DATA PREPARATION DARI SUMBER OPEN SOURCE

disusun untuk memenuhi
tugas mata kuliah Pemrosesan Mesin B

oleh :

Farah Nasywa (2208107010051)
Iwani Khairina (2208107010078)
Dinda Maharani (2208107010081)

JURUSAN INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS SYIAH KUALA
2025

Pendahuluan

Penyakit kanker merupakan salah satu penyebab utama kematian di dunia, sehingga deteksi dini menjadi faktor penting dalam meningkatkan tingkat kelangsungan hidup pasien. Dengan kemajuan dalam bidang Machine Learning (ML), berbagai metode klasifikasi dapat digunakan untuk membantu dalam prediksi dan diagnosis kanker berdasarkan data medis pasien. Laporan ini bertujuan untuk mengevaluasi performa beberapa model machine learning dalam mengklasifikasikan data pasien kanker menggunakan dataset The_Cancer_data_1500_V2.csv. Berbagai model, seperti Logistic Regression, Random Forest, Support Vector Machine (SVM), dan K-Nearest Neighbors (KNN), dibandingkan berdasarkan metrik evaluasi seperti akurasi, precision, recall, dan F1-score. Melalui analisis ini, diharapkan dapat ditemukan model yang paling optimal untuk mendukung sistem prediksi kanker yang efisien dan akurat.

Data Description

Dataset yang digunakan dalam laporan ini merupakan dataset **The_Cancer_data_1500_V2.csv** yang mana kami dapatkan dari Kaggle.com :https://www.kaggle.com/datasets/rabieelkharoua/cancer-prediction-dataset
berisi informasi mengenai faktor-faktor risiko kanker pada individu. Setiap baris dalam dataset merepresentasikan seorang individu dengan berbagai atribut yang dapat mempengaruhi kemungkinan terkena kanker. Berikut adalah beberapa atribut utama dalam dataset:

- **Age**: Usia individu.
- **BMI**: Indeks Massa Tubuh (Body Mass Index).
- **Smoking**: Status merokok individu (0 = Tidak Merokok, 1 = Merokok).
- **Alcohol Intake**: Konsumsi alkohol individu.
- **Genetic Risk**: Risiko genetik terkait kanker (0 = Tidak Ada Risiko, 1 = Risiko Sedang, 2 = Risiko Tinggi).
- **Physical Activity**: Frekuensi aktivitas fisik individu.
- **Cancer History**: Riwayat kanker dalam keluarga (0 = Tidak, 1 = Ya).
- **Diagnosis**: Diagnosis kanker pada individu (0 = Tidak Terdiagnosis, 1 = Terdiagnosis).

Variabel Target

- **Diagnosis** : Variabel utama yang diprediksi, yang menunjukkan apakah seorang pasien menderita kanker.

Distribusi Data

- Dataset seimbang sehubungan dengan distribusi fitur dan mencakup variabilitas realistis dalam data pasien.

Penggunaan

Kumpulan data ini ditujukan untuk melatih dan menguji model pembelajaran mesin untuk prediksi kanker. Kumpulan data ini dapat digunakan untuk:

- Pelatihan dan evaluasi model.
- Analisis kepentingan fitur.
- Pembandingan algoritma.

Format Data

Data Loading

Pemuatan data merupakan langkah pertama dalam proses analisis. Langkah ini mencakup membaca dataset, memahami struktur data, serta melakukan pemeriksaan awal untuk memastikan bahwa data dalam kondisi siap untuk diproses lebih lanjut.

1. Membaca Dataset Dataset diunggah dalam format CSV (*Comma-Separated Values*), sehingga kita dapat menggunakan pustaka `pandas` untuk memuatnya ke dalam *DataFrame*.

```
import numpy as np  # Untuk operasi aljabar linear dan komputasi numerik
import pandas as pd  # Untuk pemrosesan data dan membaca file CSV
import matplotlib.pyplot as plt  # Untuk visualisasi data dasar
import seaborn as sns  # Untuk visualisasi data yang lebih kompleks dan menarik

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

2. Menampilkan Informasi Dataset. Setelah dataset dimuat, langkah selanjutnya adalah memahami struktur data. Ini dapat dilakukan dengan menampilkan informasi dasar dari dataset.

```
df = pd.read_csv(r'C:\Users\asus\Downloads\MachineLearning\The_Cancer_data_1500_V2.csv')

df.head()
```

[16]

	Age	Gender	BMI	Smoking	GeneticRisk	PhysicalActivity	AlcoholIntake	CancerHistory	Diagnosis
0	58	1	16.085313	0	1	8.146251	4.148219	1	1
1	71	0	30.828784	0	1	9.361630	3.519683	0	0
2	48	1	38.785084	0	2	5.135179	4.728368	0	1
3	34	0	30.040296	0	0	9.502792	2.044636	0	0
4	62	1	35.479721	0	0	5.356890	3.309849	0	1

```
df.dtypes
```

```
Age                  int64
Gender               int64
BMI                float64
Smoking              int64
GeneticRisk          int64
PhysicalActivity   float64
AlcoholIntake      float64
CancerHistory        int64
Diagnosis            int64
dtype: object
```

Dari output df.dtypes, kita dapat melihat jumlah entri dalam dataset, tipe data masing-masing kolom, serta apakah ada nilai yang hilang.

3. Mengecek Nilai Hilang

Sebelum melakukan analisis lebih lanjut, penting untuk memastikan bahwa tidak ada data yang hilang (*missing values*), karena ini dapat mempengaruhi hasil analisis.

```
df.isna().sum()
```

```
Age                0
Gender             0
BMI                0
Smoking            0
GeneticRisk        0
PhysicalActivity   0
AlcoholIntake      0
CancerHistory      0
Diagnosis          0
dtype: int64
```

Jika ditemukan nilai yang hilang, kita dapat menangani dengan beberapa pendekatan, seperti menghapus baris yang mengandung nilai kosong atau menggantinya dengan nilai median atau rata-rata.

Pembersihan Data (Data Cleaning)

Penggunaan sns.boxplot(data=df['BMI']) dalam proses cleansing data bertujuan untuk mendeteksi dan mengatasi outlier pada variabel BMI (Body Mass Index). Berikut adalah alasan mengapa box plot digunakan dalam tahap pembersihan data:

```
sns.boxplot(data=df['BMI'])
plt.title('BMI(Box Plot)')
plt.xlabel('BMI')
```

```
Text(0.5, 0, 'BMI')
```

1. Identifikasi Outlier

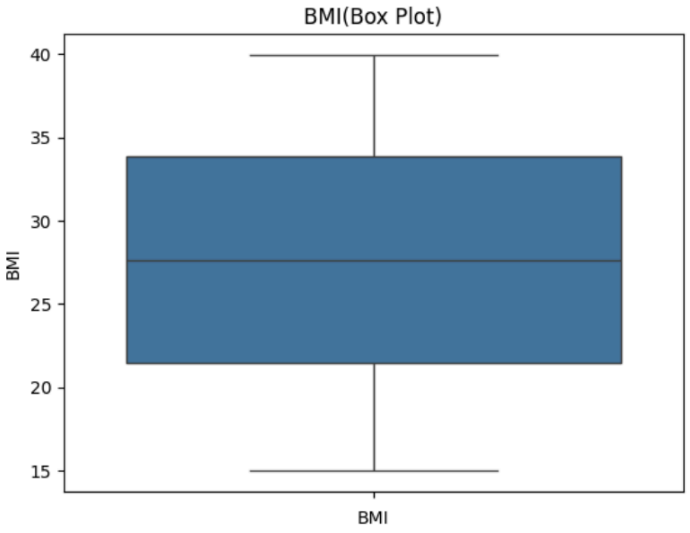
- Box plot menunjukkan distribusi data dengan menggambarkan nilai minimum, kuartil pertama (Q1), median (Q2), kuartil ketiga (Q3), dan nilai maksimum.
- Nilai yang berada di luar batas bawah (`Q1 - 1.5*IQR`) dan batas atas (`Q3 + 1.5*IQR`) dianggap sebagai outlier.

2. Memvisualisasikan Distribusi BMI

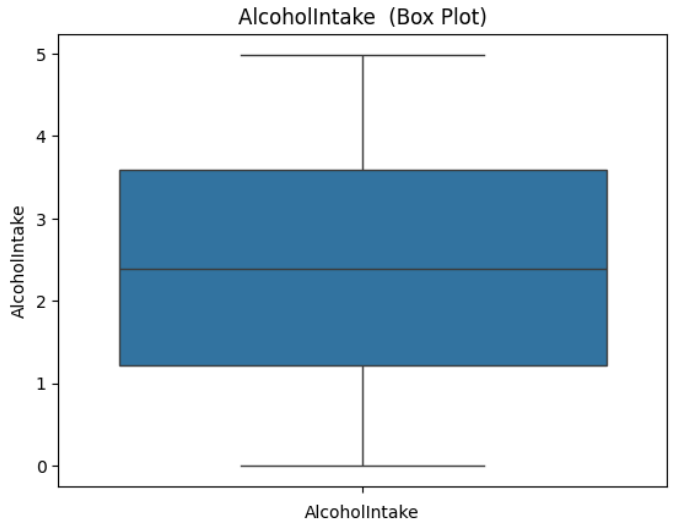
- Dengan box plot, kita bisa melihat apakah data BMI memiliki skewness atau outlier yang mencurigakan, yang dapat berdampak pada performa model Machine Learning.

3. Membersihkan Data dari Nilai Ekstrem

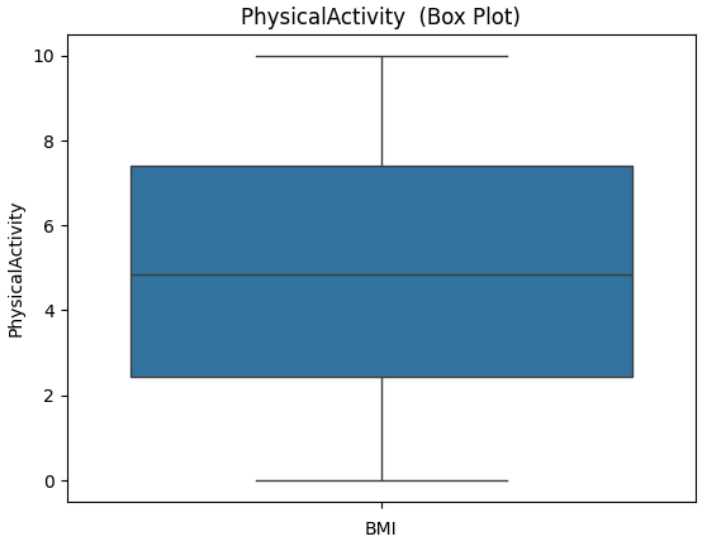
- Jika terdapat nilai outlier yang tidak masuk akal (misalnya, BMI yang terlalu kecil atau terlalu besar), maka bisa dilakukan penanganan seperti penghapusan atau transformasi data.



Gambar di atas ialah Mengidentifikasi distribusi nilai **BMI** pada dataset.



Gambar di atas ialah Mengetahui pola distribusi konsumsi alkohol pada dataset.



Gambar di atas ialah Memeriksa distribusi tingkat aktivitas fisik pada individu dalam dataset.

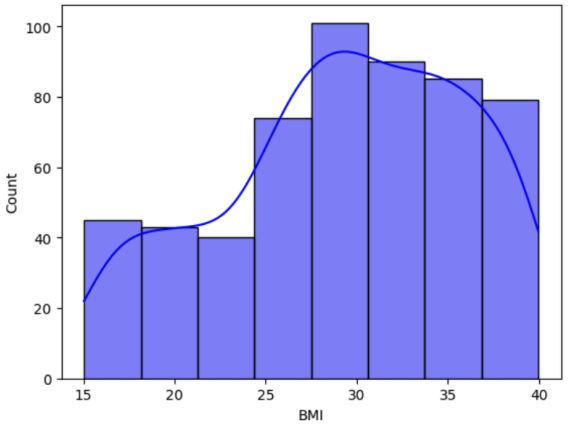
Data Understanding

Data Understanding merupakan tahap penting dalam eksplorasi data yang bertujuan untuk memahami karakteristik dataset, pola distribusi, dan kemungkinan hubungan antara variabel. Dalam analisis ini, distribusi BMI (Body Mass Index) pada pasien kanker dan non-kanker dianalisis menggunakan histogram dan boxplot.

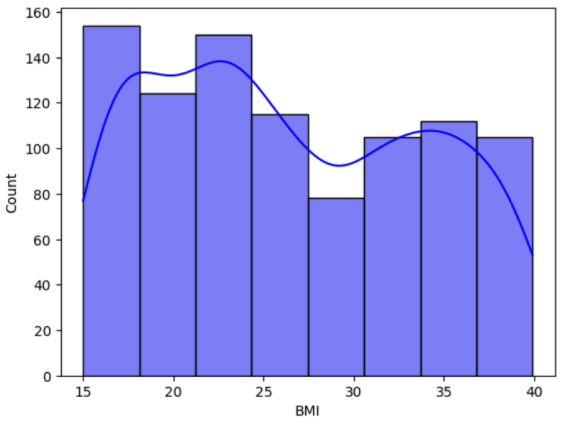
1. Histogram Distribusi BMI pada Pasien Kanker dan Non-Kanker

Kode berikut digunakan untuk memvisualisasikan distribusi **BMI** pada dua kelompok pasien:

- **Pasien kanker** (`cancerous_patients['BMI']`)
- **Pasien non-kanker** (`non_cancerous_patients['BMI']`)



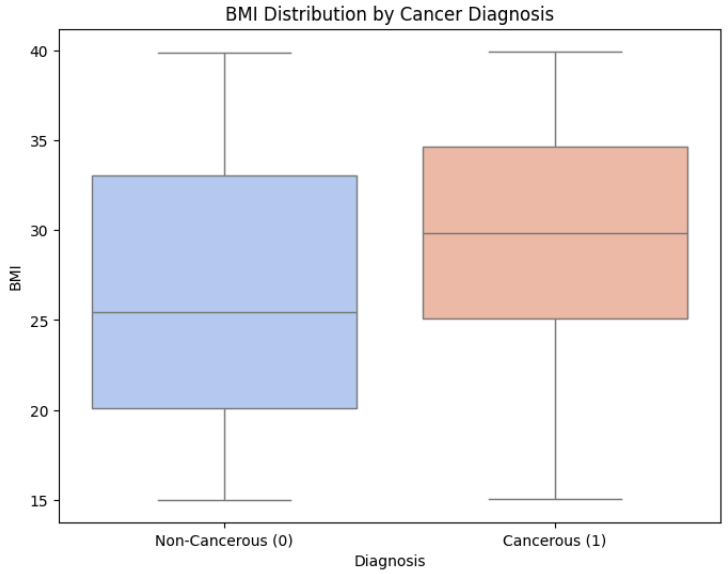
Gambar Distribusi BMI pada Pasien Kanker



Gambar Distribusi BMI pada Non-Pasien Kanker

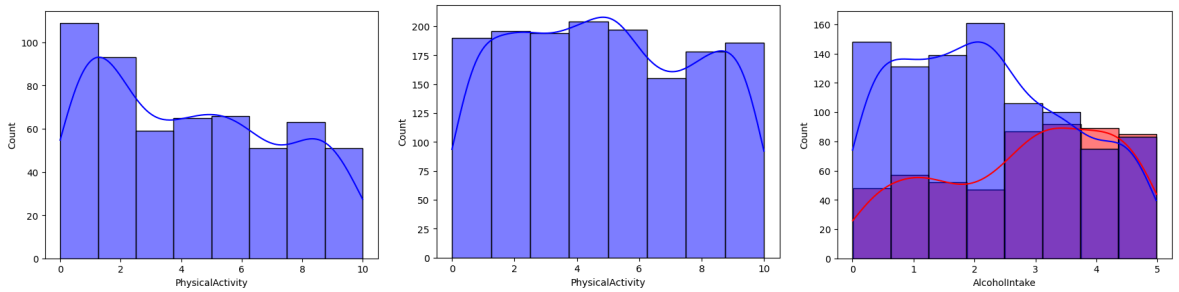
2. Boxplot Distribusi BMI Berdasarkan Diagnosis Kanker

Kode berikut digunakan untuk melihat distribusi BMI dalam bentuk boxplot berdasarkan kategori



3. Distribusi Aktivitas Fisik pada Pasien

Visualisasi ini menunjukkan bagaimana tingkat aktivitas fisik tersebar di antara pasien yang menderita kanker, membantu memahami pola kebiasaan fisik mereka.

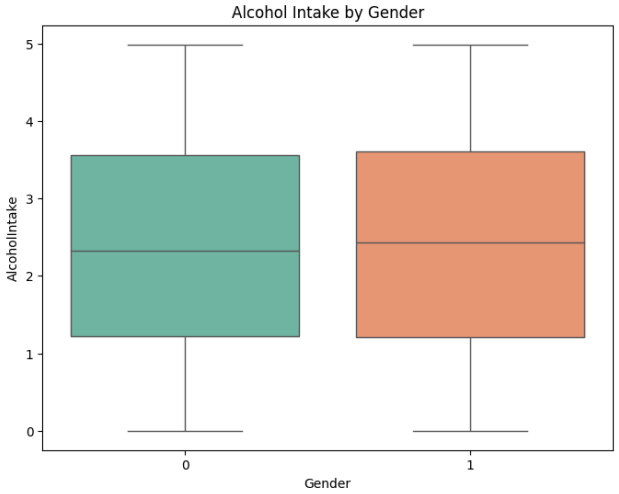


Visualisasi Data

Visualisasi data sangat penting dalam eksplorasi dataset untuk memahami pola distribusi, hubungan antar variabel, dan kemungkinan insight yang dapat digunakan untuk pemodelan Machine Learning. Berikut visualisasi yang menggunakan Boxplot, diagram batang, scattter plot, dan diagram pie.

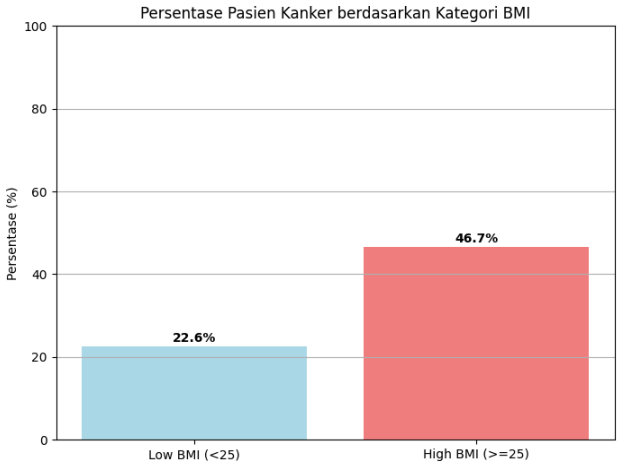
1. Boxplot Konsumsi Alkohol Berdasarkan Jenis Kelamin

```
plt.figure(figsize=(8, 6))
sns.boxplot(x='Gender', y='AlcoholIntake', data=df, palette='Set2')
plt.title('Alcohol Intake by Gender')
plt.show()
```

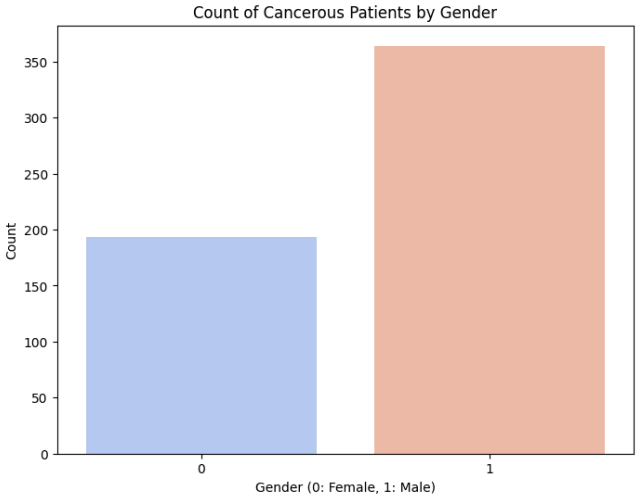


2. Diagram Batang

```
low_bmi_patients = df[df['BMI'] < 25]  # Memfilter pasien dengan BMI di bawah 25 (kategori BMI rendah)
high_bmi_patients = df[df['BMI'] >= 25]  # Memfilter pasien dengan BMI 25 atau lebih (kategori BMI tinggi)
cancerous_low_bmi = low_bmi_patients[low_bmi_patients['Diagnosis'] == 1].shape[0]
cancerous_high_bmi = high_bmi_patients[high_bmi_patients['Diagnosis'] == 1].shape[0]

total_low_bmi = low_bmi_patients.shape[0]
total_high_bmi = high_bmi_patients.shape[0]

percentage_cancerous_low_bmi = (cancerous_low_bmi / total_low_bmi) * 100 if total_low_bmi > 0 else 0
percentage_cancerous_high_bmi = (cancerous_high_bmi / total_high_bmi) * 100 if total_high_bmi > 0 else 0

labels = ['Low BMI (<25)', 'High BMI (>=25)']  # Label kategori BMI
percentages = [percentage_cancerous_low_bmi, percentage_cancerous_high_bmi]  # Persentase pasien kanker di setiap kategori

plt.figure(figsize=(8, 6))
plt.bar(labels, percentages, color=['lightblue', 'lightcoral'])

plt.title('Persentase Pasien Kanker berdasarkan Kategori BMI')
plt.ylabel('Persentase (%)')
plt.ylim(0, 100)  # Mengatur batas sumbu y hingga 100%
plt.grid(axis='y')  # Menambahkan garis bantu horizontal untuk meningkatkan keterbacaan

for i, v in enumerate(percentages):
    plt.text(i, v + 1, f'{v:.1f}%', ha='center', fontweight='bold')

plt.show()
```

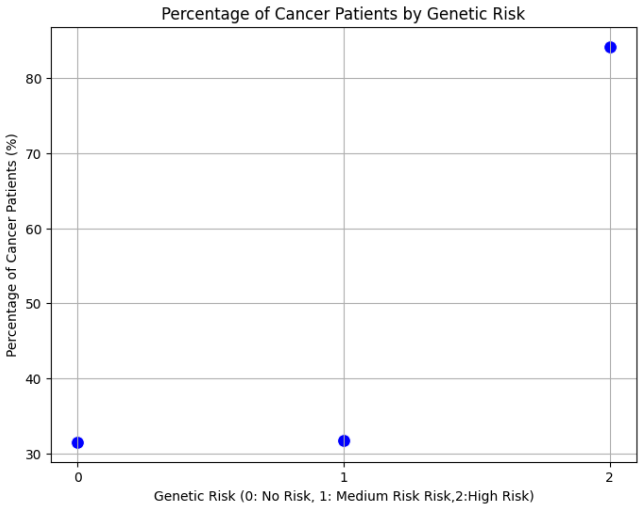



```
cancerous_patients = df[df['Diagnosis'] == 1]

# Membuat Grafik Batang (Count Plot) untuk Jenis Kelamin
plt.figure(figsize=(8, 6))
sns.countplot(x='Gender', data=cancerous_patients, palette='coolwarm')

# Menambahkan Judul dan Label
plt.title('Count of Cancerous Patients by Gender')
plt.xlabel('Gender (0: Female, 1: Male)')
plt.ylabel('Count')
plt.show()
```



3. Scatter plot pasien kanker berdasarkan risiko genetik

```
total_records_by_risk = df.groupby('GeneticRisk').size().reset_index(name='TotalRecords')
cancer_count_by_risk = df[df['Diagnosis'] == 1].groupby('GeneticRisk').size().reset_index(name='CancerCount')

# Menggabungkan Data Total Pasien dan Pasien Kanker
merged_data = pd.merge(total_records_by_risk, cancer_count_by_risk, on='GeneticRisk', how='left').fillna(0)

# Menghitung Persentase Pasien Kanker
merged_data['CancerPercentage'] = (merged_data['CancerCount'] / merged_data['TotalRecords']) * 100

# Membuat Scatter Plot
plt.figure(figsize=(8, 6))
sns.scatterplot(x='GeneticRisk', y='CancerPercentage', data=merged_data, s=100, color='blue')

# Menambahkan Judul dan Label
plt.title('Percentage of Cancer Patients by Genetic Risk')
plt.xlabel('Genetic Risk (0: No Risk, 1: Medium Risk Risk,2:High Risk)')
plt.ylabel('Percentage of Cancer Patients (%)')
plt.xticks([0, 1,2])  # Set x-ticks to match the genetic risk values
plt.grid(True)
plt.show()
```



4. Diagram Lingkaran Pasien dengan Riwayat Kanker

```
# Langkah 1: Memfilter pasien yang memiliki riwayat kanker
pasien_dengan_riwayat_kanker = df[df['CancerHistory'] == 1]

# Langkah 2: Menghitung jumlah pasien yang didiagnosis (Diagnosis = 1) dan tidak didiagnosis (Diagnosis = 0)
jumlah_diagnosis = pasien_dengan_riwayat_kanker['Diagnosis'].value_counts()
print(jumlah_diagnosis)

# Langkah 3: Menghitung persentase dari masing-masing kategori
persentase = (jumlah_diagnosis / jumlah_diagnosis.sum()) * 100

# Langkah 4: Membuat diagram lingkaran
plt.figure(figsize=(8, 6))
plt.pie(persentase, labels=['Didiagnosis (1)', 'Tidak Didiagnosis (0)'], autopct='%1.1f%%', startangle=90, colors=['red', 'green'])
plt.title('Proporsi Pasien dengan Riwayat Kanker Berdasarkan Diagnosis')
plt.axis('equal')  # Memastikan diagram lingkaran tetap berbentuk lingkaran
plt.show()
```

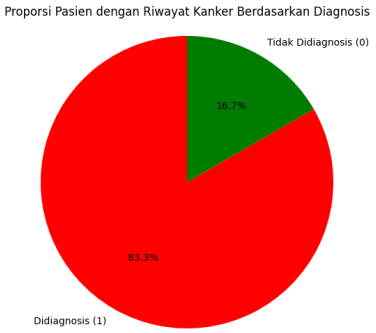
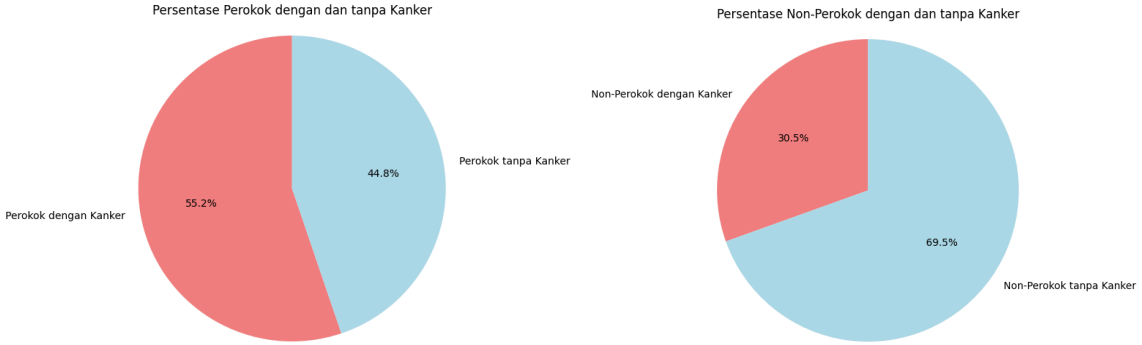
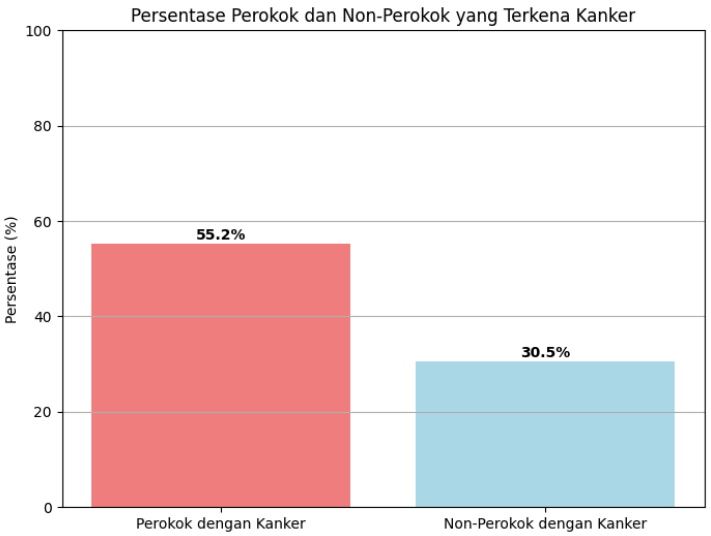


Diagram Perokok dan Non-Perokok



Dapat disimpulkan bahwa dalam diagram lingkaran ini terdapat, 83,3% dari data dengan riwayat kanker didiagnosis menderita kanker dan 55,2 persen perokok didiagnosis menderita kanker.



Sehingga presentase yang di peroleh 55,2 persen perokok didiagnosis menderita kanker sedangkan hanya 30 persen bukan perokok didiagnosis menderita kanker.

Model Initialization & Implementation

Model Machine Learning digunakan untuk memprediksi diagnosis kanker berdasarkan dataset yang tersedia. Dalam tahap *Model Initialization & Implementation*, beberapa algoritma diuji untuk menentukan model dengan performa terbaik.

Hasil evaluasi ditampilkan dalam bentuk tabel yang mencakup metrik berikut:

- **Accuracy**: Seberapa sering model memberikan prediksi yang benar.
- **Precision (0) & Precision (1)**: Kemampuan model dalam mengklasifikasikan masing-masing kelas dengan benar.
- **Recall (0) & Recall (1)**: Seberapa baik model menangkap kasus positif dan negatif.
- **F1-score (0) & F1-score (1)**: Rata-rata harmonik antara precision dan recall.
- **Support (0) & Support (1)**: Jumlah sampel dalam masing-masing kelas.
- **Implementation Time (s)**: Waktu eksekusi model.

	Model	Accuracy	Precision (0)	Recall (0)	F1-score (0)	\
0	Logistic Regression	0.8633	0.8593	0.9293	0.8930	
1	Random Forest	0.9300	0.9267	0.9620	0.9440	
2	SVM	0.8900	0.8794	0.9511	0.9138	
3	KNN	0.9000	0.9010	0.9402	0.9202	

	Precision (1)	Recall (1)	F1-score (1)	Support (0)	Support (1)	\
0	0.8713	0.7586	0.8111	184.0	116.0	
1	0.9358	0.8793	0.9067	184.0	116.0	
2	0.9109	0.7931	0.8479	184.0	116.0	
3	0.8981	0.8362	0.8661	184.0	116.0	

	Implementation Time (s)
0	0.0034
1	0.1984
2	0.0266
3	0.0138

Kesimpulan

Setelah mengevaluasi beberapa model machine learning untuk The_Cancer_data_1500_V2.csv, ditemukan bahwa Logistic Regression dan Random Forest memberikan hasil terbaik dalam keseimbangan antara akurasi, presisi, recall, dan efisiensi waktu. Logistic Regression memiliki akurasi tinggi (88.50%) dengan waktu eksekusi yang cepat, menjadikannya pilihan yang efisien untuk dataset ini.

Random Forest juga menunjukkan performa yang baik dengan kemampuan menangani fitur yang lebih kompleks, meskipun membutuhkan waktu eksekusi yang lebih lama dibandingkan Logistic Regression. SVM dan KNN memberikan hasil yang kompetitif tetapi memiliki keterbatasan dalam efisiensi, terutama dalam kecepatan prediksi untuk dataset yang lebih besar.

Dari hasil ini, Logistic Regression direkomendasikan sebagai model utama karena memberikan hasil yang stabil dan cepat, sedangkan Random Forest dapat digunakan untuk analisis lebih lanjut jika diperlukan model yang lebih kompleks.