Prediksi Deteksi Penyakit Kanker Payudara dengan Menggunakan Algoritma Decision Tree C4.5

Anggota Kelompok

KEVIN ADITYA
69875

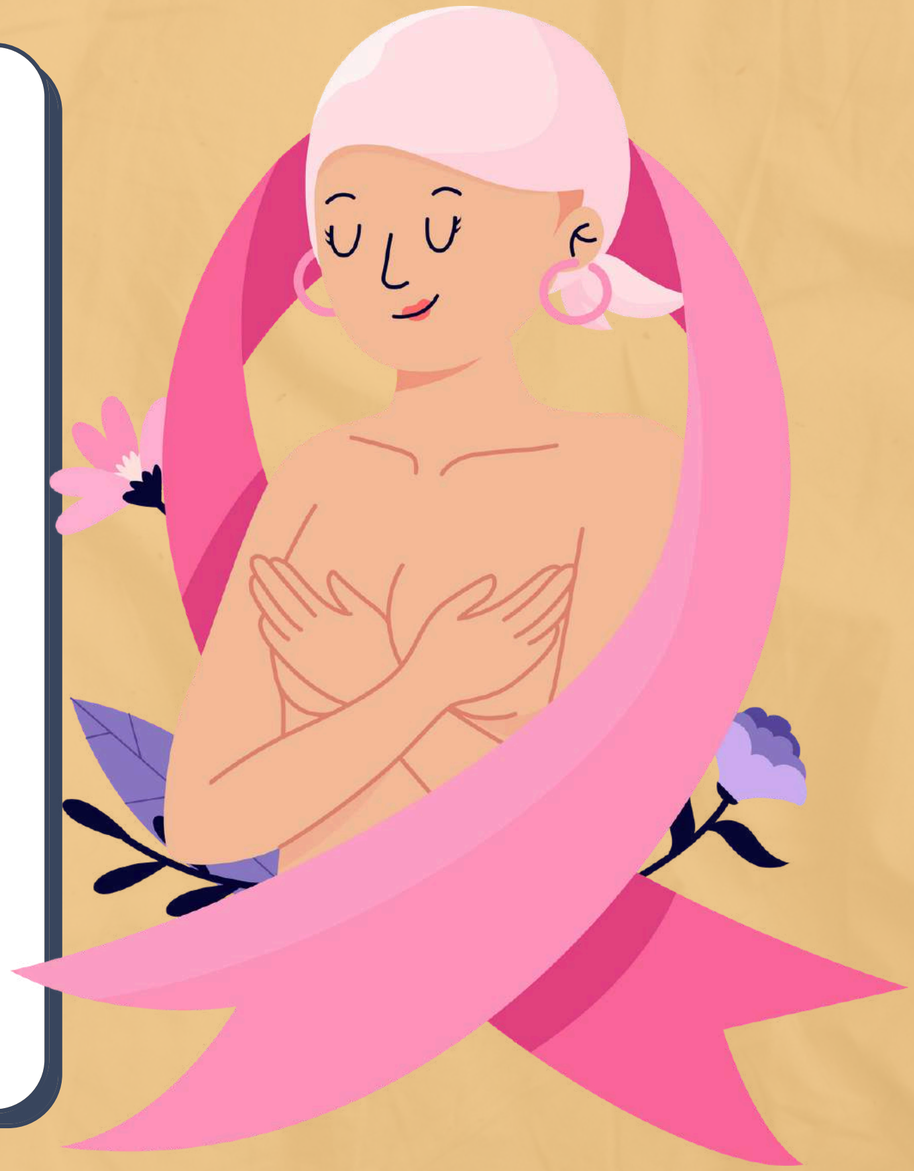
FIENA GUNAWAN
69579

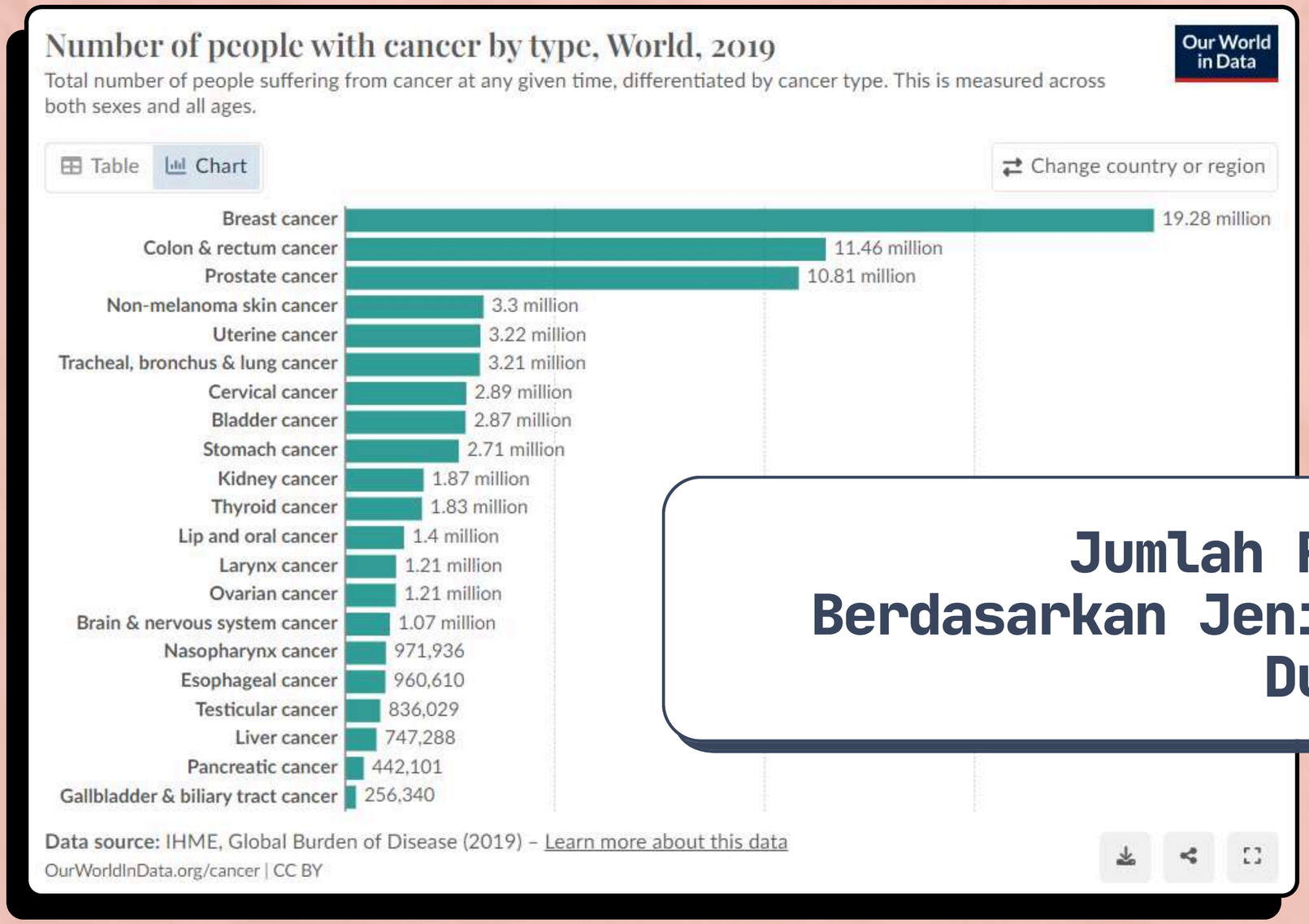
RAY ANTHONY
66655

NATHAN VILBERT
69903

Latar Belakang

Kanker payudara adalah jenis kanker umum yang berdampak besar pada kesehatan masyarakat, terutama perempuan. Jumlah kasus kanker payudara di Indonesia dan global terus meningkat, dengan perkiraan peningkatan di masa mendatang. Diagnosis kanker payudara sering memerlukan waktu dan hipotesis.

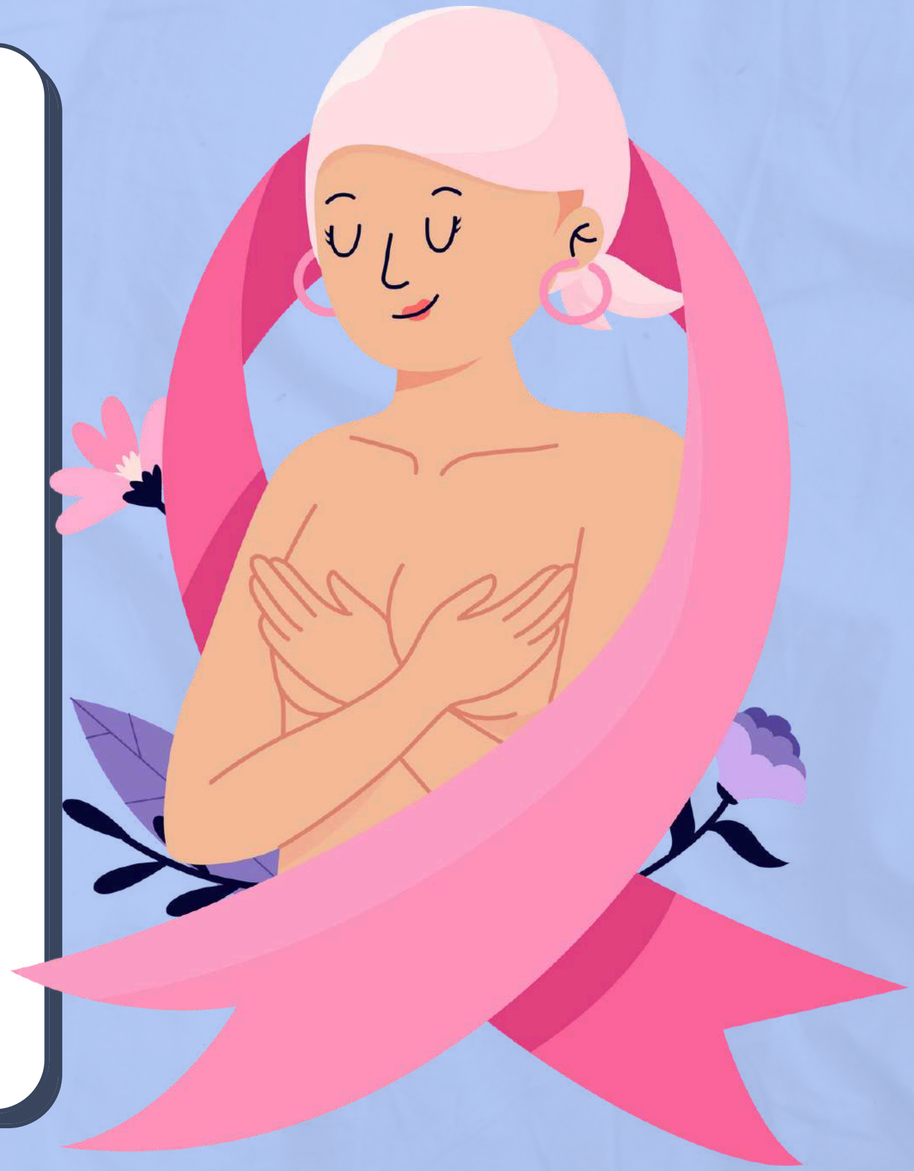




Jumlah Penderita Kanker Berdasarkan Jenisnya di Seluruh Dunia, Tahun 2019

Latar Belakang

Dalam konteks ini, Machine Learning, terutama algoritma Decision Tree C4.5, dapat menjadi alat efektif untuk deteksi dan klasifikasi kanker payudara. Penelitian ini bertujuan untuk meningkatkan pemahaman dan metode deteksi kanker payudara, dengan harapan dapat membantu dokter membuat diagnosis yang lebih tepat dan tepat waktu.



Rumusan Masalah

1

Apa saja faktor atau atribut kunci yang dapat digunakan dalam mengidentifikasi jenis kanker payudara?

2

Seberapa tinggi tingkat akurasi algoritma Decision Tree C4.5 dalam mengklasifikasikan jenis kanker payudara, yang akan diukur menggunakan accuration, confusion matrix, dan F1 score?

3

Apa saja keputusan yang bisa diambil dari hasil prediksi jenis kanker payudara menggunakan algoritma Decision Tree C4.5?

Batasan Masalah

FOKUS VARIABEL

Usia, konsumsi alkohol, merokok, riwayat keluarga kanker payudara, status menopause, terapi hormon, indeks massa tubuh, obesitas, penggunaan pil KB, paparan radiasi, kebiasaan menyusui, dan Status diagnosis.

FOKUS PENELITIAN

Prediksi status kanker payudara malignant (ganas) atau benign (jinak) berdasarkan biopsi.

METODE ANALISIS

Machine Learning, khususnya algoritma Decision Tree C4.5, untuk analisis data kanker payudara dengan fitur pruning untuk mengurangi overfitting.

Tujuan Penelitian

1

Mengidentifikasi faktor atau atribut kunci yang berperan dalam mengklasifikasikan jenis kanker payudara.

2

Mengevaluasi tingkat akurasi algoritma Decision Tree C4.5 dalam mengklasifikasikan jenis kanker payudara menggunakan metrik akurasi, matriks kebingungan, dan skor F1

3

Menentukan keputusan yang dapat diambil dari hasil analisis untuk mendukung diagnosis dan pengelolaan kanker payudara secara efektif.

Manfaat Penelitian

Meningkatkan pemahaman tentang faktor atau atribut kunci yang mempengaruhi jenis kanker payudara

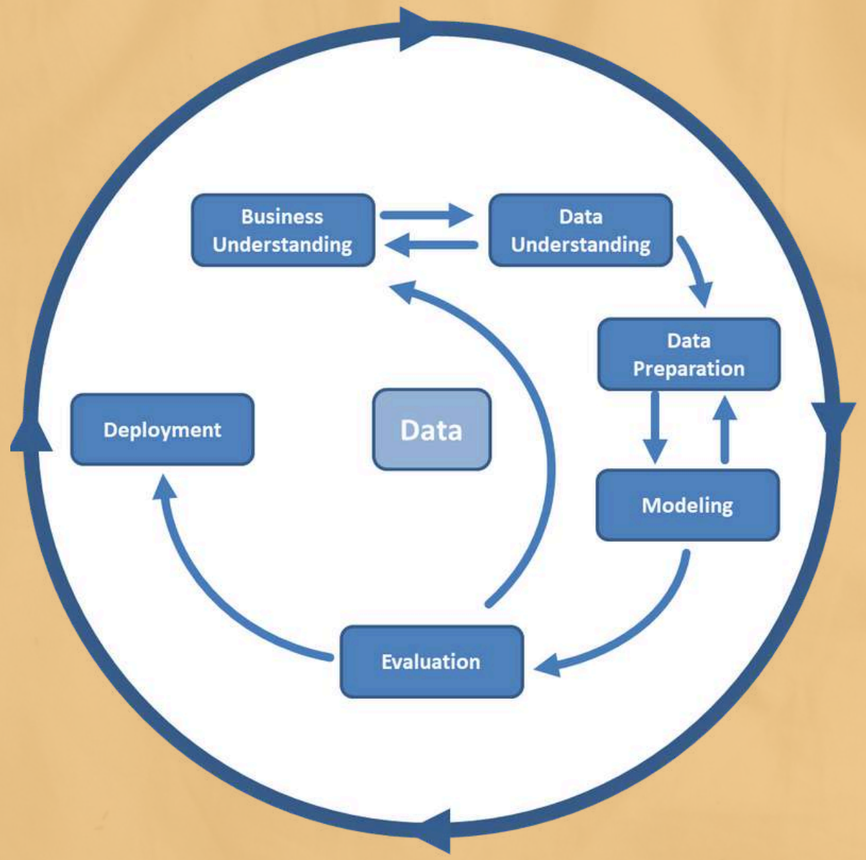
Memberikan pemahaman yang lebih mendalam tentang tingkat akurasi, efektivitas, dan kehandalan algoritma Decision Tree C4.5 dalam mengklasifikasikan jenis kanker payudara

Memungkinkan para praktisi medis untuk membuat keputusan yang lebih tepat dan efisien dalam diagnosis kanker payudara

Telaah Literatur

Supervised Machine Learning

Supervised Machine Learning (SML) menggunakan data latihan yang telah diberi label, di mana mesin belajar untuk mengenali pola dan membuat prediksi berdasarkan input yang diketahui. Contoh penerapan SML termasuk dalam masalah klasifikasi, di mana mesin menggunakan data pelatihan untuk mengklasifikasikan data baru berdasarkan pengalaman belajarnya.

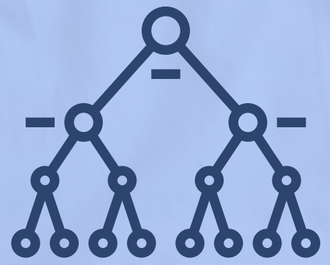


CRISP-DM

CRISP-DM adalah model proses untuk proyek data mining dengan enam tahapan: pemahaman bisnis, pemahaman data, persiapan data, pemodelan, evaluasi, dan implementasi.

Klasifikasi Data

Klasifikasi data mengelompokkan data untuk mengungkap pola dan membangun model prediktif, berdampak signifikan dalam berbagai bidang seperti pengenalan pola untuk meningkatkan efisiensi dan akurasi diagnosis medis, serta mendukung pengambilan keputusan berbasis data.



Algoritma Decision Tree

Decision Tree adalah teknik klasifikasi dalam data mining yang memproses data besar dan menghasilkan aturan konseptual dengan mudah, menggunakan algoritma seperti ID3, C4.5, dan CART dengan pengukuran entropi dan Information Gain untuk pembentukannya.

RUMUS ENTROPI

$$E = -\sum_{i=1}^{c} p_i \cdot \log_2(p_i)$$

RUMUS INFORMATION GAIN

$$Gain(S, A) = \sum_{v \in V(A)} \frac{|S_v|}{|S|} \cdot Entropy(S_v)$$

Decision Tree C4.5

C4.5 adalah pengembangan dari ID3 dalam data mining, di mana pemilihan node dilakukan berdasarkan atribut dengan gain tertinggi, menggunakan Gain Ratio sebagai parameter pemilihan node pembagi untuk meningkatkan akurasi dan efisiensi pada pembentukan Decision Tree.

RUMUS ENTROPI

$$E = -\sum_{i=1}^{c} p_i \cdot \log_2(p_i)$$

RUMUS INFORMATION GAIN C4.5

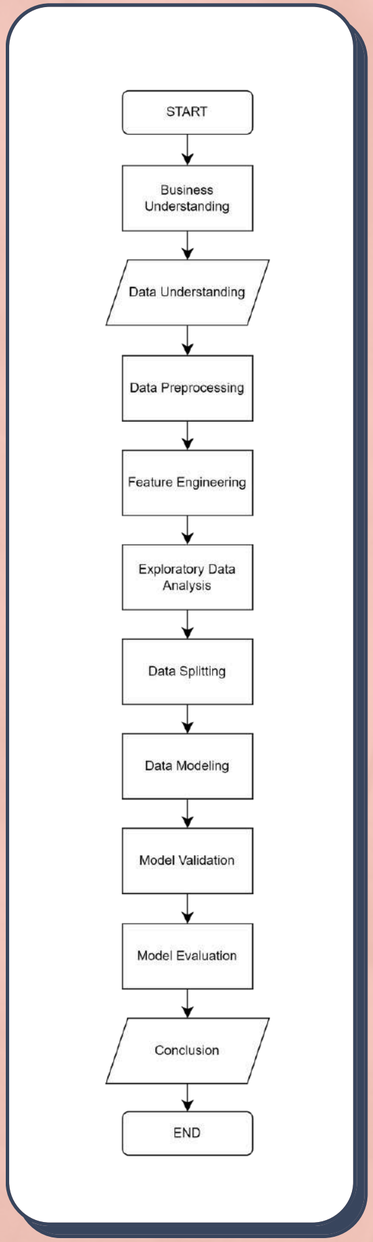
$$\text{Gain}(A) = E(S) - \sum_v \frac{|S_v|}{|S|} \cdot E(S_v)$$

RUMUS SPLIT INFO DAN GAIN RATIO

$$\text{Split Info}(A) = -\sum_i \frac{|S_i|}{|S|} \cdot \log_2\left(\frac{|S_i|}{|S|}\right)$$

$$\text{Gain Ratio}(A) = \frac{\text{Gain}(A)}{\text{Split Info}(A)}$$

METODOLOGI



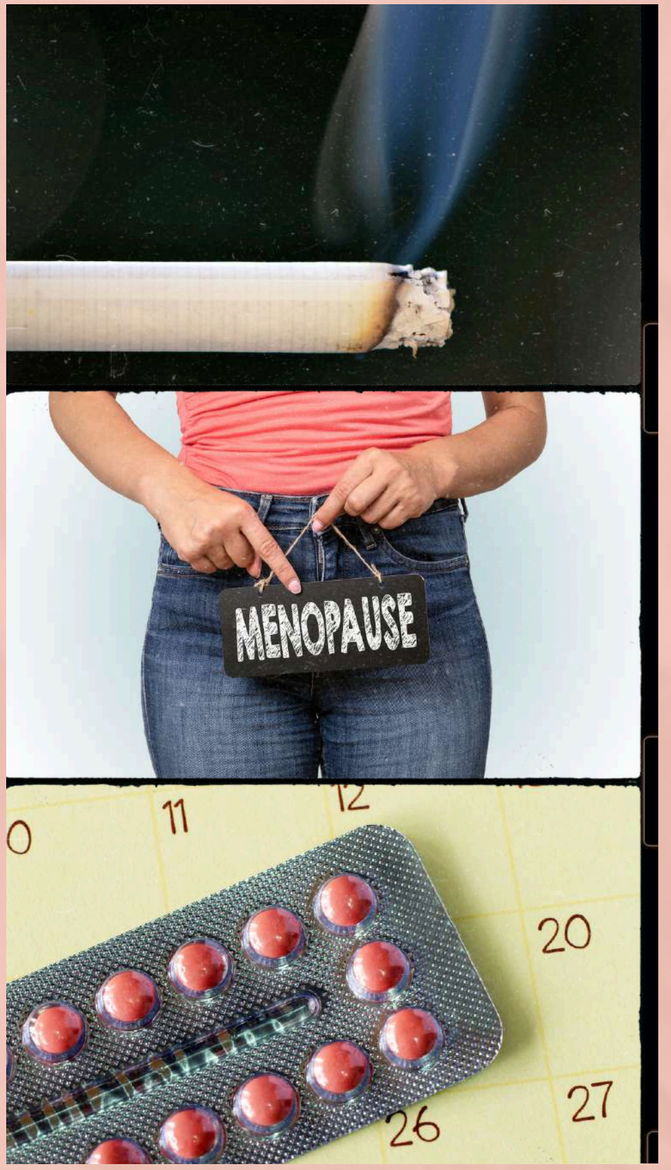
Flow Metodologi Penelitian

No	Feature	Deskripsi
1.	Age	Umur
2.	Residence Location	Lokasi Tempat Tinggal
3.	Alcohol Intake	Tingkat Konsumsi Alkohol
4.	Smoking Status	Status Merokok
5.	Family History of Breast Cancer	Riwayat Kanker Payudara Keluarga
6.	Number of Children	Jumlah Anak
7.	Age at Menarche	Usia Menstruasi Pertama Kali
8.	Menopausal Status	Status Menopause
9.	Hormone Replacement Therapy Use	Penggunaan Terapi Penggantian Hormon
10	Oral Contraceptive Use	Penggunaan Kontrasepsi Oral
11.	Breast Swelling	Kondisi Pembengkakkan Payudara
12.	Breast Lump	Kehadiran Benjol pada Payudara
13.	Breast Pain	Kehadiran Nyeri pada Payudara
14.	Breast Biopsy	Pernahkah Menjalani Biopsi Payudara
15.	Weight	Berat Badan
16.	Height	Tinggi Badan
17.	BMI	Indeks Massa Tubuh
18.	Obesity	Kategori Obesitas
19.	Exposure to Radiation	Paparan Radiasi yang Dialami
20.	Occupation	Banyaknya Pekerjaan
21.	Breast Feeding	Status Menyusui
22.	Diagnosis Status	Status Diagnosis Kanker

INPUT DATA Breast Cancer

BUSINESS UNDERSTANDING

Dalam tahap business understanding dalam proses CRISP-DM, fokusnya adalah memahami faktor-faktor risiko yang berkontribusi pada kanker payudara seperti merokok, status menopause, penggunaan pil KB, dan lain-lain. Langkah ini melibatkan identifikasi dan analisis terhadap variabel yang mempengaruhi kemungkinan seseorang terkena kanker payudara, penting untuk mengarahkan penelitian lebih lanjut dalam pengembangan model klasifikasi menggunakan machine learning. Pemahaman mendalam tentang faktor-faktor risiko ini krusial dalam memahami data yang akan digunakan pada tahap selanjutnya, yaitu tahap pemahaman data.



DATA UNDERSTANDING

Import Package Library

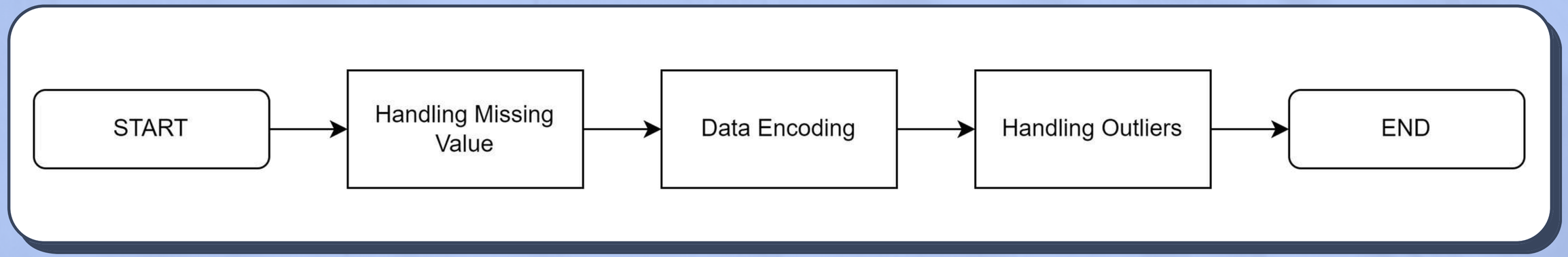
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from scipy.stats import chi2_contingency
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeClassifier, plot_tree
```

Import Data

```
df = pd.read_csv("BCdataset full.csv", sep=',')
df
```

	Age	Residence_Location	Alcohol_Intake	Smoking_Status	Family_history_of_breast_cancer	Number_of_children	Age_at_menarche	Menopausal_Status
0	32	2	0	0	0	0	13	1
1	60	2	0	0	0	12	16	3
2	44	3	0	0	0	0	16	2
3	74	1	1	1	0	5	13	4
4	64	1	0	0	0	0	14	4
...	...	...	...	...	...	...	...	...
1892	54	1	1	1	1	5	15	3
1893	39	1	0	0	0	6	12	1
1894	42	3	1	0	0	4	12	2
1895	35	1	0	0	1	2	14	1
1896	52	1	0	0	0	3	14	3

Preprocessing Data



Handling Missing Value

```
df.isnull().sum()
```

```
Age                                0
Residence_Location                 0
Alcohol_Intake                     0
Smoking_Status                     0
Family_history_of_breast_cancer    0
Number_of_children                 0
Age_at_menarche                    0
Menopausal_Status                  0
Hormone_replacement_therapy_use    0
Oral_contraceptive_use             0
Breast_Swelling                    0
Breast_Lump                        0
Breast_Pain                        0
Breast_Biopsy                      0
Weight                             0
Height                             0
BMI                                0
Obesity                            0
Exposure_to_radiation              0
Occupation                         0
Breast_Feeding                     0
Diagnosis_Status                   0
dtype: int64
```

Data Encoding

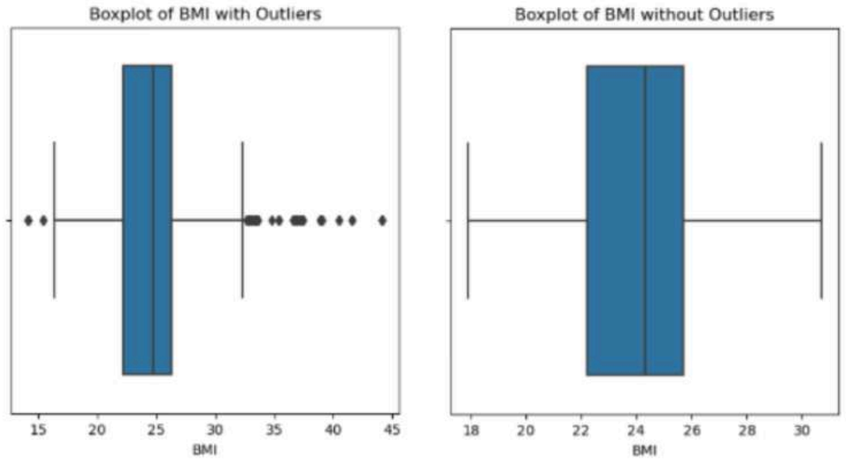
```
# Proses Encoding
df['Alcohol_Intake'] = df['Alcohol_Intake'].replace({1: 'low', 0: 'high'})
df['Smoking_Status'] = df['Smoking_Status'].replace({1: 'non-smoker', 0: 'smoker'})
df['Family_history_of_breast_cancer'] = df['Family_history_of_breast_cancer'].replace({1: 'no', 0: 'yes'})
df['Menopausal_Status'] = df['Menopausal_Status'].replace({1: 'Premenopause', 2: 'Perimenopause',
                                                           3:'Menopause', 4:'Postmenopause'})
df['Hormone_replacement_therapy_use'] = df['Hormone_replacement_therapy_use'].replace({1: 'no', 0: 'yes'})
df['Oral_contraceptive_use'] = df['Oral_contraceptive_use'].replace({1: 'no', 0: 'yes'})
df['Obesity'] = df['Obesity'].replace({0: 'obesity', 1: 'non-obesity'})
df['Exposure_to_radiation'] = df['Exposure_to_radiation'].replace({1: 'no', 0: 'yes'})
```

Output Encoding

Age	Alcohol_Intake	Smoking_Status	Family_history_of_breast_cancer	Menopausal_Status	Hormone_replacement_therapy_use	Oral_contraceptive_use
32	low	non-smoker	no	Premenopause	no	no
60	low	non-smoker	no	Menopause	no	no
44	low	non-smoker	no	Perimenopause	yes	yes
74	high	smoker	no	Postmenopause	no	no
64	low	non-smoker	no	Postmenopause	no	no

Handling Outliers

```python
Q1 = df['BMI'].quantile(q=0.25)
Q3 = df['BMI'].quantile(q=0.75)
IQR = Q3 - Q1

lower_bound = Q1 - 1.1 * IQR
upper_bound = Q3 + 1.1 * IQR

# visualisasi Boxplot waktu masih terdapat outlier
plt.figure(figsize=(5, 5))
sns.boxplot(x=df['BMI'])
plt.title('Boxplot of BMI with Outliers')
plt.show()

# Menghilangkan Outlier
outliers = df[(df['BMI'] < lower_bound) | (df['BMI'] > upper_bound)]
df = df[(df['BMI'] >= lower_bound) & (df['BMI'] <= upper_bound)]

# Visualisasi Boxplot sesudah Outlier hilang
plt.figure(figsize=(5, 5))
sns.boxplot(x=df['BMI'])
plt.title('Boxplot of BMI without Outliers')
plt.show()

# Jumlah data ketika Outlier sudah hilang
print("Shape of data without outliers:", df.shape)
```



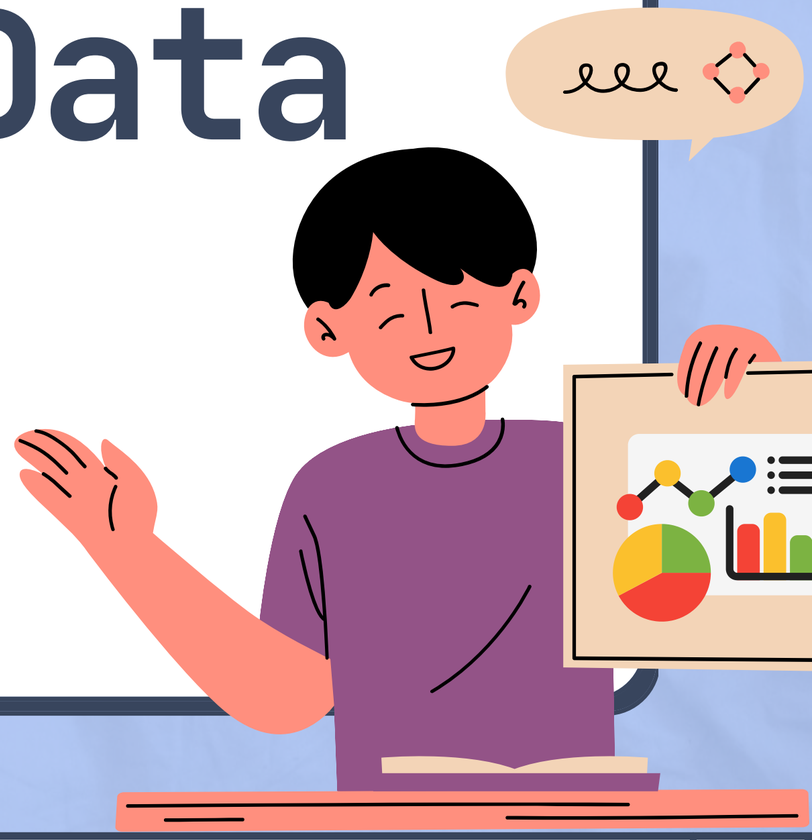

```
Shape of data with outliers: (1897, 15)

Shape of data without outliers: (1725, 15)
```

Feature Engineering

```
# Menghapus kolom yang tidak diperlukan
df = df.drop(columns=['Residence_Location', 'Number_of_children', 'Age_at_menarche',
                      'Occupation', 'Breast_Biopsy', 'Weight', 'Height', 'Breast_Swelling',
                      'Breast_Lump','Breast_Pain','Breast_Feeding'])
```

Dalam proses feature engineering, metode selection digunakan untuk memilih fitur-fitur yang relevan dari dataset dan menghapus fitur-fitur yang tidak memberikan kontribusi signifikan terhadap model. Kolom-kolom yang dihapus meliputi 'Residence_Location', 'Number_of_children', 'Age_at_menarche', 'Occupation', 'Breast_Biopsy', 'Weight', 'Height', 'Breast_Swelling', 'Breast_Lump', 'Breast_Pain', dan 'Breast_Feeding'. Penghapusan kolom-kolom ini membuat DataFrame lebih fokus pada atribut penting untuk analisis data.

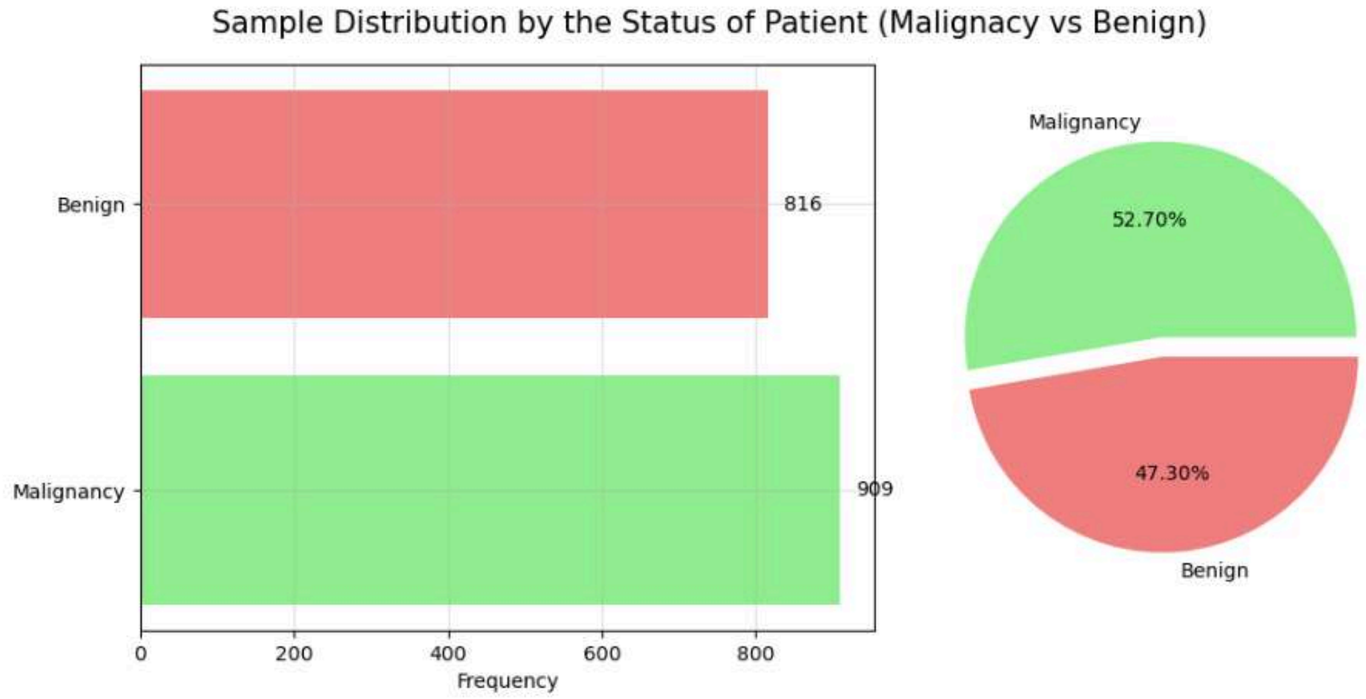
Exploratory Data Analysis



VISUALISASI STATUS DIAGNOSIS (CODE)

```
status = df['Diagnosis_Status'].value_counts()
colors = ['lightgreen','lightcoral']

figure, axes = plt.subplots(1,2, figsize=(10,5), gridspec_kw={'width_ratios':[1.5,1]})
axes[0].barh(y=status.index, width=status.values, color=colors)
axes[0].set_xlabel('Frequency')

for index, values in enumerate(status):
    axes[0].text(values+20, index, str(values), va='center')

axes[0].grid(alpha=0.4)

axes[1].pie(status.values, labels=status.index, autopct='%.2f%%', explode=([0.05]*len(status.index)), colors=colors)
figure.suptitle('Sample Distribution by the Status of Patient (Malignacy vs Benign)', fontsize=15)
plt.tight_layout(pad=1)
plt.show()
```

COUNTPLOT & PIE CHART STATUS DIAGNOSIS



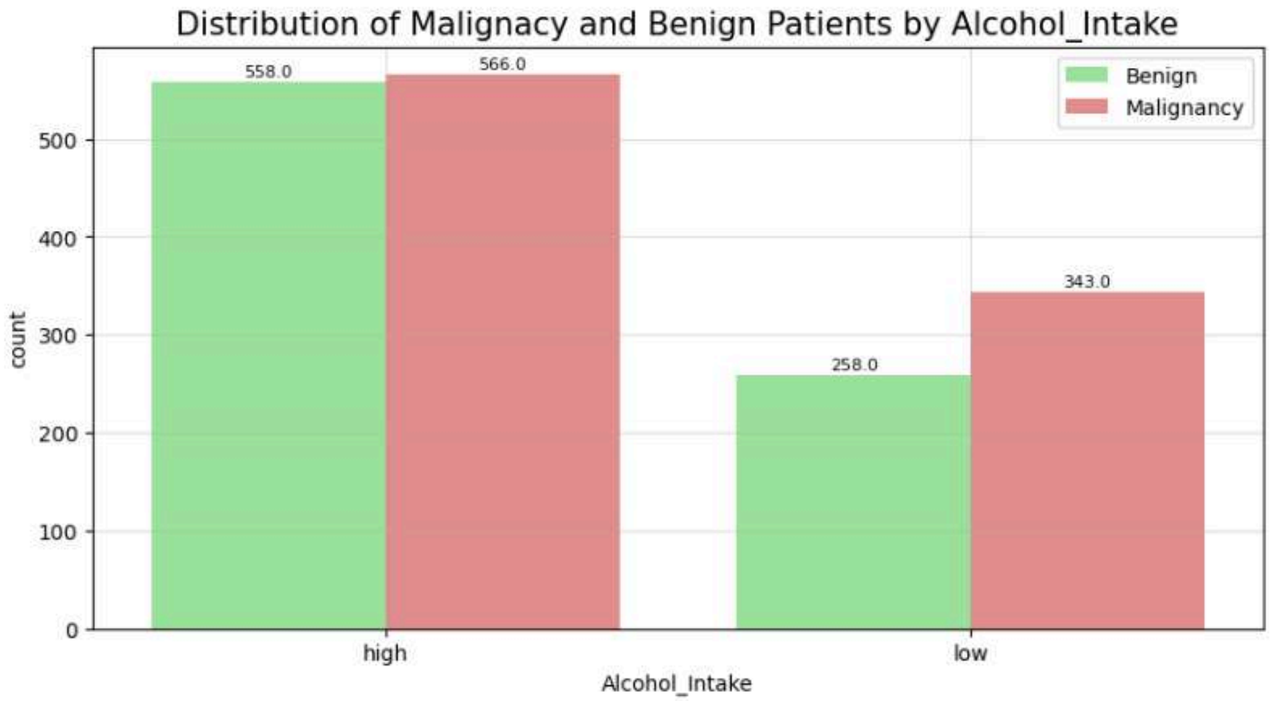
FUNGSI VISUALISASI (CODE)

```
def visualisasi(feature):
    plt.figure(figsize=(10,5))
    ax = sns.countplot(data=df, x=df[feature], hue='Diagnosis_Status', palette=colors)
    plt.grid(alpha=0.4)
    plt.title(f'Distribution of Survived and Dead Patients by {feature}', fontsize=15)

    # Ukuran Font label sumbu x dan y
    plt.tick_params(axis='x', labelsize=10)
    plt.tick_params(axis='y', labelsize=10)

    # Legend
    plt.legend(fontsize=10)

    # Nilai pada bar
    for p in ax.patches:
        ax.annotate(f'{p.get_height()}', (p.get_x() + p.get_width() / 2., p.get_height()),
                    ha='center', va='center', fontsize=8, color='black', xytext=(0, 5),
                    textcoords='offset points')
    plt.show()

    # Contingency Table
    contingency_table = pd.crosstab(df[feature], df['Diagnosis_Status'])
    print('Contingency Table: ')
    display(contingency_table)
    res = chi2_contingency(contingency_table)
    pvalue = round(res[1], 4)
```

COUNTPLOT STATUS KONSUMSI ALKOHOL

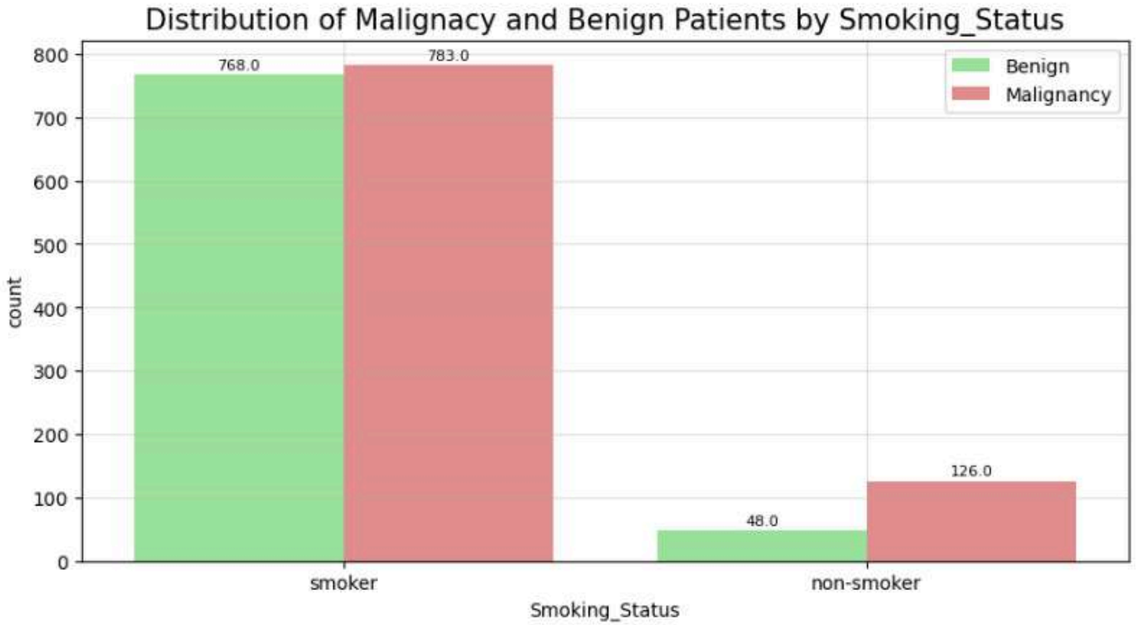
Contingency Table:

Diagnosis_Status	Benign	Malignancy
Alcohol_Intake		
high	558	566
low	258	343



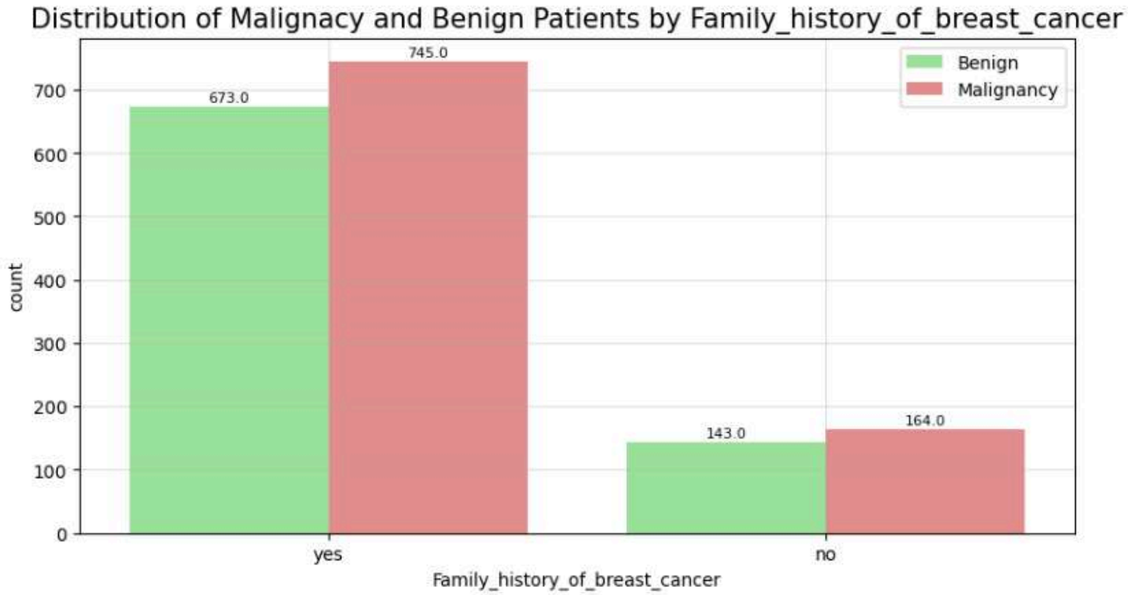
COUNTPLOT STATUS MEROKOK

Diagnosis_Status	Benign	Malignancy
Smoking_Status		
non-smoker	48	126
smoker	768	783

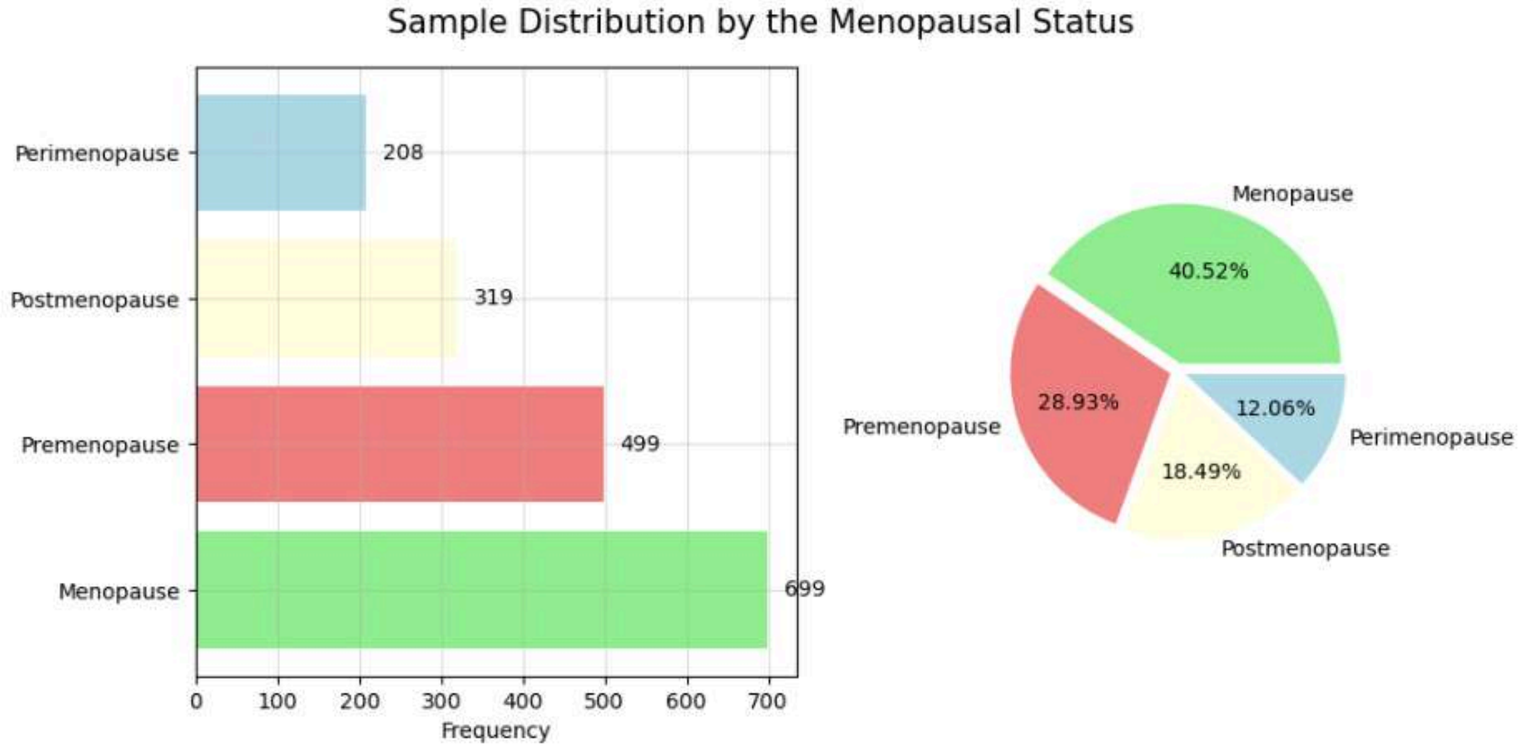


COUNTPLOT STATUS PENYAKIT KETURUNAN

Diagnosis_Status	Benign	Malignancy
Family_history_of_breast_cancer		
no	143	164
yes	673	745

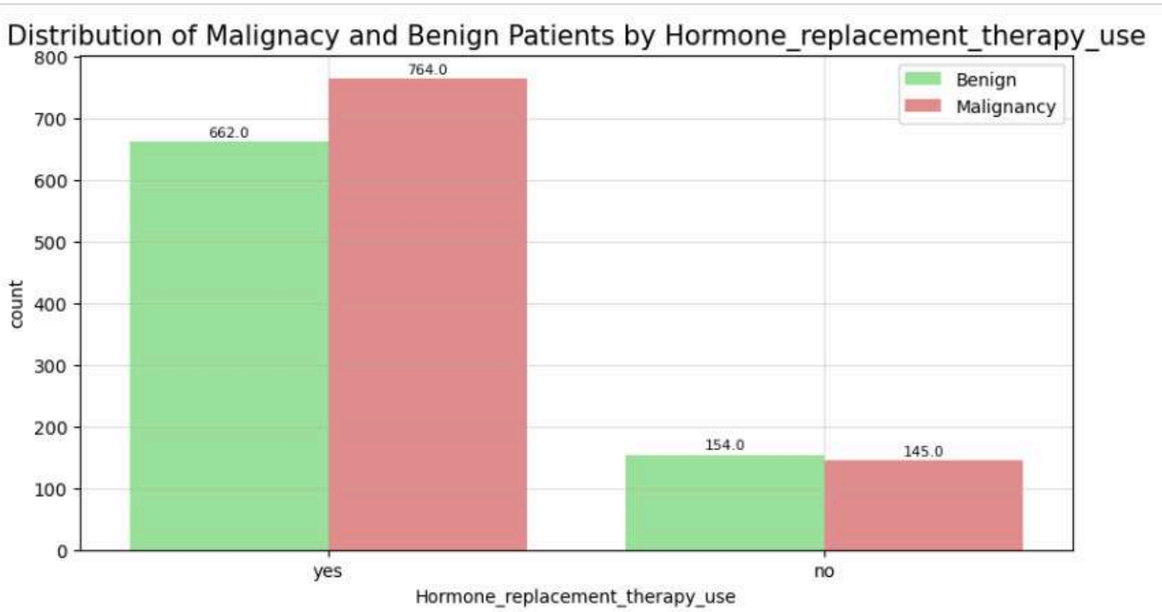


COUNTPLOT STATUS MENOPAUSE



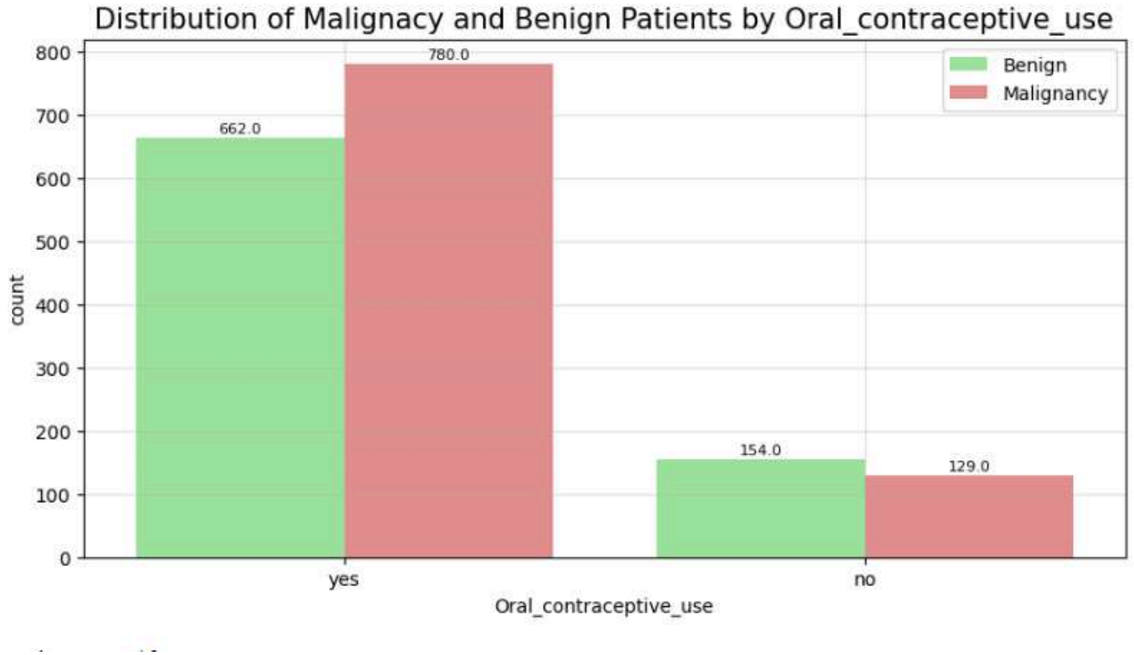
COUNTPLOT STATUS TERAPI HORMON

Hormone_replacement_therapy_use	Diagnosis_Status Benign	Malignancy
no	154	145
yes	662	764



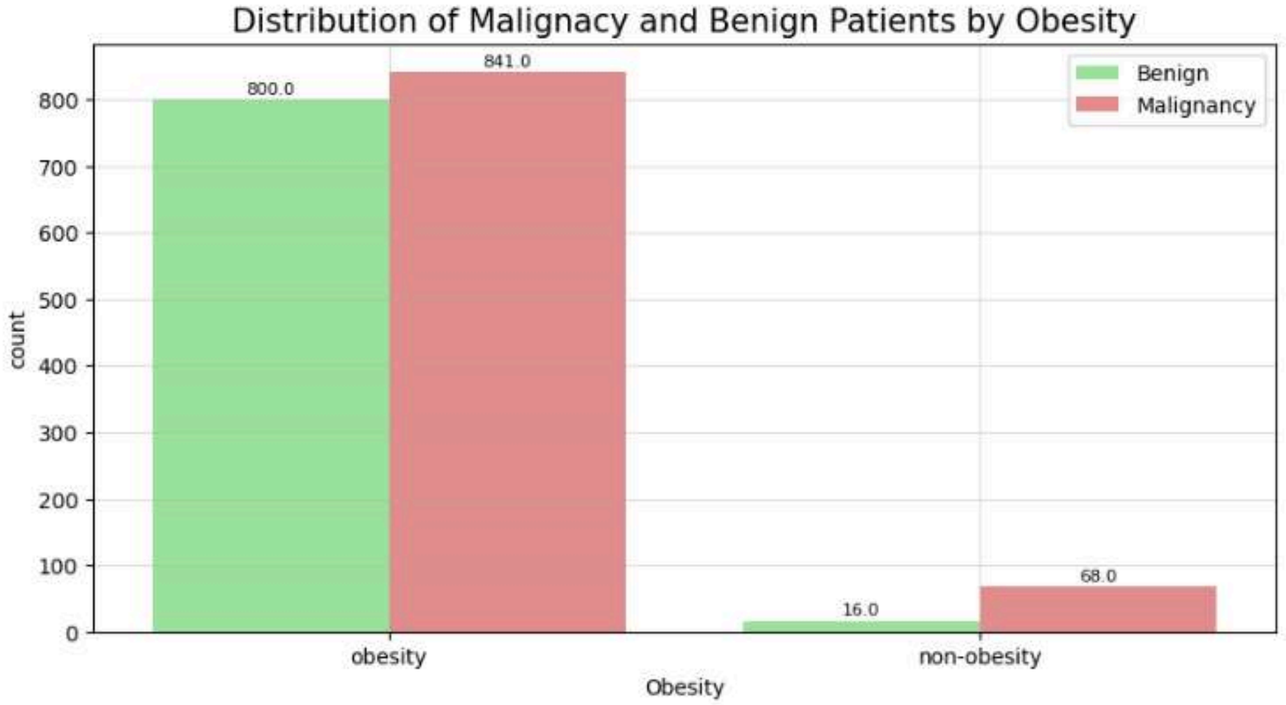
COUNTPLOT STATUS KONTRASEPSI

Diagnosis_Status	Benign	Malignancy
Oral_contraceptive_use		
no	154	129
yes	662	780



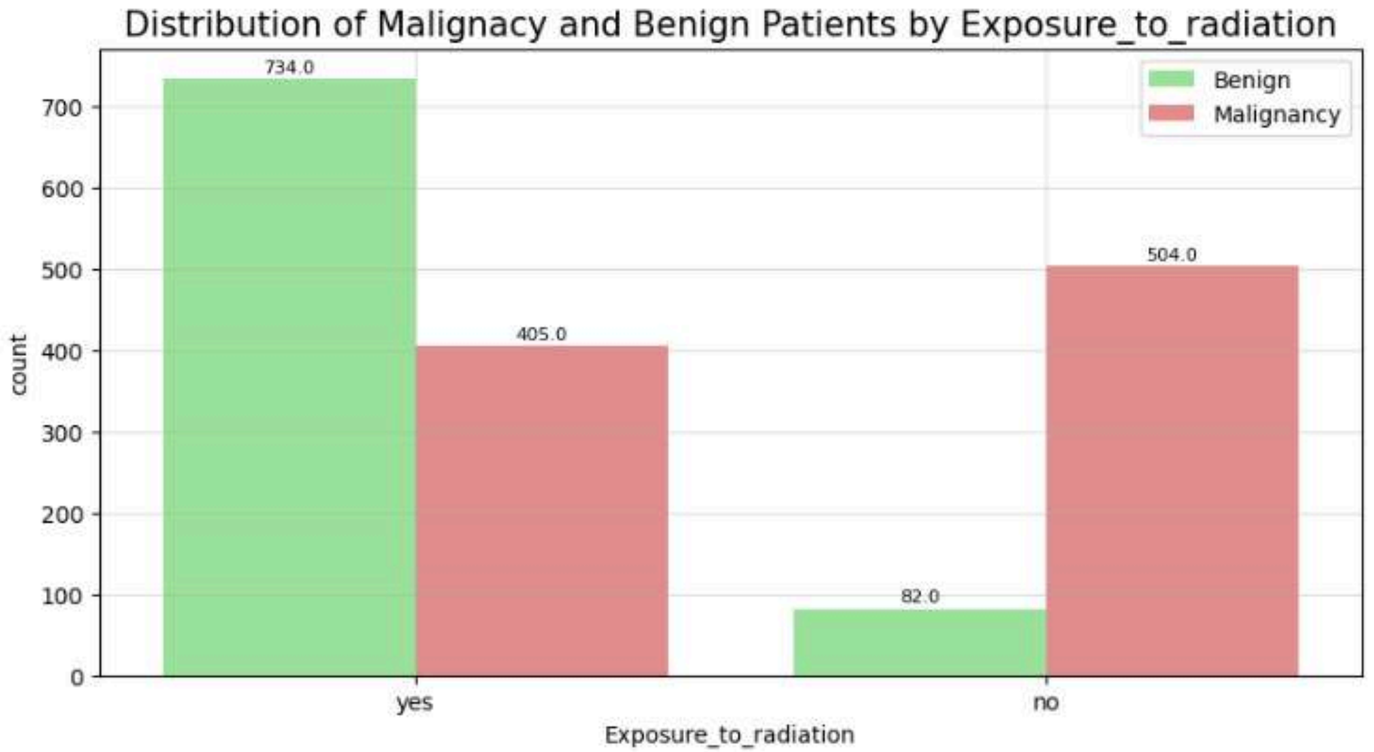
COUNTPLOT STATUS OBESITAS

Diagnosis_Status Obesity	Benign	Malignancy
non-obesity	16	68
obesity	800	841



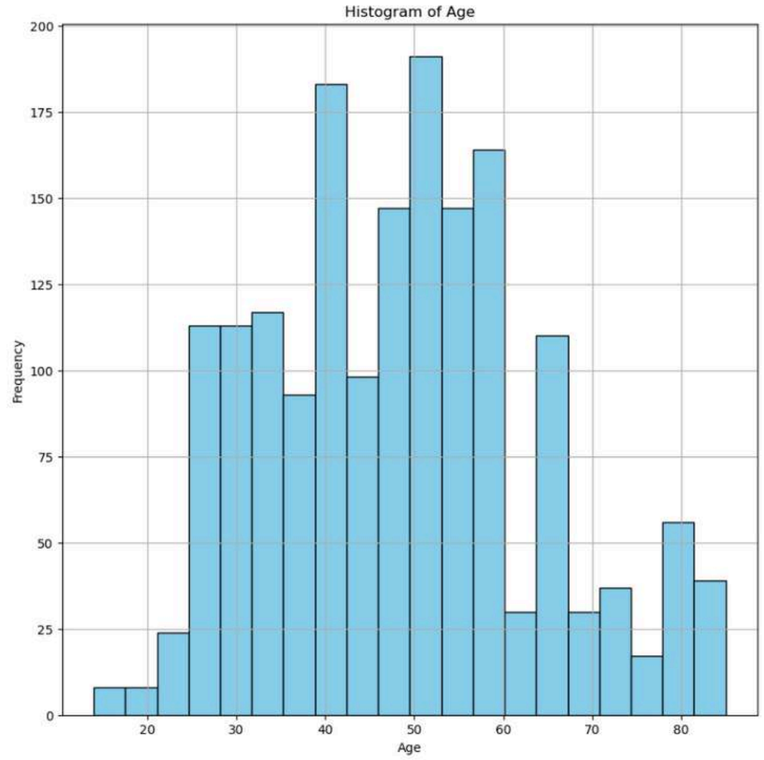
COUNTPLOT STATUS PAPARAN RADIASI

Diagnosis_Status	Benign	Malignancy
Exposure_to_radiation		
no	82	504
yes	734	405



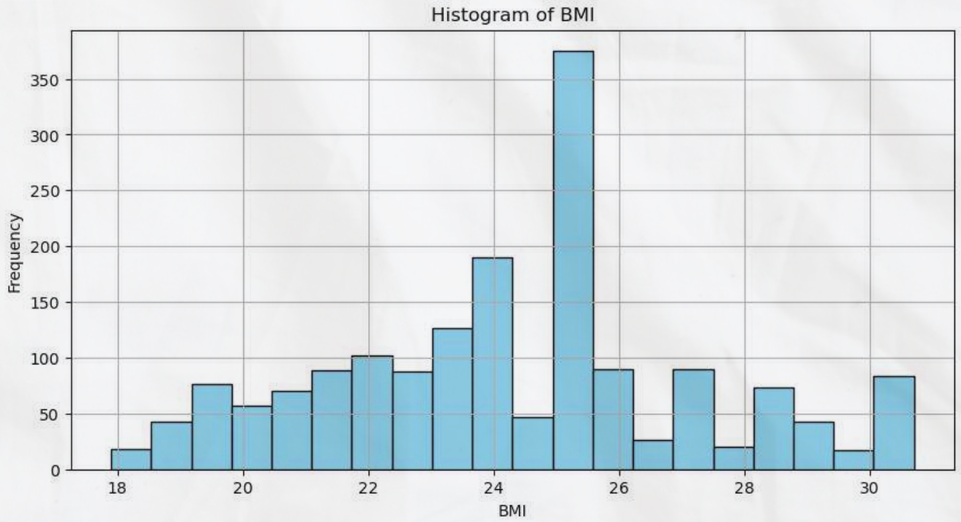
HISTOGRAM UMUR

```
# Menambahkan label dan judul
plt.figure(figsize=(10,10))
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.title('Histogram of Age')
plt.hist(df['Age'], bins=20, color='skyblue', edgecolor='black')
# Menampilkan histogram
plt.grid(True)
plt.show()
```



HISTOGRAM BMI

```
# Menambahkan label dan judul
plt.figure(figsize=(10,5))
plt.xlabel('BMI')
plt.ylabel('Frequency')
plt.title('Histogram of BMI')
plt.hist(df['BMI'], bins=20, color='skyblue', edgecolor='black')
# Menampilkan histogram
plt.grid(True)
plt.show()
```



HEATMAP


Correlation Heatmap
Age 1.00 0.14 -0.16 -0.07 -0.39 0.04 0.10 0.05 -0.05 -0.04 0.00
Alcohol_Intake 0.14 1.00 -0.35 -0.13 -0.10 -0.10 -0.01 -0.09 -0.05 -0.16 0.06
Smoking_Status -0.16 -0.35 1.00 0.35 -0.04 0.18 0.00 0.10 -0.08 0.25 -0.13
Family_history_of_breast_cancer -0.07 -0.13 0.35 1.00 -0.09 0.14 -0.05 0.07 -0.10 0.13 -0.01
Menopausal_Status -0.39 -0.10 -0.04 -0.09 1.00 -0.06 0.03 -0.02 0.20 -0.13 -0.04
Hormone_replacement_therapy_use 0.04 -0.10 0.18 0.14 -0.06 1.00 0.64 0.05 -0.08 0.07 0.04
Oral_contraceptive_use 0.10 -0.01 0.00 -0.05 0.03 0.64 1.00 0.02 -0.10 -0.04 0.06
BMI 0.05 -0.09 0.10 0.07 -0.02 0.05 0.02 1.00 -0.48 0.01 -0.08
Obesity -0.05 -0.05 -0.08 -0.10 0.20 -0.08 -0.10 -0.48 1.00 0.01 -0.13
Exposure_to_radiation -0.04 -0.16 0.25 0.13 -0.13 0.07 -0.04 0.01 0.01 1.00 -0.48
Diagnosis_Status 0.00 0.06 -0.13 -0.01 -0.04 0.04 0.06 -0.08 -0.13 -0.48 1.00
Age
Alcohol_Intake
Smoking_Status
Family_history_of_breast_cancer
Menopausal_Status
Hormone_replacement_therapy_use
Oral_contraceptive_use
BMI
Obesity
Exposure_to_radiation
Diagnosis_Status
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4

Data Splitting

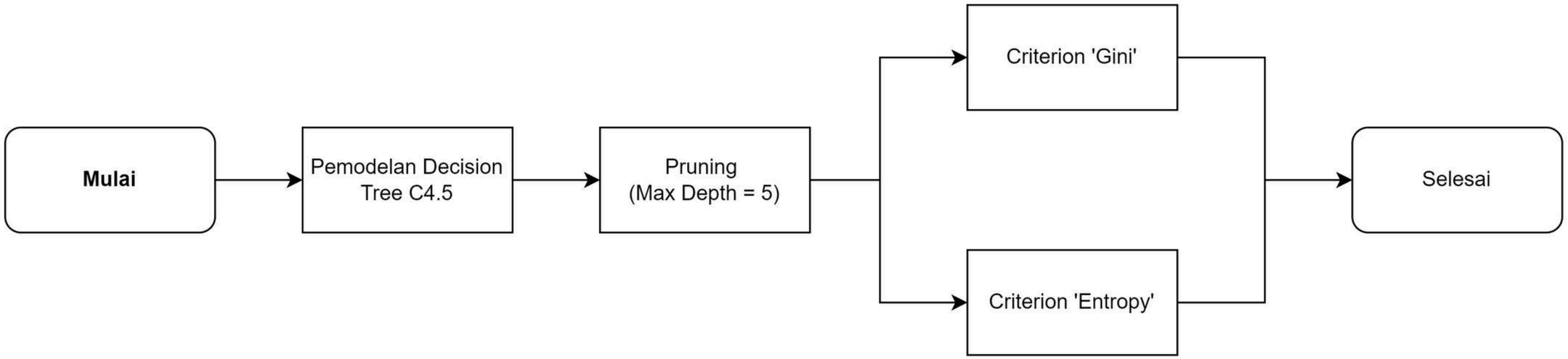
```
X = data.drop(['Diagnosis_Status'], axis=1)
y = data['Diagnosis_Status']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=51)
print(X_train.shape, X_test.shape)
```

```
(1207, 10) (518, 10)
```

- Kolom ‘Diagnosis_Status’ dihapus dari `df`.
- Data yang tersisa disimpan dalam `X`, dan variabel target dalam `y`.
- Data dipisahkan menjadi set pelatihan (X_train, y_train) dan pengujian (X_test, y_test) menggunakan `train_test_split` (30% untuk pengujian, `random_state` 51).
- Bentuk set pelatihan adalah (1207, 10) dan set pengujian adalah (518, 10).

Data Modeling

Flow Tahap Modeling



Dasar Perhitungan

Smoking_Status	Exposure_to_radiation	Diagnosis_Status
FALSE	FALSE	Benign
TRUE	TRUE	Malign
FALSE	FALSE	Benign
TRUE	TRUE	Benign
FALSE	FALSE	Benign
TRUE	TRUE	Malign
FALSE	FALSE	Malign
FALSE	FALSE	Malign
FALSE	TRUE	Malign
FALSE	FALSE	Malign

Perhitungan Entropi

$$Entropy(S) = -\Sigma_{i=1}^{n} pi\ log2(pi)$$

Rumus Entropy Keseluruhan

Rumus Gain

$$Gain(A) = E(S) - \Sigma v \frac{|Sv|}{|S|} \cdot E(Sv)$$

Perhitungan Gini

$$Gini = 1 - \sum_{i=1}^{n} (pi)^2$$

Rumus Gini

Rumus Gini Index

$$Gini\ Index = \sum_{v \in \{T,F\}} \frac{|Sv|}{|S|} Gini(Sv)$$

Evaluasi Perhitungan

Perbandingan Hasil Perhitungan Gain

Gain ‘Smoking_Status’	0,05
Gain ‘Exposure_to_Radiation’	0,07

Evaluasi Perhitungan

Perbandingan Hasil Perhitungan Gini Index

Gini Index ‘Smoking_Status’	0,495
Gini Index ‘Exposure_to_Radiation’	0,45

Proses Pemodelan

```
clf_c45_gini = DecisionTreeClassifier (max_depth=5, criterion='gini')
clf_c45_gini.fit(X_train, y_train)
y_pred_gini = clf_c45_gini.predict(X_test)
accuracy = accuracy_score(y_test, y_pred_gini)

print("Accuracy on training set: {:.3f}".format(clf_c45_gini.score(X_train, y_train)))
print("Accuracy on test set: {:.3f}".format(clf_c45_gini.score(X_test, y_test)))
print("Accuracy:", accuracy)
```

```
Accuracy on training set: 0.862
Accuracy on test set: 0.834
Accuracy: 0.833976833976834
```

Akurasi menggunakan Criteration ‘Gini’.

Proses Pemodelan

```
clf_c45_entropy = DecisionTreeClassifier (max_depth=5, criterion='entropy')
clf_c45_entropy.fit(X_train, y_train)
y_pred_entropy = clf_c45_entropy.predict(X_test)
accuracy = accuracy_score(y_test, y_pred_entropy)

print("Accuracy on training set: {:.3f}".format(clf_c45_entropy.score(X_train, y_train)))
print("Accuracy on test set: {:.3f}".format(clf_c45_entropy.score(X_test, y_test)))
print("Accuracy:", accuracy)
```

```
Accuracy on training set: 0.860
Accuracy on test set: 0.832
Accuracy: 0.832046332046332
```

Akurasi menggunakan Criteration ‘Entropy’.

Kode Entropy Information Gain

```python
import numpy as np

def entropy(y):
    """Calculate the entropy of a target variable"""
    unique_classes, class_counts = np.unique(y, return_counts=True)
    probs = class_counts / len(y)
    entropy = -np.sum(probs * np.log2(probs))
    return entropy

def information_gain(X, y, feature_idx):
    """Calculate the information gain for a specific feature"""
    parent_entropy = entropy(y)
    ""
    unique_values, value_counts = np.unique(X[:, feature_idx], return_counts=True)

    children_entropy = 0
    for value, count in zip(unique_values, value_counts):
        child_entropy = entropy(y[X[:, feature_idx] == value])
        children_entropy += (count / len(y)) * child_entropy

    information_gain = parent_entropy - children_entropy
    return information_gain

target_entropy = entropy(df['Diagnosis_Status'].values)
print("Entropy of target variable (Diagnosis_Status):", target_entropy)

features = df.drop(columns=['Diagnosis_Status']).values
for i, feature in enumerate(df.columns[:-1]):
    gain = information_gain(features, df['Diagnosis_Status'].values, i)
    print("Information Gain for feature '{}': {:.4f}".format(feature, gain))
```

Output Entropy Information Gain

```
Entropy of target variable (Diagnosis_Status): 0.9979023026797438
Information Gain for feature 'Age': 0.2794
Information Gain for feature 'Alcohol_Intake': 0.0030
Information Gain for feature 'Smoking_Status': 0.0131
Information Gain for feature 'Family_history_of_breast_cancer': 0.0000
Information Gain for feature 'Menopausal_Status': 0.0040
Information Gain for feature 'Hormone_replacement_therapy_use': 0.0011
Information Gain for feature 'Oral_contraceptive_use': 0.0029
Information Gain for feature 'Breast_Swelling': 0.0075
Information Gain for feature 'Breast_Lump': 0.0002
Information Gain for feature 'Breast_Pain': 0.0037
Information Gain for feature 'BMI': 0.5436
Information Gain for feature 'Obesity': 0.0128
Information Gain for feature 'Exposure_to_radiation': 0.1795
Information Gain for feature 'Breast_Feeding': 0.0005
```

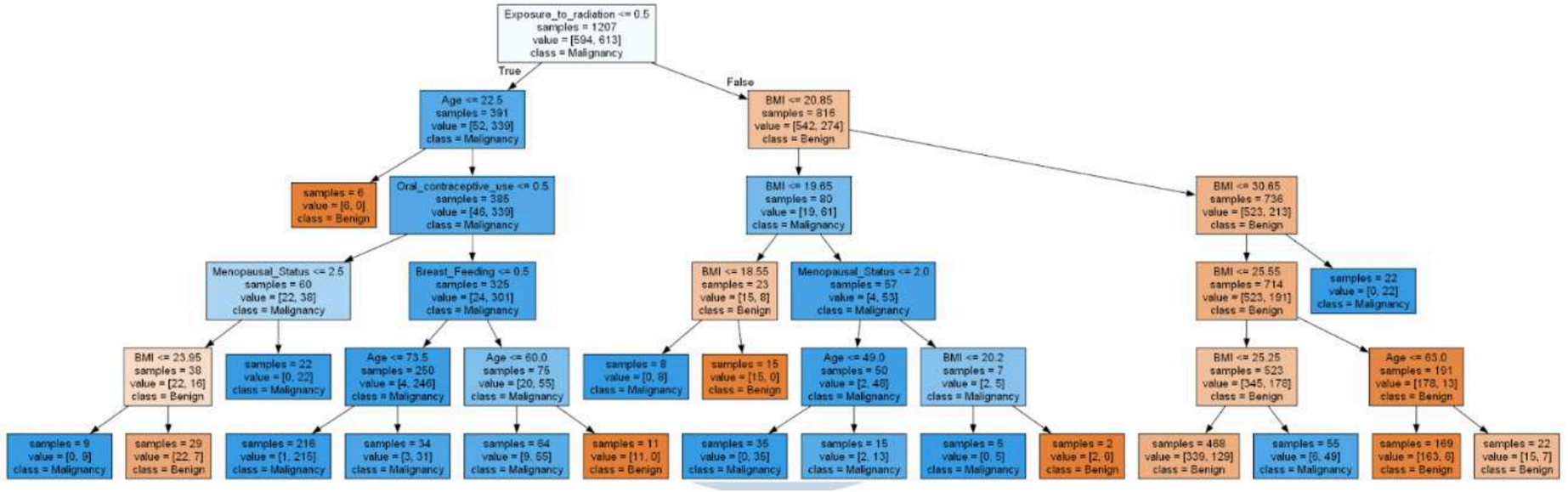
Kode Gini Index Information Gain

```python
import numpy as np

def gini(y):
    """Calculate the Gini index of a target variable"""
    unique_classes, class_counts = np.unique(y, return_counts=True)
    probs = class_counts / len(y)
    gini = 1 - np.sum(probs ** 2)
    return gini

def information_gain_gini(X, y, feature_idx):
    """Calculate the information gain for a specific feature using Gini index"""
    parent_gini = gini(y)

    unique_values, value_counts = np.unique(X[:, feature_idx], return_counts=True)

    children_gini = 0
    for value, count in zip(unique_values, value_counts):
        child_gini = gini(y[X[:, feature_idx] == value])
        children_gini += (count / len(y)) * child_gini

    information_gain_gini = parent_gini - children_gini
    return information_gain_gini

import pandas as pd

y = df['Diagnosis_Status'].values
X = df.drop(columns=['Diagnosis_Status']).values

target_gini = gini(y)
print("Gini index of target variable (Diagnosis_Status):", target_gini)

for i, feature in enumerate(df.columns[:-1]):
    gain_gini = information_gain_gini(X, y, i)
    print("Information Gain for feature '{}' (gini): {:.4f}".format(feature, gain_gini))
```

Output Gini Index Information Gain

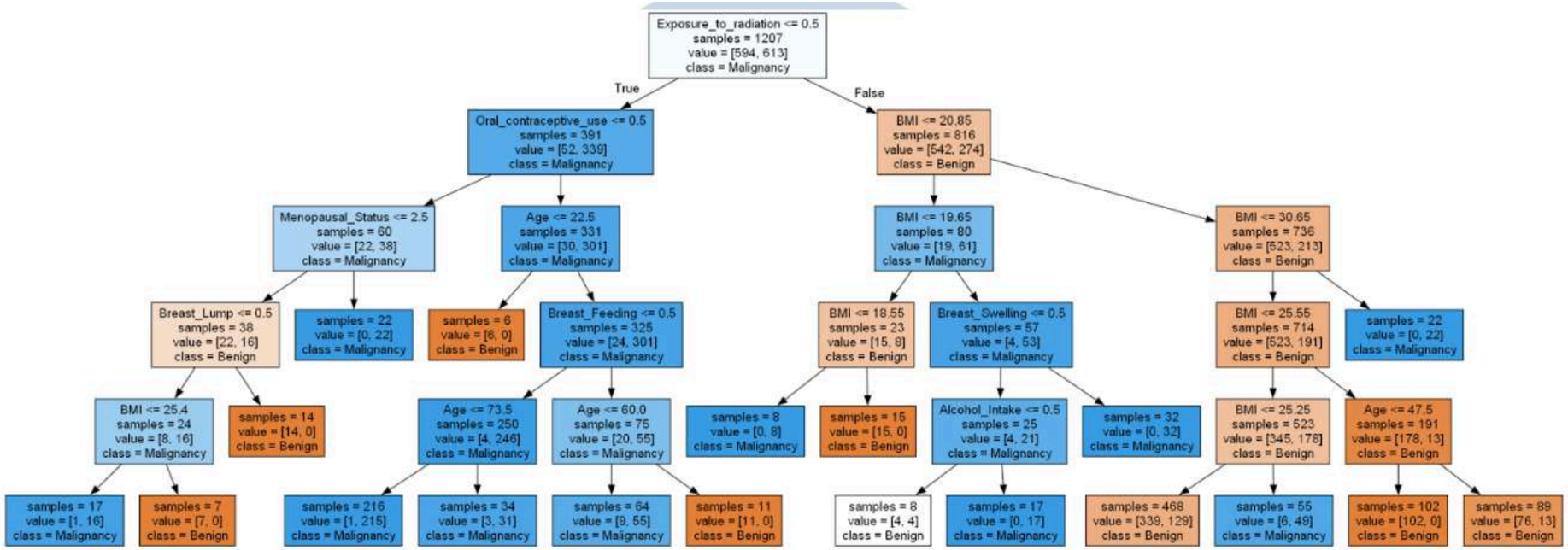
```
Gini index of target variable (Diagnosis_Status): 0.4985466918714556
Information Gain for feature 'Age' (gini): 0.1530
Information Gain for feature 'Alcohol_Intake' (gini): 0.0020
Information Gain for feature 'Smoking_Status' (gini): 0.0087
Information Gain for feature 'Family_history_of_breast_cancer' (gini): 0.0000
Information Gain for feature 'Menopausal_Status' (gini): 0.0028
Information Gain for feature 'Hormone_replacement_therapy_use' (gini): 0.0007
Information Gain for feature 'Oral_contraceptive_use' (gini): 0.0020
Information Gain for feature 'Breast_Swelling' (gini): 0.0051
Information Gain for feature 'Breast_Lump' (gini): 0.0002
Information Gain for feature 'Breast_Pain' (gini): 0.0025
Information Gain for feature 'BMI' (gini): 0.2852
Information Gain for feature 'Obesity' (gini): 0.0082
Information Gain for feature 'Exposure_to_radiation' (gini): 0.1142
Information Gain for feature 'Breast_Feeding' (gini): 0.0004
```

Visualisasi Decision Tree



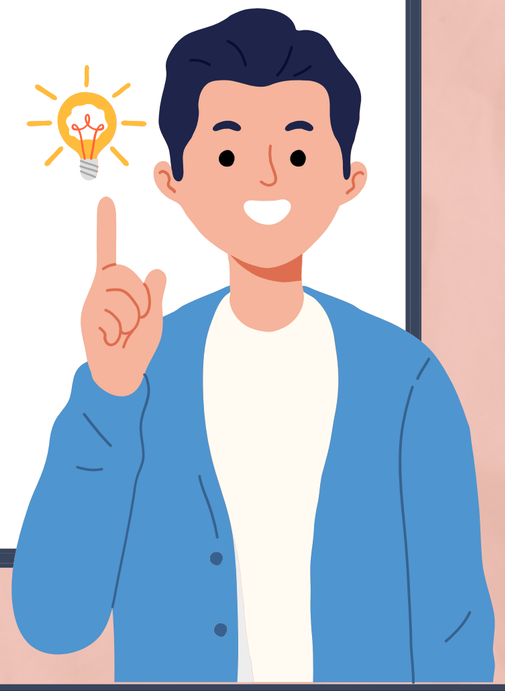
Decision Tree Criterion ‘Gini’

Visualisasi Decision Tree



Decision Tree Criterion ‘Entropy’

Hasil Validasi dan Evaluasi Model



Validasi Model

```python
from sklearn.model_selection import cross_val_score
from sklearn.tree import DecisionTreeClassifier
import numpy as np
import pandas as pd

X = data.drop(columns=['Diagnosis_Status'])
y = data['Diagnosis_Status']

clf_c45_gini = DecisionTreeClassifier(max_depth=5, criterion='gini')
cv_scores_gini = cross_val_score(clf_c45_gini, X, y, cv=5)
mean_cv_score_gini = np.mean(cv_scores_gini)

print("Mean Cross-Validation Score (Gini):", mean_cv_score_gini)

clf_c45_entropy = DecisionTreeClassifier(max_depth=5, criterion='entropy')
cv_scores_entropy = cross_val_score(clf_c45_entropy, X, y, cv=5)
mean_cv_score_entropy = np.mean(cv_scores_entropy)

print("Mean Cross-Validation Score (Entropy):", mean_cv_score_entropy)
```

```
Mean Cross-Validation Score (Gini): 0.8469565217391304
Mean Cross-Validation Score (Entropy): 0.8400000000000001
```

Hasil menunjukkan skor rata-rata cross-validation dari model Decision Tree dengan 5 fold. Model ini mencetak rata-rata 0.846 untuk kriteria Gini dan 0.840 untuk kriteria Entropy, menunjukkan bahwa kriteria Gini sedikit lebih unggul pada dataset ini.

3

	Predicted Benign	Predicted Malignancy		
Actual Benign	224	72	Positive Predictive Value	0.76
Actual Malignancy	14	208	Negative Predictive Value	0.93
	Sensitivity	Specificity	Accuracy = 0.83	
	0.94	0.74		

Evaluasi Perhitungan

$$Accuracy = \frac{224 + 208}{224 + 208 + 72 + 14}$$

$$Accuracy = 0.83$$

$$Precision = \frac{224}{224 + 72}$$

$$Precision = 0.75$$

$$Recall = \frac{224}{224 + 14}$$

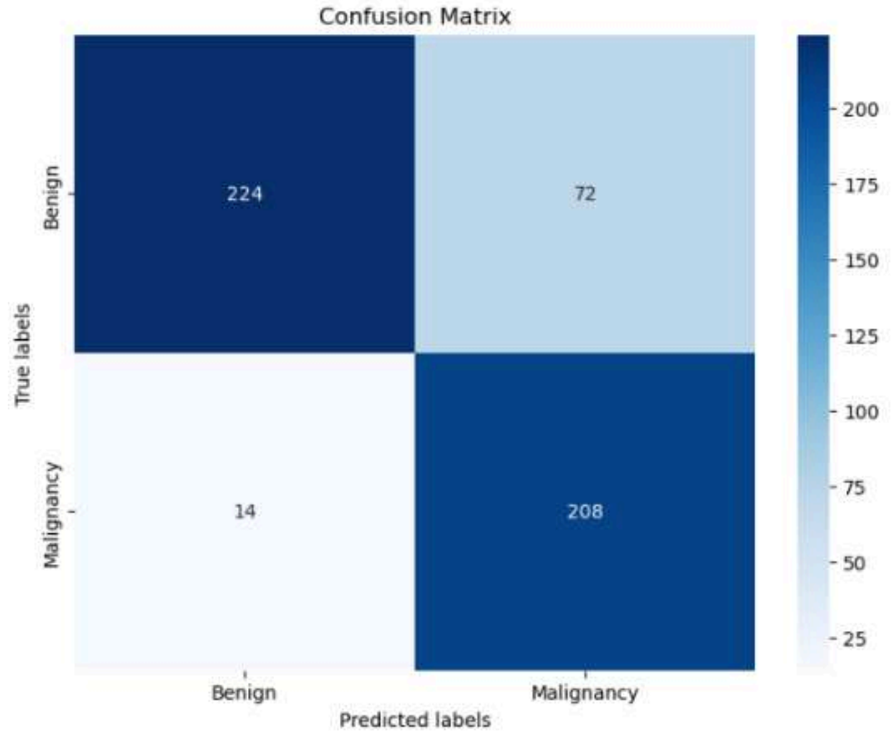
$$Recall = 0.94$$

$$Specificity = \frac{TN}{TN + FP}$$

$$Specificity = \frac{208}{208 + 72}$$

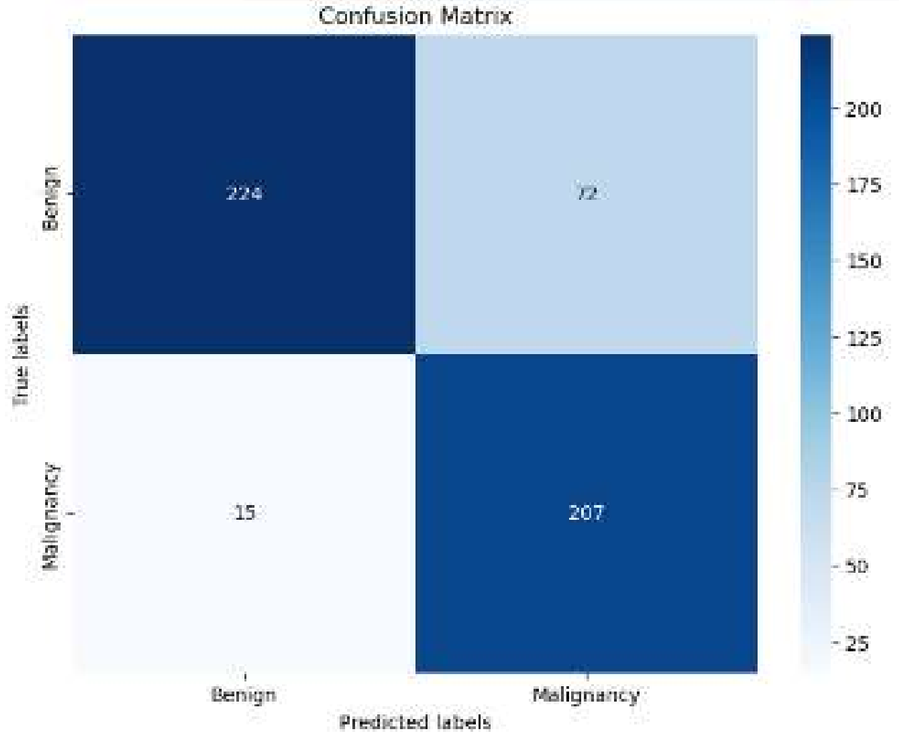
$$Specificity = 0.74$$

Evaluasi Model



Grafik menunjukkan confusion matriks dari model dengan kriteria 'gini'. Terdapat 224 true positives (benign), 14 false negatives (malign), 72 false positives (benign), dan 208 true negatives (malign). Ini menggambarkan kemampuan model dalam memprediksi klasifikasi "benign" dan "malign".

Evaluasi Model



Ilustrasi matriks kebingungan dari model dengan kriteria 'Entropy' menunjukkan 224 true positives (benign), 15 false negatives (malign), 72 false positives (benign), dan 207 true negatives (malign). Ini mengindikasikan kemampuan model dalam memprediksi klasifikasi "benign" dan "malign".

Evaluasi Model

```
from sklearn.metrics import f1_score, precision_score, recall_score

precision1 = precision_score(y_test, y_pred_gini)
recall1 = recall_score(y_test, y_pred_gini)
f1_gini = f1_score(y_test, y_pred_gini)

print("Precision:", precision1)
print("Recall:", recall1)
print("F1 Score:", f1_gini)
```

```
Precision: 0.7428571428571429
Recall: 0.9369369369369369
F1 Score: 0.8286852589641435
```

f1 Score dengan Criterion ‘Gini’

Evaluasi Model

```
from sklearn.metrics import f1_score, precision_score, recall_score

precision2 = precision_score(y_test, y_pred_entropy)
recall2 = recall_score(y_test, y_pred_entropy)
f1_entropy = f1_score(y_test, y_pred_entropy)

print("Precision:", precision2)
print("Recall:", recall2)
print("F1 Score:", f1_entropy)
```

```
Precision: 0.7419354838709677
Recall: 0.9324324324324325
F1 Score: 0.8263473053892216
```

f1 Score dengan Criterion ‘Entropy’

Features Importances

```
print ("Features importances:\n{}". format(clf_c45_gini.feature_importances_))
```

```
Features importances:
[0.0824946  0.         0.         0.         0.         0.
 0.02466883 0.46911002 0.         0.42372654]
```

Criterion
‘Gini’

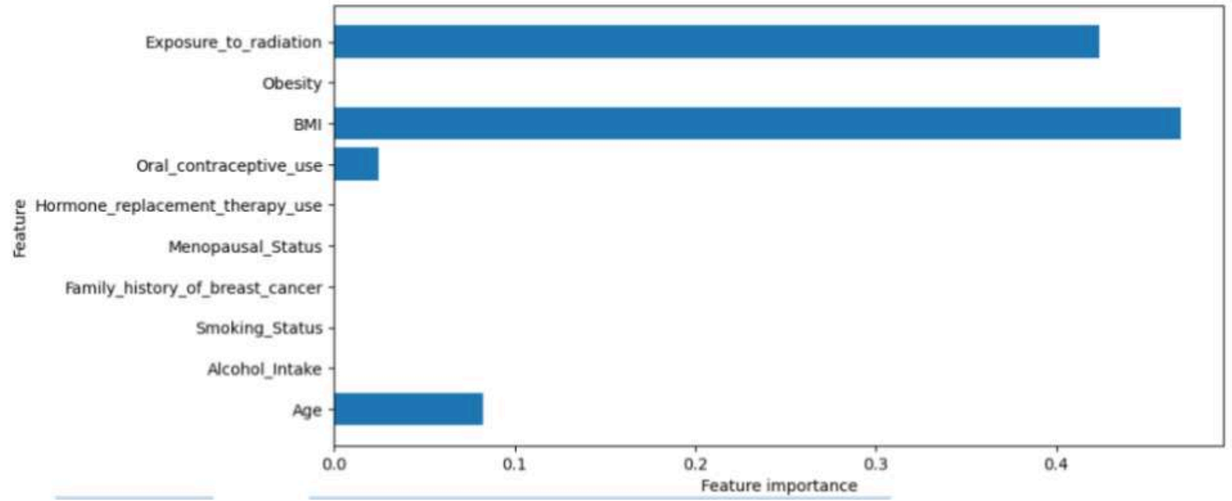
Criterion
‘Entropy’

```
print ("Features importances:\n{}". format(clf_c45_entropy.feature_importances_))
```

```
Features importances:
[0.13313924 0.         0.         0.         0.         0.
 0.0294961  0.4752849  0.         0.36207975]
```

Visualisasi features importances

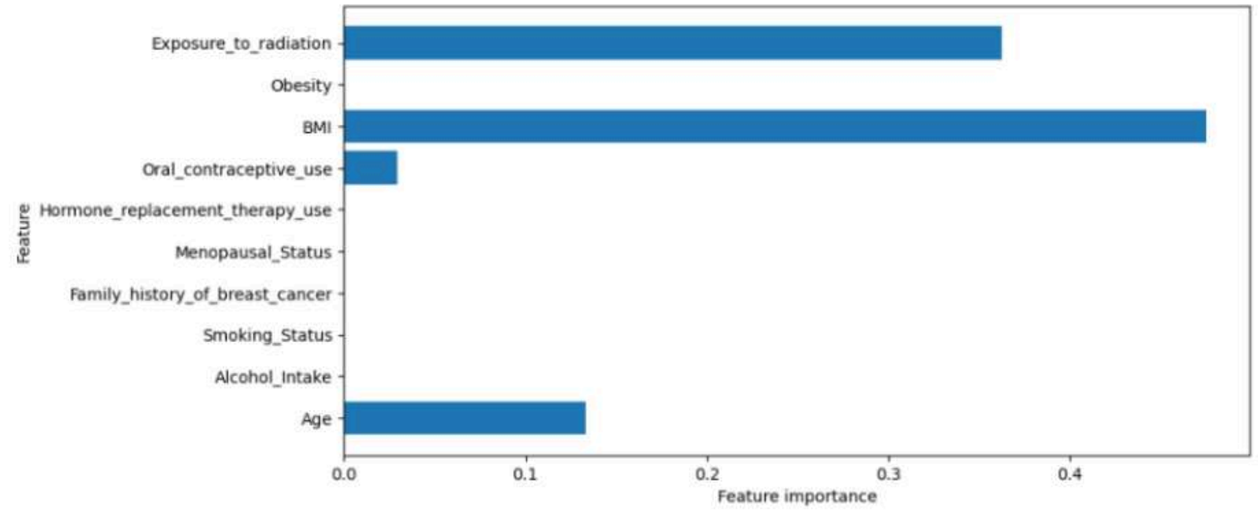
```
def plot_feature_importances(model):
    n_features = len(X.columns)
    plt.barh(range(n_features), model.feature_importances_, align='center')
    plt.yticks(np.arange(n_features), X.columns)
    plt.xlabel("Feature importance")
    plt.ylabel("Feature")

plot_feature_importances(clf_c45_gini)
```



Criterion ‘Gini’

Visualisasi features importances

```
def plot_feature_importances(model):
    n_features = len(X.columns)
    plt.barh(range(n_features), model.feature_importances_, align='center')
    plt.yticks(np.arange(n_features), X.columns)
    plt.xlabel("Feature importance")
    plt.ylabel("Feature")

plot_feature_importances(clf_c45_entropy)
```



Criterion ‘Entropy’

Kesimpulan

Penelitian menunjukkan bahwa Indeks Massa Tubuh (BMI), paparan terhadap faktor risiko, dan usia sangat penting dalam identifikasi jenis kanker payudara. Penggunaan algoritma Decision Tree C4.5 dengan Gini Index sebagai kriteria pemilihan fitur terbukti efektif, dengan akurasi 83% dan skor F1 sebesar 0.828. Algoritma ini unggul dalam membedakan tumor "benign" dan "malign", memberikan klasifikasi intuitif dan membantu dalam diagnosis serta perawatan kanker payudara.

Saran

1. Memperhitungkan inklusi faktor-faktor lain dalam analisis risiko kanker payudara;
2. Melakukan uji validasi eksternal pada model yang dikembangkan dengan menggunakan dataset yang berbeda untuk mengkonfirmasi keakuratannya secara independen;
3. Mengintegrasikan data dari berbagai sumber seperti data genetik, citra medis, dan data klinis untuk mendapatkan pemahaman yang lebih komprehensif tentang kanker payudara

Laporan_Kelompok_1_UAS IF540_TAGenap20232024

ORIGINALITY REPORT

15%	11%	5%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Academic Library Consortium Student Paper	2%
2	etheses.uin-malang.ac.id Internet Source	1%
3	journal.irpi.or.id Internet Source	1%
4	ejournals.umn.ac.id Internet Source	1%
5	eprints.ums.ac.id Internet Source	1%
6	ejournal.itn.ac.id Internet Source	<1%
7	kc.umn.ac.id Internet Source	<1%
8	Submitted to Sriwijaya University Student Paper	<1%
9	docplayer.info Internet Source	<1%

Lampiran

Thank You