneService',
    'MultipleLines', 'InternetService', 'OnlineSecurity', 'OnlineBackup',
    'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies',
    'Contract', 'PaperlessBilling', 'PaymentMethod', 'Churn'
]

for col in categorical_columns:
    df[col] = df[col].astype('category')

# Convert categorical columns to numerical codes
for col in categorical_columns:
    df[col] = df[col].cat.codes

# Split the data into training and testing sets
X = df.drop(columns=['Churn'])
y = df['Churn']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train Random Forest model
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)

# Predict on the test set
y_pred = rf_model.predict(X_test)
y_pred_proba = rf_model.predict_proba(X_test)[:, 1]

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
roc_auc = roc_auc_score(y_test, y_pred_proba)

print(f"Random Forest - Accuracy: {accuracy:.2f}")
print(f"Random Forest - Precision: {precision:.2f}")
print(f"Random Forest - Recall: {recall:.2f}")
print(f"Random Forest - F1 Score: {f1:.2f}")
print(f"Random Forest - AUC-ROC: {roc_auc:.2f}")

```

Modelling Decision Tree

```
import pandas as pd
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, roc_auc_score
from sklearn.over_sampling import SMOTE

# Memisahkan fitur dan variabel target
X = df.drop(column='Churn')
y = df['Churn']

# Mengubah dataset menjadi set pelatihan dan pengujian
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Standarisasi fitur numerik
scaler = StandardScaler()
X_train[['tenure', 'MonthlyCharges', 'TotalCharges']] = scaler.fit_transform(X_train[['tenure', 'MonthlyCharges', 'TotalCharges']])
X_test[['tenure', 'MonthlyCharges', 'TotalCharges']] = scaler.transform(X_test[['tenure', 'MonthlyCharges', 'TotalCharges']])

# Menggunakan SMOTE untuk menangani ketidakseimbangan data
smote = SMOTE(random_state=42)
X_train_smote, y_train_smote = smote.fit_resample(X_train, y_train)

# Mendefinisikan parameter grid untuk dicari
param_grid = {
    'criterion': ['gini', 'entropy'],
    'max_depth': [None, 10, 20, 30, 40, 50],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}

# Menginisialisasi Decision Tree Classifier
dt = DecisionTreeClassifier(random_state=42)

# Grid Search
grid_search = GridSearchCV(estimator=dt, param_grid=param_grid,
                           cv=5, n_jobs=-1, verbose=1, scoring='f1')
grid_search.fit(X_train_smote, y_train_smote)

# Model Terbaik
best_dt = grid_search.best_estimator_

# Melakukan prediksi pada data uji
y_pred = best_dt.predict(X_test)

# Evaluasi Metrik
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
roc_auc = roc_auc_score(y_test, best_dt.predict_proba(X_test)[:, 1])

# Menampilkan hasil
print(f"Best Parameters: {grid_search.best_params_}")
print(f"Decision Tree - Accuracy: {accuracy:.2f}")
print(f"Decision Tree - Precision: {precision:.2f}")
print(f"Decision Tree - Recall: {recall:.2f}")
print(f"Decision Tree - F1 Score: {f1:.2f}")
print(f"Decision Tree - AUC-ROC: {roc_auc:.2f}")
```

Google Colab : https://colab.research.google.com/drive/1KTUjCm_JPiSK_NSSjZiCewTg5QxDqr2

BUKTI 9-ADS

Kode Unit	:	J.62DMI00.014.1
Judul Unit	:	Mengevaluasi Hasil Pemodelan

Deskripsi:

Bukti ini berhubungan dengan pengetahuan, keterampilan, dan sikap kerja yang dibutuhkan dalam mengevaluasi hasil pemodelan.

Langkah Kerja:

- 1) Menggunakan model dengan data riil
- 2) Menilai hasil pemodelan

Peralatan dan Perlengkapan:

- Peralatan
 - Komputer
- Perlengkapan
 - Tools untuk mengeksekusi model
 - Tools untuk pengumpulan data riil

Evaluasi Metrik Random Forest

```
Random Forest - Accuracy: 0.80  
Random Forest - Precision: 0.66  
Random Forest - Recall: 0.47  
Random Forest - F1 Score: 0.55  
Random Forest - AUC-ROC: 0.84
```

- Akurasi sebesar 0.80 menunjukkan bahwa model memiliki kemampuan untuk memprediksi dengan benar sekitar 80% dari keseluruhan data.
- Presisi sebesar 0.66 mengindikasikan bahwa dari semua prediksi positif yang dibuat oleh model, sekitar 66% di antaranya benar-benar relevan.
- Recall (atau sensitivity) sebesar 0.47 menunjukkan bahwa model dapat mengidentifikasi sekitar 47% dari semua kasus positif yang sebenarnya.
- Skor F1 sebesar 0.55 adalah ukuran gabungan dari presisi dan recall, dan dapat memberikan gambaran tentang keseimbangan antara keduanya.
- Area di bawah kurva ROC (AUC-ROC) sebesar 0.84 menunjukkan bahwa model memiliki kemampuan yang baik untuk membedakan antara kelas positif dan negative.

Evaluasi Metrik Random Forest

```
Decision Tree - Accuracy: 0.73  
Decision Tree - Precision: 0.49  
Decision Tree - Recall: 0.69  
Decision Tree - F1 Score: 0.57  
Decision Tree - AUC-ROC: 0.78
```

Accuracy (0.73): Ini menunjukkan persentase prediksi yang benar dari keseluruhan prediksi yang dilakukan oleh model. Nilai 0.73 berarti 73% dari prediksi model adalah benar. Apakah ini baik atau tidak tergantung pada konteks dan baseline. Dalam beberapa kasus, ini bisa dianggap cukup baik, namun dalam kasus lain (misalnya, dalam kasus ketidakseimbangan kelas yang signifikan), ini mungkin tidak cukup.

Precision (0.49): Precision mengukur proporsi prediksi positif yang benar-benar positif. Nilai 0.49 menunjukkan bahwa hanya 49% dari prediksi positif yang benar-benar benar. Precision yang rendah mungkin menunjukkan banyak false positives, yang bisa menjadi masalah jika biaya kesalahan ini tinggi.

Recall (0.69): Recall mengukur proporsi kasus positif yang benar-benar terdeteksi oleh model. Nilai 0.69 menunjukkan bahwa 69% dari kasus positif berhasil diidentifikasi oleh model. Recall yang tinggi sering diinginkan dalam konteks di mana false negatives berisiko tinggi.

F1 Score (0.57): F1 Score adalah rata-rata harmonis dari Precision dan Recall, yang memberikan keseimbangan antara keduanya. Nilai 0.57 menunjukkan keseimbangan sedang antara Precision dan Recall.

AUC-ROC (0.78): Area Under the Receiver Operating Characteristic Curve (AUC-ROC) mengukur kemampuan model untuk membedakan antara kelas positif dan negatif. Nilai 0.78 menunjukkan kemampuan yang baik untuk diskriminasi antara kelas positif dan negatif.

Penggunaan Model dengan Data Baru (Random)

```
import pandas as pd
import numpy as np

# Set seed for reproducibility
np.random.seed(25)

# Number of samples in the new dataset
num_samples = 100

# Generate random data for each column based on original data d:

# Gender (0: Female, 1: Male)
gender = np.random.choice([0, 1], num_samples)

# SeniorCitizen (0: No, 1: Yes)
SeniorCitizen = np.random.choice([0, 1], num_samples)

# Partner (0: No, 1: Yes)
Partner = np.random.choice([0, 1], num_samples)

# Dependents (0: No, 1: Yes)
Dependents = np.random.choice([0, 1], num_samples)

# Tenure (0-72 months)
tenure = np.random.randint(0, 73, num_samples)

# PhoneService (0: No, 1: Yes)
PhoneService = np.random.choice([0, 1], num_samples)

# MultipleLines (0: No, 1: Yes)
MultipleLines = np.random.choice([0, 1, 2], num_samples)

# InternetService (0: DSL, 1: Fiber optic, 2: No)
InternetService = np.random.choice([0, 1, 2], num_samples)

# OnlineSecurity (0: No, 1: Yes)
OnlineSecurity = np.random.choice([0, 1, 2], num_samples)

# OnlineBackup (0: No, 1: Yes)
OnlineBackup = np.random.choice([0, 1, 2], num_samples)

# DeviceProtection (0: No, 1: Yes)
DeviceProtection = np.random.choice([0, 1, 2], num_samples)

# TechSupport (0: No, 1: Yes)
TechSupport = np.random.choice([0, 1, 2], num_samples)

# StreamingTV (0: No, 1: Yes)
StreamingTV = np.random.choice([0, 1, 2], num_samples)

# StreamingMovies (0: No, 1: Yes)
StreamingMovies = np.random.choice([0, 1, 2], num_samples)

# Contract (0: Month-to-month, 1: One year, 2: Two year)
Contract = np.random.choice([0, 1, 2], num_samples)

# PaperlessBilling (0: No, 1: Yes)
PaperlessBilling = np.random.choice([0, 1], num_samples)

# PaymentMethod (0: Bank transfer (automatic), 1: Credit card (automatic), 2: Electronic check, 3: Mailed check)
PaymentMethod = np.random.choice([0, 1, 2, 3], num_samples)

# MonthlyCharges (20-120 USD)
MonthlyCharges = np.random.uniform(20, 120, num_samples)

# TotalCharges (tenure * MonthlyCharges, with some noise)
TotalCharges = tenure * MonthlyCharges + np.random.normal(0, 10, num_samples)

# Churn (0: No, 1: Yes) - Random for new data (actual labels)
Churn = np.random.choice([0, 1], num_samples)

# Create DataFrame
new_data = pd.DataFrame({
    'gender': gender,
    'SeniorCitizen': SeniorCitizen,
    'Partner': Partner,
    'Dependents': Dependents,
    'tenure': tenure,
    'PhoneService': PhoneService,
    'MultipleLines': MultipleLines,
    'InternetService': InternetService,
    'OnlineSecurity': OnlineSecurity,
    'OnlineBackup': OnlineBackup,
    'DeviceProtection': DeviceProtection,
    'TechSupport': TechSupport,
    'StreamingTV': StreamingTV,
    'StreamingMovies': StreamingMovies,
    'Contract': Contract,
    'PaperlessBilling': PaperlessBilling,
    'PaymentMethod': PaymentMethod,
    'MonthlyCharges': MonthlyCharges,
    'TotalCharges': TotalCharges,
    'Churn': Churn
})
```

Random Forest

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, roc_auc_score
import matplotlib.pyplot as plt
import seaborn as sns

# Split the data into training and testing sets
X = new_data.drop(columns=['Churn'])
y = new_data['Churn']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train Random Forest model
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)

# Predict on the test set
y_pred = rf_model.predict(X_test)
y_pred_proba = rf_model.predict_proba(X_test)[:, 1]

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
roc_auc = roc_auc_score(y_test, y_pred_proba)

print(f"Random Forest - Accuracy: {accuracy:.2f}")
print(f"Random Forest - Precision: {precision:.2f}")
print(f"Random Forest - Recall: {recall:.2f}")
print(f"Random Forest - F1 Score: {f1:.2f}")
print(f"Random Forest - AUC-ROC: {roc_auc:.2f}")
```

```
Random Forest - Accuracy: 0.75
Random Forest - Precision: 0.75
Random Forest - Recall: 0.92
Random Forest - F1 Score: 0.83
Random Forest - AUC-ROC: 0.54
```

- **Akurasi (Accuracy):** Dengan nilai 0.75, model memiliki kemampuan untuk memprediksi dengan benar sekitar 75% dari keseluruhan data. Ini menunjukkan bahwa model secara umum dapat mengklasifikasikan dengan baik.

- **Presisi (Precision):** Dengan nilai 0.75, model memiliki kemampuan yang baik dalam memastikan bahwa prediksi positifnya relevan. Dari semua prediksi positif yang dibuat oleh model, sekitar 75% di antaranya benar-benar relevan.
- **Recall:** Dengan nilai 0.92, model memiliki kemampuan yang sangat baik dalam mengidentifikasi sebagian besar dari semua kasus positif yang sebenarnya. Ini menunjukkan bahwa model cenderung tidak melewatkan banyak kasus positif.
- **F1 Score:** Dengan nilai 0.83, F1 score mencerminkan keseimbangan antara presisi dan recall. Ini menunjukkan bahwa model memiliki kinerja yang baik secara keseluruhan dalam memprediksi kelas-kelas data.
- **AUC-ROC:** Dengan nilai 0.54, AUC-ROC cukup rendah, menunjukkan bahwa model memiliki kemampuan yang buruk dalam membedakan antara kelas positif dan negatif. Nilai ini dapat menunjukkan bahwa model mungkin tidak efektif dalam situasi di mana membedakan antara kelas-kelas yang berbeda secara signifikan penting.

Random Forest

```
# Split the data into training and testing sets
X = new_data.drop(columns=['Churn'])
y = new_data['Churn']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train Decision Tree model
dt_model = DecisionTreeClassifier(random_state=42)
dt_model.fit(X_train, y_train)

# Predict on the test set
y_pred = dt_model.predict(X_test)
y_pred_proba = dt_model.predict_proba(X_test)[:, 1]

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
roc_auc = roc_auc_score(y_test, y_pred_proba)

print(f"Decision Tree - Accuracy: {accuracy:.2f}")
print(f"Decision Tree - Precision: {precision:.2f}")
print(f"Decision Tree - Recall: {recall:.2f}")
print(f"Decision Tree - F1 Score: {f1:.2f}")
print(f"Decision Tree - AUC-ROC: {roc_auc:.2f}")
```

Decision Tree - Accuracy: 0.75
Decision Tree - Precision: 0.79
Decision Tree - Recall: 0.85
Decision Tree - F1 Score: 0.81
Decision Tree - AUC-ROC: 0.71

- **Accuracy (0.75):** Ini menunjukkan bahwa 75% dari prediksi model adalah benar. Secara umum, ini adalah tingkat akurasi yang cukup baik.
- **Precision (0.79):** Precision mengukur proporsi prediksi positif yang benar-benar positif. Nilai 0.79 menunjukkan bahwa 79% dari prediksi positif benar-benar benar. Ini adalah nilai precision yang tinggi dan menunjukkan bahwa model menghasilkan sedikit false positives.
- **Recall (0.85):** Recall mengukur proporsi kasus positif yang benar-benar terdeteksi oleh model. Nilai 0.85 menunjukkan bahwa 85% dari kasus positif berhasil diidentifikasi oleh model. Ini adalah nilai recall yang tinggi, yang berarti model baik dalam mendeteksi kasus positif.
- **F1 Score (0.81):** F1 Score adalah rata-rata harmonis dari Precision dan Recall. Nilai 0.81 menunjukkan keseimbangan yang sangat baik antara Precision dan Recall.
- **AUC-ROC (0.71):** Area Under the Receiver Operating Characteristic Curve (AUC-ROC) mengukur kemampuan model untuk membedakan antara kelas positif dan negatif. Nilai 0.71 menunjukkan kemampuan yang moderat untuk diskriminasi antara kelas

positif dan negatif. Meskipun ini tidak setinggi metrik lain, nilai ini masih menunjukkan bahwa model memiliki kemampuan yang baik dalam membedakan antara dua kelas.

Kesimpulannya, meskipun model memiliki performa yang baik dalam presisi, recall, dan F1 score, perlu diperhatikan bahwa AUC-ROC yang rendah bisa menjadi perhatian. Hal ini mungkin menunjukkan bahwa model memiliki masalah dalam membedakan antara kelas positif dan negatif, yang bisa menjadi fokus untuk perbaikan lebih lanjut.