# Classification of Breast Cancer Diagnosis Using the Instance Based Learning Method

Kenny Budiarso (00000081065)0[1], Nabila Az Zahra (00000081399)[2], Rafi Aldino(00000081108)[3]
Department of Information Systems, Multimedia Nusantara University, Indonesia
kenny.budiarso@student.umn.ac.id
nabil.az@student.umn.ac.id
rafi.aldino@student.umn.ac.id

***Abstract*—Classification is the main method operated in data mining. This technique is often applied to many data sets to solve a problem in a study. This classification itself is a method of separating and grouping data that involves training data using a classification algorithm. The aim of this study is to predict whether the tumor is benign or malignant based on its characteristics. This could be of great assistance to medical research, especially in identifying cancer-causing tumors.**

***Keywords* – Classification; Algorithm; Tumor; Benign; Malignant; Data; KNN; Decision Tree**

## I. INTRODUCTION

Teknologi informasi merupakan sebuah aspek penting saat ini yang nantinya akan terus mengalami kemajuan dan perkembangan dari masa ke masa. Teknologi informasi sendiri sudah memberikan berbagai jenis informasi serta data-data yang bermanfaat bagi kehidupan masyarakat pada umumnya, yang mana data data tersebut sangat berdampak baik bagi penyuluhan dan penyebaran informasi yang didapatkan oleh masyarakat. Informasi yang dihasilkan telah mencakup seluruh bidang kehidupan manusia mulai dari ekonomi, politik, sejarah, sosial, kesehatan dan sebagainya. Teknologi juga memberikan peran dalam kemajuan berbagai institusi, salah satunya instansi kesehatan. Data yang dihasilkan di bidang kesehatan bisa berupa data mengenai penyakit yang dianggap mematikan seperti kanker, yang pasti tentu saja data tersebut akan dapat dimanfaatkan untuk menggaliberbagai jenis informasi lebih dalam terkait penyakit kanker itu sendiri, baik untuk pengobatan ataupun pencegahan terhadap pasien yang belum dan atau sudah mengalami penyakit kanker. Kanker adalah sekelompok besar penyakit yang dapat muncul di hampir semua organ atau jaringan tubuh ketika sel-sel yang tidak normal tumbuh tak terkendali. Kanker merupakan penyebab kematian terbanyak di dunia.

Setiap tahun, 12 juta orang di dunia menderita kanker dan 7,6 juta diantaranya meninggal. Pada tahun 2013 di Indonesia prevalensi tumor/kanker di Indonesia adalah 1,4 per 1000 penduduk. Berdasarkan hasil Riskesdas 2013 bahwa kasus penyakit kanker di Indonesia terbanyak ada di Provinsi Jawa Tengah, dengan prevalensi 2,1 per seribu penduduk. Jika penduduk Jawa Tengah tahun 2017 sejumlah 35 juta jiwa, berarti ada sekitar 70.000 penduduk yang menderita penyakit kanker. Jenis kanker biasanya diberi nama sesuai organ atau jaringan tempat kanker itu terbentuk, salah satunya adalah kanker payudara.

Kanker payudara merupakan penyebab kematian tertinggi kanker pada perempuan di Indonesia. Hal ini disebabkan penderita kanker payudara pergi ke pelayanan kesehatan saat kanker payudara sudah stadium lanjut. Keterlambatan penanganan ini disebabkan kurangnya pengetahuan masyarakat tentang kanker payudara dan belum tahunya cara periksa payudara sendiri (SADARI) untuk deteksi dini kanker payudara. Dikutip dari situs resmi CNN pada tahun 2018, jumlah penderita kanker di seluruh dunia terus meningkat secara signifikan. Laporan terbaru yang dirilis oleh International Agency for Research on Cancer, WHO (Organisasi Kesehatan Dunia), mengestimasi terdapat 18,1 juta kasus kanker baru dan 9,6 juta kematian yang terjadi pada tahun 2018. Serangan kanker yang masif ini diprediksi oleh WHO akan menjadi penyebab kematian nomor satu di dunia pada akhir akhir ini. Hasil analisa tersebut didapat setelah peneliti dari WHO menganalisis data dari 185 negara di dunia dengan fokus pada 36 jenis kanker. Beberapa jenis kanker yang disebutkan seperti kanker paru, kolorektal, lambung, hati dan payudara. Organisasi kesehatan dunia (WHO) menyatakan bahwa insiden lima besar kanker di dunia salah satunya adalah kanker payudara. Amerika yang dikenal maju dalam segala hal, termasuk dari tingkat kemampuan menjaga kesehatan diri dan keluarga, ternyata kondisinya sangat memprihatinkan.

Pada kasus umumnya kanker payudara ditemukan pada stadium lanjut akibat kelalaian penderita dalam mendeteksi benjolan ataupun kelainan pada payudaranya. Padahal, kemungkinan sembuh tentu akan semakin besarv apabila benjolan kanker dapat terdeteksi lebih awal, yang mana hal tersebut membuat seseorng yang terkena penyakit kanker payudara yang terdeteksi lebih awal dapat melakukan pencegahan dan penanggulangan dengan lebih sigap dan lebih tepat agar tidak semakin membesar. Sebagai cara atau teknis untuk dapat memaksimalkan analisis kanker payudara maka dapat menggunakan methode atau taknis machine learing. Machine Learning atau yang biasa dikenal dengan sebutan (ML) adalah salah satu pengaplikasian dari Artificial Intilligent (AI) yang berfokus kepada pengembangan sebuah sistem yang mampu belajar sendiri tanpa harus diprogram berulang kali. ML membutuhkan sebuah data (data traning) sebagai proses learning sebelum menghasilkan sebuah hasil final.

| | |
|---|---|
| **Primary Tumor (T)** | |
| **T0** | No evidence of primary tumor |
| **Tis** | Carcinoma in situ |
| **T1, T2, T3, T4** | Increasing size and/or local extension of the primary tumor |
| **TX** | Primary tumor cannot be assessed (use of TX should be minimized) |
| **Regional Lymph Nodes (N)** | |
| **N0** | No regional lymph node metastases |
| **N1, N2, N3** | Increasing number or extend of regional lymph node involvement |
| **NX** | Regional lymph nodes cannot be assessed (use of NX should be minimized) |
| **Distant Metastasis (M)** | |
| **M0** | No distant metastases |
| **M1** | Distant metastases present |

*Sumber: American Join Committee on Cancer (2010)*

Berbagai jenis penelitian dalan bidang medis telah dilakukan untuk dapat melihat dan memprediksi sebuah penyakit. Dalam penelitian kali ini algotitma yang digunakan untuk memprediksi penyakit kangker payudara ialah metode KNN (K-Nearest Neighbor), dan metode SVM (Support Vektor Machine), yang mana dengan metode tersebut diharapkan dapat menciptakan sebuah inovasi baru untuk dapat menemukan atau melihat prediksi kanker payudara didalam tubuh seseorang.

Melalui berbagai jenis penelitian sebelumnya dapat dinyatakan bahwasannya seiring dengan bertambahnya usia seseorang, risiko untuk dapat terkena kanker payudara cenderung meningkat, terutama dengan faktor-faktor seperti riwayat keluarga atau mutasi genetik. Hormon, pola menstruasi, dan faktor reproduksi juga berpengaruh. Lingkungan dan gaya hidup, seperti alkohol, obesitas, dan radiasi, turut berkontribusi pada peningkatan risiko. Dengan itu pencegahan untuk terkena hal tersebut dapat di minimalisir dengan melakukan untuk melalui gaya hidup sehat, pemeriksaan payudara sendiri, dan pemeriksaan medis rutin, termasuk mamografi, sangat penting untuk deteksi dini dan pengelolaan kanker payudara. Dengan pemahaman lebih baik terhadap penyebabnya, diharapkan upaya pencegahan, deteksi, dan pengobatan dapat ditingkatkan, meningkatkan kualitas hidup dan harapan hidup pasien. Untuk itu perlukan upaya untuk mencegah meningkatnya risiko pertumbuhan tumor kanker panyudara di kalangan Wanita.

## II. LITERATURE

a. Machine Learning

Machine learning ialah salah satu pengaplikasian dari Artificial Intilligent (AI) yang berfokus kepada pengembangan sebuah sistem yang mampu belajar sendiri tanpa harus diprogram berulang kali. ML membutuhkan sebuah data (data traning)sebagai proses learning sebelum menghasilkan sebuah hasil. Berbagai jenis penelitian dalan bidang medis telah dilakukan untuk dapat melihat dan memprediksi sebuah penyakit.

b. Kanker Payudara

Kanker payudara atau Carsinoma Mammaeadalah kondisi ketika sel kanker terbentuk di jaringan payudara. Kanker bisa terbentuk di kelenjar yang menghasilkan susu (lobulus), atau di saluran (duktus) yang membawa air susu dari kelenjar ke puting payudara, kanker juga bisa terbentuk di jaringan lemak atau jaringan ikat di dalam payudara. Kanker payudara merupakan penyebab utama tingkat kematian kedua pada wanita.

c. KNN (K-Nearest Neighbor)

K-Nearest Neighbor (K-NN) termasuk kelompok instance-based learning. Algoritma ini juga merupakan salah satu teknik lazy learning. kNN dilakukan dengan mencari kelompok k objek dalam data training yang paling dekat (mirip) dengan objek pada data baru atau data testing. diperlukan suatu sistem klasifikasi sebagai sebuah sistem yang mampu mencari informasi.

d. SVM (Support Vector Machine)

Konsep SVM bertujuan untuk menemukan hyperplaneyang memisahkan himpunan datake dalam dua kelas secara linier (Suyanto, 2018:99). Hyperplan adalah istilah yang dibuatgeneral untuk semua dimensi. SVM berusaha untuk menemukan hyperplane yang paling optimum atau terbaik. Proses klasifikasi memiliki dua proses, yaitu (1) proses training dan (2) proses testing. Proses trainingdigunakan untuk membangunmodeldari suatu traningset. Algoritma SVM pada proses training dilakukan dengan delapan langkah penyelesaia.

## II. METHODOLOGY

### *A. Object of Research*

Data yang digunakan adalah data sekunder dari dataset Wisconsin Breast Cancer Diagnosis (WBCD). Dataset ini berisi 569 data pasien dengan 30 fitur, termasuk diagnosis kanker payudara (benign atau malignant).

### *B. Methods of Collecting Data*

Metode yang digunakan adalah metode IBL dengan algoritma k-Nearest Neighbors (KNN). Algoritma KNN akan mencari k data terdekat dari data baru dan kemudian menggunakan label dari k data tersebut untuk memprediksi label data baru.

### *C. Methods of Research*

Penelitian ini menggunakan metode CRISP–DM yang merupakan singkatan dari Cross – Industry Standard Process for Data Mining. Memiliki 6 tahapan yaitu Pemahaman Bisnis, Pemahaman Data, Persiapan Data, Pemodelan, Evaluasi, dan Penerapan [9]. Data ini dianalisis menggunakan algoritma K-Nearest Neighbor (KNN) dan SVM untuk melakukan prediksi dan menentukan pasien terindikasi penyakit kanker atau tidak. Gambar 1 menunjukkan kerangka operasional yang harus diikuti dalam penelitian ini. Kerangka penelitian akan digunakan untuk

mengimplementasikan langkah-langkah yang diambil selama penelitian. Hal ini digunakan sebagai pedoman bagi peneliti agar lebih fokus pada ruang lingkup penelitiannya.

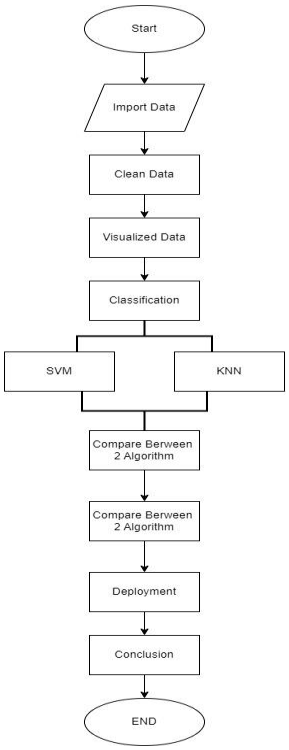


Fig. 1. The Research Framework

Gambar diatas merupakan gambaran metode penelitian yang kami lakukan. Awalnya kami membentuk tim. Setelah itu, kami mencari dataset yang ingin kami gunakan dalam penelitian ini. Setelah selesai memilih dataset dari website Kaggle, kita mendownload dataset tersebut. Kemudian, langkah kita selanjutnya adalah memasukkan dataset ke dalam jupyter notebook untuk dianalisis.
Setelah data diimport ke notebook jupyter, kami melakukan pembersihan pada data yang ada. Data yang tidak relevan dengan penelitian ini akan dihilangkan atau dibersihkan agar terlihat lebih baik dan lebih mudah dianalisis. Selanjutnya data divisualisasikan, dimana masing-masing

Langkah pengkodean telah dilakukan secara individual, dan data divisualisasikan dengan berbagai cara untuk mendukung skripsi.
Untuk mengklasifikasikan data, penulis menggunakan algoritma KNN dan SVM, setelah data tersebut divisualisasikan. Setelah itu, penulis membandingkan algoritma dan mencocokkan korelasinya dengan data setelah dikodekan. Kesimpulan setelah terbentuknya pencocokan model. Kesimpulan merupakan hasil akhir penelitian ini, yang diharapkan dapat menjadi jawaban yang tepat terhadap penelitian tersebut. Bagian terakhir adalah kesimpulan jurnal yang merupakan bagian penutup jurnal ini.

## III. RESULT AND DISCUSSION

### *A. Exploratory Data Analysis*

```
df = pd.read_csv('Tumor.csv')
df.head()
```

| | id | diagnosis | radius_mean | texture_mean | perimeter_mean | area_mean | s |
|---|---|---|---|---|---|---|---|
| 0 | 842302 | M | 17.99 | 10.38 | 122.80 | 1001.0 | |
| 1 | 842517 | M | 20.57 | 17.77 | 132.90 | 1326.0 | |
| 2 | 84300903 | M | 19.69 | 21.25 | 130.00 | 1203.0 | |
| 3 | 84348301 | M | 11.42 | 20.38 | 77.58 | 386.1 | |
| 4 | 84358402 | M | 20.29 | 14.34 | 135.10 | 1297.0 | |

Fig. 2. Import Data

The image above is the image of the output result from importing data, where the first thing done is calling the package that will be used during the entire analysis activity, then after calling all the available packages, inserting the data in the form of a . CSV into the jupyter notebook. After the data is called, the next step is to display the data using the .head() code.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 33 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   id                       569 non-null    int64
 1   diagnosis                569 non-null    object
 2   radius_mean              569 non-null    float64
 3   texture_mean             569 non-null    float64
 4   perimeter_mean           569 non-null    float64
```

Fig. 3. Data Information

Fig. 3. is the output result of the data.info, where this image will display the name of each column in the data. In addition, it will also display the number of data in each column. Then, it can also display the data type of each variable and it can be seen that there are 31 float data types, 1 int data type, and 1 object data type.

```
df.describe()
```

| | id | radius_mean | texture_mean | p |
|---|---|---|---|---|
| count | 5.690000e+02 | 569.000000 | 569.000000 | |
| mean | 3.037183e+07 | 14.127292 | 19.289649 | |
| std | 1.250206e+08 | 3.524049 | 4.301036 | |
| min | 8.670000e+03 | 6.981000 | 9.710000 | |
| 25% | 8.692180e+05 | 11.700000 | 16.170000 | |
| 50% | 9.060240e+05 | 13.370000 | 18.840000 | |
| 75% | 8.813129e+06 | 15.780000 | 21.800000 | |
| max | 9.113205e+08 | 28.110000 | 39.280000 | |

Fig. 4. Data Describe

The image above is data that displays the entire description of the data, where it displays the count, mean, std, min, 25%, 50%, 75%, and a max of the analyzed data.

```
df.shape
```

```
(569, 33)
```

Fig. 5. Data Shape

The image above is the output of the data shape, which will display information about the data that has been analyzed.

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

```
id                0
diagnosis         0
radius_mean       0
texture_mean      0
perimeter_mean    0
area_mean         0
```

Fig. 6. Find Missing Value

The image above is the output of the result of searching for missing values. Missing values are used to find out if there is data with a null value or not, the purpose is to make the data more accurate when analyzed. Therefore, from the above output, it can be concluded that there are no null values or empty values in the data.

```
df = df.drop(columns=['Unnamed: 32', 'radius_mean', 'texture_mean', 'perimeter_mean', 'area_mean',
                      'smoothness_mean', 'compactness_mean', 'concavity_mean', 'symmetry_mean', 'concave points_mean',
                      'fractal_dimension_mean', 'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se',
                      'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se', 'fractal_dimension_se'])
df.head()
```

| | id | diagnosis | radius_worst | texture_worst | perimeter_worst | area_worst | smoothness_worst | compactness_worst | concavity_worst | c |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 842302 | M | 25.38 | 17.33 | 184.60 | 2019.0 | 0.1622 | 0.6656 | 0.7119 | |
| 1 | 842517 | M | 24.99 | 23.41 | 158.80 | 1956.0 | 0.1238 | 0.1866 | 0.2416 | |
| 2 | 84300903 | M | 23.57 | 25.53 | 152.50 | 1709.0 | 0.1444 | 0.4245 | 0.4504 | |
| 3 | 84348301 | M | 14.91 | 26.50 | 98.87 | 567.7 | 0.2098 | 0.8663 | 0.6869 | |
| 4 | 84358402 | M | 22.54 | 16.67 | 152.20 | 1575.0 | 0.1374 | 0.2050 | 0.4000 | |

Fig. 7. Drop Unused Columns

The picture above shows the code that was done to delete some columns in data that were deemed unnecessary. So in the end there will be 12 columns left that will be used.

*B. Visualization Data*

```
# Convert 'diagnosis' column to categorical data type
df['diagnosis'] = df['diagnosis'].astype('category')

# Data Visualization
sns.countplot(data=df, x='diagnosis')
```

```
<AxesSubplot: xlabel='diagnosis', ylabel='count'>
```

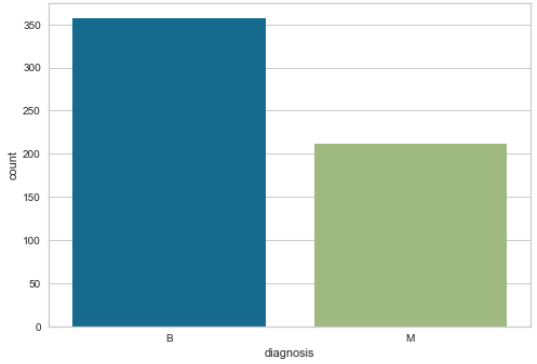


Fig. 8. Diagnosis Chart

The image above is a bar chart showing data on tumor diagnosis where there are two types of tumors which M stands for Malignant and B stands for Benign.

This visualization is made by using sns.countplot() which is a function from the Python library seaborn that is used to visualize the frequency or count of categories in a categorical data set. The first argument to the function, df['diagnosis'], is a column in a pandas DataFrame that contains categorical data. The function will plot a bar chart where the x-axis represents the categories, and the y-axis represents the count of each category.

```
M = df[df.diagnosis == "M"]
B = df[df.diagnosis == "B"]

# Scatter Plot
plt.scatter(M.radius_worst,M.texture_worst,color="red",label="malignant",alpha= 0.3)
plt.scatter(B.radius_worst,B.texture_worst,color="gold",label="benign",alpha= 0.3)
plt.xlabel("radius_worst")
plt.ylabel("texture_worst")
plt.legend()
plt.show()
#The label argument specifies the label for this data series, which will be used in the legend.
#The alpha argument specifies the transparency of the points.
#The legend function is used to add a legend to the plot, and the show function is used to display the plot.

df.diagnosis = [1 if each == "M" else 0 for each in df.diagnosis]
y = df.diagnosis.values
x_df = df.drop(["diagnosis"],axis=1)
#The code then modifies the diagnosis column of the df DataFrame so that it contains 1 for malignant tumors and 0 for benign tumors.
#The y variable is created to store the modified diagnosis column as an array.
#The x_df variable is created to store the df DataFrame with the diagnosis column removed.
```

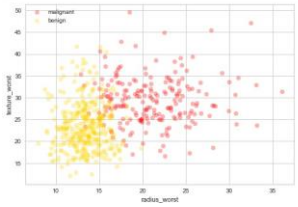


Fig. 9. Scatter Plot

This scatter plot is created using the matplotlib library. The scatter function is used to plot the data. The first call to scatter plots the radius_mean and texture_mean columns of the M DataFrame (which contain data for the malignant tumors) and uses the color argument to specify that the points should be plotted in red. The second call to scatter plots the radius_mean and texture_mean columns of the B DataFrame (which contain data for the benign tumors) and uses the color argument to specify that the points should be plotted in gold. The xlabel and ylabel functions are used to specify the labels for the x-axis and y-axis, respectively.

*C. Encoding*

```
def encode_data(feature_name):
    mapping_dict = {}
    unique_values = list(df[feature_name].unique())
    for idx in range(len(unique_values)):
        mapping_dict[unique_values[idx]] = idx
    return mapping_dict

df['diagnosis'].replace(encode_data('diagnosis'),inplace=True)
```

```
X = df[['radius_worst','texture_worst','perimeter_worst','area_worst','smoothness_worst','compactness_worst','concavity_worst',
        'concave points_worst','symmetry_worst','fractal_dimension_worst']]
y = df['diagnosis']

# Normalized X Using MinMaxScaler
mmscal = MinMaxScaler()
X_scaled = mmscal.fit_transform(X)
```

```
y.head()
```

```
0    0
1    0
2    0
3    0
4    0
Name: diagnosis, dtype: int64
```

Fig. 10. Encoding Diagnosis Benign Column

The picture above is created by applying the get_dummies function on the “df” dataframe. The get_dummies function is a useful way to convert categorical variables, such as the diagnosis column in this case, into dummy/indicator variables. Diagnosis column has the values “Benign" and "Malignant", the resulting dataframe will have two new columns, "diagnosis_Benign" and "diagnosis_Malignant", with values of 1 or 0 depending on the value in the original diagnosis column.

*D. Correlation*

```
# Melihat korelasi antar kolom
corr = df.corr().abs()
sol = (corr.where(np.triu(np.ones(corr.shape), k=3).astype(bool)).stack().sort_values(ascending=False))

print("CORRELATION COLUMN")
print(sol[sol > 0.01])
```

```
CORRELATION COLUMN
radius_worst       area_worst                0.984015
perimeter_worst    concave points_worst      0.816322
compactness_worst  fractal_dimension_worst   0.810455
diagnosis          concave points_worst      0.793566
radius_worst       concave points_worst      0.787424
diagnosis          perimeter_worst           0.782914
area_worst         concave points_worst      0.747419
diagnosis          area_worst                0.733825
concavity_worst    fractal_dimension_worst   0.686511
diagnosis          concavity_worst           0.659610
```

Fig. 11. Correlation between columns

The image above is the output of the value of correlations between each column. From the values above, it can be said that the column with the highest correlation is the radius_worst with the area_worst, which has a correlation value of 0.984015.

```
# Visualisasi Korelasi Antar Kolom
corr = df.corr(method='pearson')
plt.figure(figsize=(30,30))
sns.heatmap(corr, cbar = True,  square = True, annot=True, fmt= '.2f', cmap= 'coolwarm')
```

```
<AxesSubplot: >
```

Fig. 12. Data Heatmap

The image above is a heat map of the analyzed data, where from the heat map we can see the values of correlations between each column. From the heat map, we can also see the highest correlation values, and in addition to looking at the numbers in each column, we can determine the high correlation value from the colors. The darker the color, the better the correlation value.

## *E. Train Test*

```
x_train, x_test, y_train, y_test = train_test_split(X_new, y_new, stratify=y_new, random_state=21)
print ('Train set:', x_train.shape, y_train.shape)
print ('Test set:', x_test.shape, y_test.shape)
```

```
Train set: (535, 10) (535,)
Test set: (179, 10) (179,)
```

Fig. 13. Train & Test Set

The image above is the output of the Train Set and Test Set results, where 535 data are trained to make a prediction. In addition, 179 data are tested to see their accuracy.

```
# Count the occurrences of each class in the resampled data
class_counts = y_new.value_counts()

# Plot the class distribution
plt.figure(figsize=(8, 6))
plt.bar(class_counts.index, class_counts.values)
plt.xlabel('Class')
plt.ylabel('Count')
plt.title('Balanced Class Distribution')
plt.show()
```

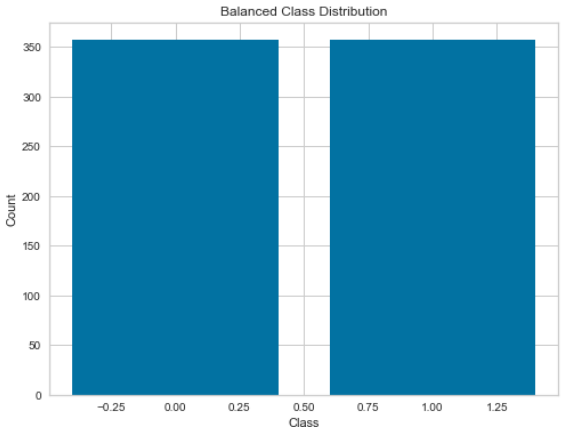


Fig. 14. Y Variable After SMOTE

The figure above shows the count of the occurrences of each class in the resampled data. It calculates and visualizes the distribution of the resampled classes in the form of a histogram. The histogram provides information about the frequency of each class in the resampled data.

## *F. Machine Learning (KNN)*

```
error_rate =[]
for i in range(1, 30):
    knn = KNeighborsClassifier(n_neighbors= i)
    knn.fit(x_train, y_train)
    y_pred_i = knn.predict(x_test)
    error_rate.append(np.mean(y_pred_i != y_test))

plt.figure(figsize = (10, 6))
plt.plot(range(1,30), error_rate, color='blue', linestyle='dashed', marker='o', markerfacecolor='red', markersize=8)
plt.title('Error Rate vs. K Value', fontsize=20)
plt.xlabel('K', fontsize=15)
plt.ylabel('Error (misclassification) Rate', fontsize=15)

lowest_err_index = error_rate.index(min(error_rate)) + 1
print("Lowest Error rate is at index:", lowest_err_index, "with value of ",  min(error_rate))
```

```
Lowest Error rate is at index: 9 with value of  0.005586592217877095
```

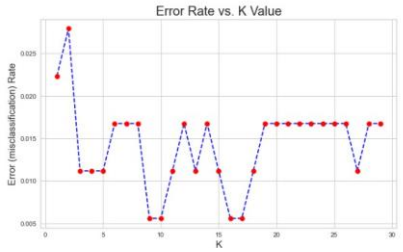


Fig. 15. Error Rate vs. K Value

The image above is a visualization of the error rate based on the number of K values (the number of neighbors in the KNN algorithm). This figure is used to determine the best K value based on the resulting error. From the image above, the smallest error rate is at K = 9, 10, 16, and 17 (error value). But the obtained K value is not necessarily the k value that will be used in the KNN model. In this research, we also use GridSearch to identify the best parameter that can be used in a KNN Model.

```
param_grid = {
    'n_neighbors': [2,3,5],
    'weights': ['uniform', 'distance'],
    'p': [1, 2,3]
}

# Perform grid search to find the best hyperparameters
grid_search = GridSearchCV(knn, param_grid, cv=5)
grid_search.fit(x_train, y_train)

# Get the best hyperparameters
best_params = grid_search.best_params_

print("The Best Param for KNN:", best_params)
```

```
The Best Param for KNN: {'n_neighbors': 3, 'p': 1, 'weights': 'uniform'}
```

Fig. 16. Best Param for KNN

The figure above indicates the best hyperparameters found through a grid search using cross-validation for the K-nearest neighbors (KNN) classifier. The output shows that the best hyperparameters for the KNN classifier are n_neighbors=3, p=1, and weights='uniform'. These hyperparameters indicate that the classifier considers the 3 nearest neighbors, and uses the Minkowski distance with a power of 1 (equivalent to the Manhattan distance).

```
knn = KNeighborsClassifier(**best_params)
knn.fit(x_train, y_train)
y_pred_knn = knn.predict(x_test)

#Measure Accuracy
print("CONFUSION MATRIX KNN: \n", confusion_matrix(y_test, y_pred_knn))
print("ACCURACY SCORE: ", accuracy_score(y_test, y_pred_knn)*100, "%")
print("RECALL SCORE: ", recall_score(y_test, y_pred_knn)*100, "%")
print("F-MEASURE SCORE: ", f1_score(y_test, y_pred_knn)*100, "%")

skplt.metrics.plot_confusion_matrix(y_test, y_pred_knn, figsize=(6,6), cmap='YlGnBu')
```

```
CONFUSION MATRIX KNN:
 [[89  0]
 [ 0 90]]
ACCURACY SCORE:  100.0 %
RECALL SCORE:  100.0 %
F-MEASURE SCORE:  100.0 %
```

Fig. 17. KNN - Accuracy Score

The evaluation includes calculating the confusion matrix, accuracy score, recall score, and F-measure score.

- The confusion matrix indicates that there are 89 true negatives, 0 false positives, 0 false negatives, and 90 true positives.
- The accuracy score is approximately 100%, meaning that the KNN classifier correctly predicted the labels for 100% of the test samples.
- The recall score is approximately 100%, indicating that the classifier correctly identified 100% of the positive samples.
- The F-measure score is also approximately 100%, representing a balanced measure of precision and recall.

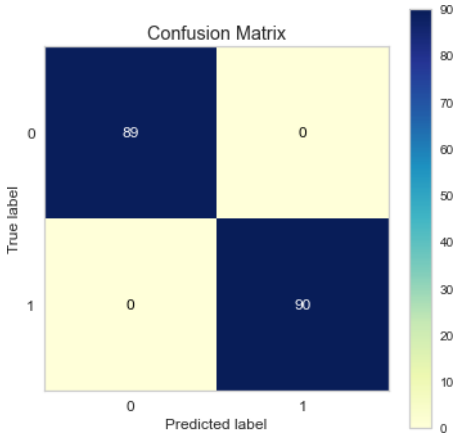


Fig. 18. KNN - Confusion Matrix

The matrix indicates 89 true negatives, 0 false positives, 0 false negatives, and 90 true positives.

## G. *Machine Learning (Decision Tree)*

```
from sklearn.tree import DecisionTreeClassifier

tree = DecisionTreeClassifier()
tree.fit(x_train, y_train)
print("Accuracy on training set : {:.3f}".format(tree.score(x_train, y_train)))
print("Accuracy on test set     : {:.3f}".format(tree.score(x_test, y_test)))
```

```
Accuracy on training set : 1.000
Accuracy on test set     : 0.950
```

Fig. 19. Decision Tree training and test accuracy

In the picture above it can be seen that the accuracy value for the training results is 1, while for the test results, it is 0.950.

```
# Make predictions on the test set
y_pred_tree = tree.predict(x_test)

print("CONFUSION MATRIX Decision Tree: \n", confusion_matrix(y_test, y_pred_tree))
print("ACCURACY SCORE: ", accuracy_score(y_test, y_pred_tree)*100, "%")
print("RECALL SCORE: ", recall_score(y_test, y_pred_tree)*100, "%")
print("F-MEASURE SCORE: ", f1_score(y_test, y_pred_tree)*100, "%")

skplt.metrics.plot_confusion_matrix(y_test, y_pred_tree, figsize=(6,6), cmap='YlGnBu')
```

```
CONFUSION MATRIX Decision Tree:
 [[83  6]
 [ 3 87]]
ACCURACY SCORE:  94.97206703910615 %
RECALL SCORE:  96.66666666666667 %
F-MEASURE SCORE:  95.08196721311477 %
```

Fig. 20. Decision Tree Accuracy

The evaluation includes calculating the confusion matrix, accuracy score, recall score, and F-measure score.

- The confusion matrix indicates that there are 83 true negatives, 6 false positives, 3 false negatives, and 87 true positives.
- The accuracy score is approximately 94.97%, meaning that the KNN classifier correctly predicted the labels for 94.97% of the test samples.
- The recall score is approximately 96.67%, indicating that the classifier correctly identified 96.67% of the positive samples.
- The F-measure score is also approximately 95.08%, representing a balanced measure of precision and recall.

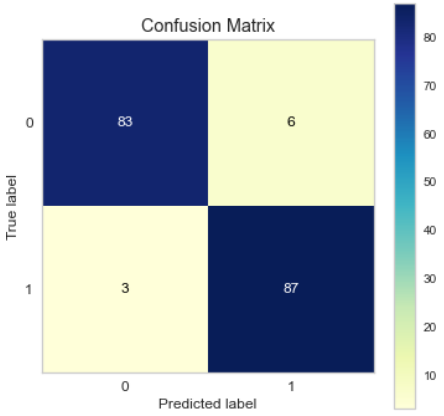


Fig. 21. Decision Tree Confusion Matrix

The confusion matrix indicates 83 true negatives, 6 false positives, 3 false negatives, and 87 true positives.

```
print("Feature Importances :\n{}".format(tree.feature_importances_))
```

```
Feature Importances :
[0.01903987 0.07103817 0.76632465 0.         0.04609927 0.01262725
 0.00783752 0.04103491 0.03599835 0.        ]
```

Fig. 22. Feature Importances

In the picture above, there are 10 feature importance values as follows: 0.01903987

0.07103817 0.76632465 0. 0.04609927 0.01262725 0.00783752 0.04103491 0.03599835 0.

```
def plot_feature_importances_diag(model):
    n_features = X.shape[1]
    plt.barh(range(n_features), model.feature_importances_, align='center')
    plt.yticks(np.arange(n_features), X)
    plt.xlabel("Feature importance")
    plt.ylabel("Feature")

plot_feature_importances_diag(tree)
```

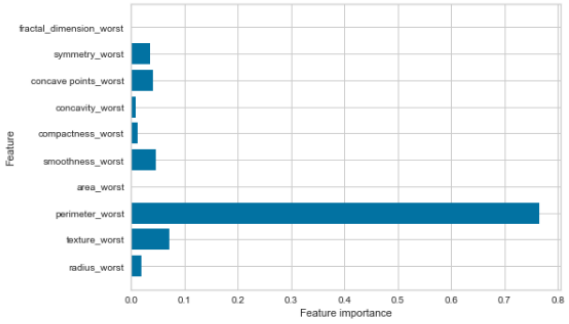


Fig. 23. Feature Importance bar plot

Figure 22 shows that there are 10 bars that previously had important feature values. From the visualization above, it can be seen that perimeter_worst has the highest feature importance value compared to the other four bars.

```
import graphviz
from sklearn.tree import export_graphviz
export_graphviz(tree, out_file="tree.dot", class_names=["Tidak Ganas","Ganas"], impurity=False, filled=True)

with open("tree.dot") as f:
    dot_graph = f.read()
graphviz.Source(dot_graph)
```

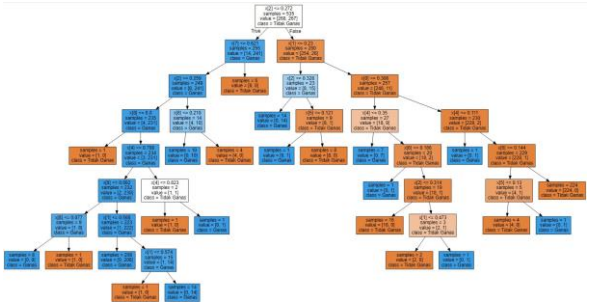

Fig. 24. Tree Visualization

Figure 23 is a visualization of the decision tree, where it can be seen that the head of the tree itself is x[1] with a value <= 0.276 with a total sample of 535. Then it will split into 2 roots, namely True and False which will divide it into 2 different samples, namely sample 264 and sample 271. In the end, it will create several derivatives until the sample becomes 1.

```
import pickle

filename = 'tumor_model.sav'
pickle.dump(knn, open(filename, 'wb'))

# loading the saved model
loaded_model = pickle.load(open('tumor_model.sav', 'rb'))

input_data = (25.38,17.33,184.60,2019.0,0.1622,0.6656,0.7119,0.2654,0.4601,0.11890)

# changing the input_data to numpy array
input_data_as_numpy_array = np.asarray(input_data)

# reshape the array as we are predicting for one instance
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

prediction = knn.predict(input_data_reshaped)
print(prediction)

if (prediction[0] == 0):
    print('Tumor JINAK')
else:
    print('Tumor Ganas')
```

```
[0]
Tumor JINAK
```

Fig. 25. Deployment Code

The image above is the code for deploying, where we will use the pickle environment. Then we will use data testing with the input_data variable. After that, make predictions using the KNN algorithm, because it is a better algorithm than the decision tree. After that create a .py file to make streamlit run.

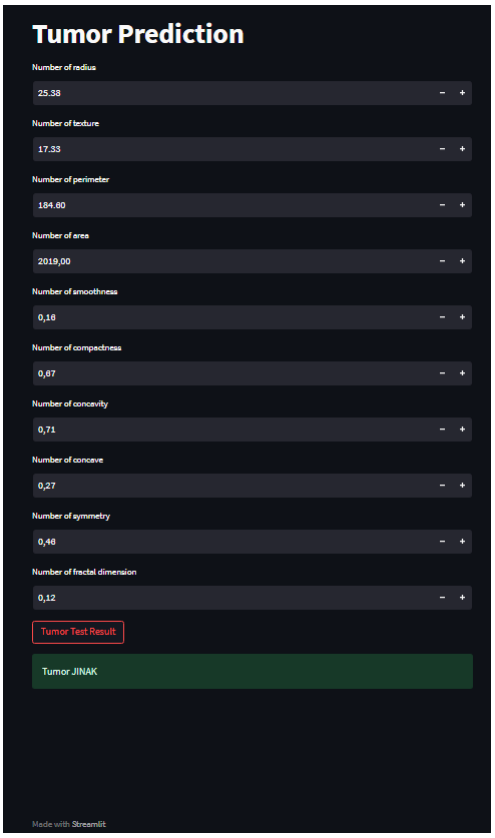


Fig. 26. Deployment Using Streamlit

The image above is the result of the deployment which was carried out with data from input_data, which in the end after inputting data. Predictions indicate that the tumor is not malignant.

## IV. CONCLUSION

Researchers can conclude that there are differences between the use of the KNN algorithm and the use of the Decision Tree. Meanwhile, when compared with other studies using the Decision Tree algorithm, such as research 1 entitled Diagnosis of Breast Cancer using Decision Tree Data Mining Technique, the accuracy of the data was 94.56% [11], it was concluded that the accuracy used the decision tree algorithm with the theme of health. regarding cancer to get an accuracy of 94.56%. When compared with the accuracy value of this study, the accuracy obtained is better than the comparison journal, which is 94.978%. Meanwhile, when compared with another 2nd study entitled Performance Analysis of Genetic Algorithm with kNN and SVM for Features Selection in Tumor Classification, shows the use of the kNN algorithm with an accuracy of 95% [12]. So when compared with this study which talks about the same theme, namely tumors and using the kNN algorithm, the accuracy results are 100%.

Therefore, in the end, this study shows that when using the KNN algorithm will get better results compared to using the Decision Tree algorithm because the results accuracy and results of the visualization of the confusion matrix are better to provide better classification to patients who will or nearly got a tumor. Therefore, the researchers concluded that the tumor dataset is very suitable for the K-Nearest Neighbor (KNN) classification algorithm, compared to the Decision Tree.

## WORK DIVISION

Rafi Aldino: Making coding preprocessing, compiling Introduction to methodology, making conclusions, and making deployment.

**Kenny Budiarso Lawson**: Making visualization and modeling, compiling results, and discussion

**Nabila Az Zahra**: Make an evaluation, compile results, and discussion

## REFERENCES


[1] Winslow, T. (2021, May 5). *What is cancer?* National Cancer Institute. Retrieved December 21, 2022, from https://www.cancer.gov/about-cancer/understanding/what-is-cancer

[2] Thomas, C. A. (2020). *Cancer Today*. Global Cancer Observatory. Retrieved December 21, 2022, from https://gco.iarc.fr/today

[3] Komputer, M. M. A. F. I., & Komputer, F. I. (2017, September 8). *Implementasi metode dempster-shafer untuk mendiagnosis jenis tumor Jinak Pada Manusia*. Implementasi Metode Dempster-Shafer untuk Mendiagnosis Jenis Tumor Jinak pada Manusia | Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer. Retrieved December 20, 2022, from https://j-ptiik.ub.ac.id/index.php/jptiik/article/view/1494

[4] Farokhah, L. (2020, July 22). *Implementasi K-nearest neighbor Untuk Klasifikasi Bunga Dengan Ekstraksi FITUR Warna RGB*. Jurnal Teknologi Informasi dan Ilmu Komputer. Retrieved December 20, 2022, from https://jtiik.ub.ac.id/index.php/jtiik/article/view/2608

[5] Mardi, Y. (2017). Data Mining : Klasifikasi Menggunakan Algoritma C4.5. *Jurnal Edik Informatika*, 2-4.

[6] Aryanti, D., & Setiawan, J. (2018). Ultima InfoSys. Visualisasi Data Penjualan dan Produksi PT Nitto Alam Indonesia Periode 2014-2018, 2.

[7] Elfaladonna, F., & Rahmadani, A. (2019). Sintech Journal. Analisa Metode Classification-Decission Tree Dan Algoritma C.45 Untuk Memprediksi Penyakit Diabetes Dengan Menggunakan Aplikasi Rapid Miner, 3.

[8] Sutoyo, I. (2018). Implementasi Algoritma Decision Tree Untuk Klasifikasi Data Peserta Didik. Jurnal Pilar Nusa Mandiri, 14(2), 217-224.

[9] Binsar, F., & Mauritsius, T. (2020, September 18). *Cross-industry standard process for data mining (CRISP-DM)*. MMSI BINUS University. Retrieved December 21, 2022, from https://mmsi.binus.ac.id/2020/09/18/cross-industry-standard-process-for-data-mining-crisp-dm/

[10] Naraei, P. (2020). *Research framework*. Research Framework - an overview | ScienceDirect Topics. Retrieved December22, 2022, from https://www.sciencedirect.com/topics/computer-science/research-framework

[11] R. Sumbaly, N. Vishnusri, and S. Jeyalatha, "International Journal of Computer Applications," Diagnosis of Breast Cancer using Decision Tree Data Mining Technique , vol. 98, no. 10, pp. 16–23, Mar. 2015.

[12] C. Gunavathi and K. Premalatha, "World Academy of Science, Engineering and Technology," Performance Analysis of Genetic Algorithm with kNN and SVM for Feature Selection in Tumor Classification, vol. 8, no. 8, pp. 1357–1363, 2014.


## APPENDIX

1. Project Github Link
   https://github.com/KennyLawson/ML-Kelompok6