

Implementasi Algoritma K-Nearest Neighbor Dalam Klasifikasi Penyakit Kanker Paru Paru

Zikri Hadiansyah*, Zaenur Rozikin, Muhamad Fatchan

Fakultas Teknik, Program Studi Teknik Informatika, Universitas Pelita Bangsa, Kab. Bekasi, Indonesia
Email: ^{1,*}zikri.hadiansyah21@email.com, ²zaenurrozikin@pelitabangsa.ac.id, ³fatchan@pelitabangsa.ac.id

Email Penulis Korespondensi: zikri.hadiansyah21@email.com

Submitted: 04/11/2024; Accepted: 15/11/2024; Published: 15/11/2024

Abstrak—Kanker paru-paru merupakan salah satu jenis kanker dengan angka kematian tertinggi di dunia. Merokok menjadi faktor risiko utama yang menyebabkan 20% kematian akibat kanker dan 70% kematian akibat kanker paru-paru di dunia. Meski demikian, orang yang tidak merokok juga dapat menderita kanker paru-paru, terutama jika sering terpapar polusi udara, tinggal di lingkungan yang tercemar zat berbahaya, atau memiliki keluarga yang menderita kanker paru-paru. Deteksi dini dalam klasifikasi penyakit kanker paru paru menjadi faktor penting dalam meningkatkan peluang hidup pasien. Oleh karena itu, penelitian ini bertujuan untuk mengklasifikasikan penyakit kanker paru paru menggunakan algoritma *K-Nearest Neighbor*. Algoritma *K-Nearest Neighbor* dipilih karena pada berbagai penelitian memiliki tingkat akurasi yang lebih baik dibandingkan dengan algoritma supervised learning lainnya. Untuk mengatasi ketidakseimbangan data digunakan teknik *Random oversampling*. Berdasarkan pengujian yang dilakukan menggunakan *Confusion Matrix*, hasil dari pengukuran nilai performa *Accuracy*, *Precision*, *Recall* dan *f1-score* menggunakan algoritma *K-Nearest Neighbor* dengan teknik *Random oversampling*, dapat disimpulkan algoritma *K-Nearest Neighbor* mendapat nilai *Accuracy* 0.99, *Precision* 0.99, *Recall* 0.99 dan *f1-score* 0.99.

Kata Kunci: Kanker Paru-Paru; Merokok; *K-Nearest Neighbors*; *Random Oversampling*; *Confusion Matrix*

Abstract—Lung cancer is one type of cancer with the highest death rate in the world. Smoking is the main risk factor that causes 20% of cancer deaths and 70% of lung cancer deaths in the world. However, people who do not smoke can also suffer from lung cancer, especially if they are frequently exposed to air pollution, live in an environment contaminated with dangerous substances, or have a family member who suffers from lung cancer. Early detection in the classification of lung cancer is an important factor in increasing the patient's chances of survival. Therefore, this study aims to classify lung cancer using the *K-Nearest Neighbor* algorithm. The *K-Nearest Neighbor* algorithm was chosen because in various studies it has a better level of accuracy compared to other supervised learning algorithms. To overcome data imbalance, the *Random oversampling* technique is used. Based on tests carried out using the *Confusion Matrix*, the results of measuring the performance values of *Accuracy*, *Precision*, *Recall* and *f1-score* using the *K-Nearest Neighbor* algorithm with *Random oversampling* technique, it can be concluded that the *K-Nearest Neighbor* algorithm received an *Accuracy* value of 0.99, *Precision* 0.99, *Recall* 0.99 and *f1-score* 0.99.

Keywords: Lung Cancer; Smoking; *K-Nearest Neighbors*; *Random Oversampling*; *Confusion Matrix*

1. PENDAHULUAN

Saat ini kanker menjadi penyakit penyebab kematian tertinggi di Indonesia setelah stroke dan hipertensi. Data prevalensi penyakit ini naik dari 1,4% menjadi 1,8% pada tahun 2018. Merokok menjadi faktor risiko utama yang menyebabkan 20% kematian akibat kanker dan 70% kematian akibat kanker paru-paru di dunia[1]. Kanker paru-paru paling sering terjadi akibat kebiasaan merokok. Meski demikian, orang yang tidak merokok juga dapat menderita kanker paru-paru, terutama jika sering terpapar polusi udara, tinggal di lingkungan yang tercemar zat berbahaya, atau memiliki keluarga yang menderita kanker paru-paru[2].

Menghirup asap rokok merupakan salah satu kebiasaan yang sulit terhindarkan, data dari Badan Pusat Statistik (BPS) mencatat, persentase penduduk Indonesia berusia 15 tahun ke atas yang merokok sebesar 28,62% pada 2023. Persentase tersebut meningkat 0,36% poin dari tahun lalu yang sebesar 28,26%[3]. Bukan hanya orang yang merokoknya saja yang akan terkena dampak dari asap rokok, tetapi risiko kesehatan juga mengancam para perokok pasif. Perokok pasif adalah orang di sekitar yang ikut menghirup asap tembakau orang lain. Asap dapat terhirup jika berada di dekat atau dalam satu ruangan tertutup dengan orang yang sedang merokok. Jika para perokok pasif ini terlalu sering menghirup asap rokok, maka risiko terkena kanker paru tak dapat dihindari lagi. Oleh karena itu, penting diadakannya klasifikasi penyakit kanker paru-paru menggunakan gejala awal dan faktor risiko untuk mencegah hal tersebut.

Klasifikasi yaitu proses mengelompokkan objek-objek dengan karakteristik yang mirip dalam beberapa kelas. Pada umumnya pengklasifikasian dokumen diwakili oleh kalimat-kalimat penting dengan menentukan ciri-ciri atau karakteristik[4]. Salah satu metode *machine learning* dalam klasifikasi yaitu *K-Nearest Neighbors*. Algoritma *K-Nearest Neighbor* atau sering disingkat KNN merupakan algoritma klasifikasi berdasarkan sejumlah *K* data yang terdekat atau tetangga terdekat. Kelebihan dari KNN, yaitu tangguh terhadap data latih yang memiliki noise dan efektif jika digunakan pada data latih dengan jumlah yang besar [5].

Adapun beberapa penelitian sebelumnya yang telah memberikan dukungan bagi penelitian ini. Penelitian pertama yang dilakukan oleh Teguh Abdi Mangun, Odi Nurdian dan Ade Irma Purnamasari dengan judul “Lung Cancer Analysis Using K-Nearest Neighbor Algorithm” pada tahun 2023. penelitian ini menggunakan

algoritma *K-Nearest Neighbor* dengan studi kasus Kanker paru paru yang dievaluasi dengan *confusion matrix* mendapat akurasi sebesar 80.40% [6]. Penelitian kedua yang dilakukan oleh Widhi Ramdhani, David Bona, Rafi Bagus Musyaffa & Chaerur Rozikin dengan judul “Klasifikasi Penyakit Kanker Payudara Menggunakan Algoritma *K-Nearest Neighbor*” pada tahun 2022. Evaluasi atau pengujian nilai performa dari pemodelan algoritma KNN yang digunakan dengan melakukan beberapa percobaan nilai K. Pengujian dengan nilai akurasi tertinggi diperoleh dari nilai $k=21$ dan $k=11$ dengan nilai akurasi mencapai 98% [7]. Penelitian ketiga yang dilakukan oleh Adinda Amalia, Ati Zaidiah & Ika Nurlaili Isnainiyah dengan judul “Prediksi Kualitas Udara Menggunakan Algoritma *K-Nearest Neighbor*” pada tahun 2022. Pada evaluasi model algoritma dengan menggunakan *confusion matrix* menghasilkan bahwa nilai $K = 7$ memiliki performa yang terbaik dimana nilai akurasinya sebanyak 96%, presisi 92%, *recall* 95%, dan *f-measure* 93%[8].

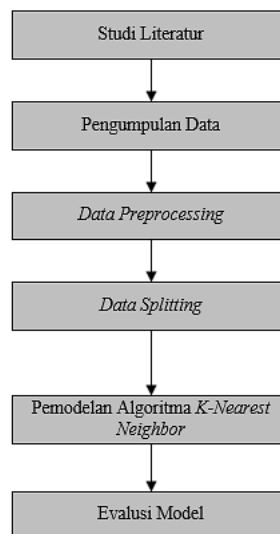
Penelitian keempat yang dilakukan oleh Afivatu Pratama Agustin, Abd. Charis Fauzan & Harliana dengan judul “Implementasi *K-Nearest Neighbor* Dengan Jarak *Minkowski* Untuk Deteksi Dini Covid 19 Pada Citra *Ct-Scan* Paru-Paru” pada tahun 2022. Pada penelitian ini menggunakan dua skenario pengujian, yaitu Skenario ke-1 pengujian dengan data latih berjumlah 400, meliputi 200 citra *Covid-19*, 200 citra *Non Covid-19* dan data uji berjumlah 346 meliputi 149 citra *Covid-19* dan citra 197 *Non Covid-19*. Skenario ke-2 pengujian dengan jumlah data latih 595, meliputi 279 citra *Covid-19*, citra *Non Covid-19* 316 dan data uji berjumlah 147, meliputi 69 citra *Covid-19* dan 78 citra *Non Covid-19*. Hasil yang didapat dari uji coba skenario ke-1 yaitu nilai $K = 1$ *Accuracy* 62%, *Precision* 57%, *Recall* 55%, *F1-Score* 55%. $K=3$ *Accuracy* 65%, *Precision* 38% *Recall* 73%, *F1-Score* 49% dan $K=5$ memiliki *Accuracy* 62%, *Precision* 20%, *Recall* 73%, *F1-Score* 31%. Sedangkan Uji coba menggunakan skenario ke 2 hasil yang diperoleh yaitu $K=1$ dengan *Accuracy* 80%, *Precision* 70%, *Recall* 84% dan *F1 Score* 76%. $K=3$ dengan *Accuracy* 72%, *Precision* 46%, *Recall* 88% dan *F1-Score* 60%, sedangkan $K=5$ *Accuracy* 63%, *Precision* 20%, *Recall* 100% dan *F1-Score* 22%[9]. Penelitian kelima yang dilakukan oleh Sri Indra Maiyanti, Des Alwine Zayanti, Yuli Andriani, Bambang Suprihatin, Anita Desiani, Aulia Salsabila & Nyayu Chika Marselina dengan judul “Perbandingan Klasifikasi Penyakit Kanker Paru-Paru Menggunakan Support Vector Machine Dan *K-Nearest Neighbor*” pada tahun 2023. Algoritma SVM dan KNN sama-sama menghasilkan nilai akurasi di atas 90% yang menunjukkan bahwa kedua algoritma tersebut memiliki performa yang baik dalam melakukan prediksi kanker paru-paru pada penelitian ini[10].

Penelitian ini diharapkan dapat digunakan untuk melihat hasil akurasi dalam klasifikasi kanker paru-paru menggunakan metode algoritma *K-Nearest Neighbors* (KNN) yang diharapkan dapat memberikan informasi tentang performa yang paling baik dan sudut pandang berbeda dalam pengaplikasiannya sehingga menghasilkan klasifikasi yang lebih akurat dalam pendiagnosaan penyakit kanker paru-paru.

2. METODOLOGI PENELITIAN

2.1 Kerangka penelitian

Dalam penelitian ini, penulis melakukan beberapa tahapan. Tahap-tahap tersebut adalah Studi literatur, pengumpulan data, *data preprocessing*, *data splitting*, pembuatan model dan terakhir evaluasi data. Tahapan penelitian dapat dilihat pada Gambar 1.



Gambar 1. Kerangka Penelitian

Pada gambar 1, dapat dilihat bahwa tahap yang pertama yaitu studi literatur, proses ini dilakukan dengan meninjau beberapa jurnal terdahulu yang digunakan sebagai referensi pada penelitian ini. Setelah itu tahap kedua yaitu proses pengumpulan data. Pengumpulan data dilakukan dengan mencari data sekunder yaitu dataset survey

lung cancer yang terdapat pada *website kaggle*. Tahap ketiga yaitu *data preproccesing* yang dilakukan dengan menormalisasi data agar dapat dioleh lebih mudah dan tertata. Tahap keempat yaitu *data splitting* yang dilakukan dengan membagi data menjadi 2, yaitu *data training* dan *data testing*. Tahap selanjutnya adalah pemodelan model algoritma *K-Nearest Neighbor* yang dilakukan komparasi dengan menggunakan teknik *random oversampling* dan tidak menggunakan teknik *random oversampling*. Tahap terakhir adalah evaluasi model algoritma. Evaluasi model dilakukan untuk mengetahui apakah model algoritma *K-Nearest Neighbor* layak digunakan untuk klasifikasi penyakit kanker paru paru atau tidak.

2.2 K-Nearest Neighbor

K-Nearest Neighbor adalah metode algoritma yang mengklasifikasikan objek berdasarkan data pembelajaran (neighbors) yang jaraknya paling dekat dengannya, yang dihitung dengan nilai geometris. Algoritma tetangga paling dekat (K) digunakan dalam kasus ini untuk memprediksi klasifikasi objek berdasarkan data pembelajaran yang paling dekat (K), dan setelah nilai K ditentukan, diambil 1 (n) tetangga untuk melihat apakah semua tetangga tersebut mengalami kanker atau tidak [11]. Prinsip kerja *K-Nearest Neighbor* adalah mencari jarak terdekat antara data yang akan dievaluasi dengan K tetangga (*Neighbor*) terdekatnya dalam data pelatihan [12]. Proses kerja diatur sebagai berikut.

1. Menentukan parameter k (jumlah tetangga paling dekat).
2. Menghitung kuadrat jarak *euclidean (euclidean distance)* masing-masing obyek terhadap data sampel yang diberikan. Visualisasi algoritma *K-Nearest Neighbor* dapat dilihat pada gambar 2.



Gambar 2. *K-Nearest Neighbor*

2.3 Euclidean Distance

Euclidean Distance adalah suatu metode pencarian antara dua titik variabel, semakin dekat dan mirip suatu data maka semakin kecil jarak antara dua titik tersebut. Euclidean Distance dikatakan baik jika suatu data baru memiliki jarak minimum dan memiliki kemiripan yang tinggi. Perhitungan jarak dilakukan dengan rumus Euclidean Distance sebagai berikut.

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_{training}^i - y_{testing}^i)^2} \quad (1)$$

Keterangan dari rumus tersebut sebagai berikut : $d(x, y)$ = Jarak, $x_{training}^i$ = Data training, $y_{testing}^i$ = Data testing, i = Variabel data dan n = Dimensi data.

2.4 Confusion matrix

Confusion matrix adalah metrik evaluasi yang digunakan untuk mengukur performa model *machine learning* yang telah dikembangkan. *Output* dari *confusion matrix* meliputi variabel *precision*, *recall*, *f1-score*, dan *support* untuk setiap kelas setelah melalui proses pelatihan dan pengujian dengan algoritma yang telah ditentukan. *Confusion matrix* memiliki empat nilai utama, yaitu *True Positive*, *True Negative*, *False Positive*, dan *False Negative*. Dari kombinasi nilai-nilai ini, kita akan dapat menghitung performa model dengan mengukur *precision*, yaitu kemampuan model untuk mengidentifikasi kasus positif secara akurat; *recall*, yang menunjukkan seberapa baik model dalam menemukan semua kasus positif dari seluruh iterasi yang dilakukan; serta *f1-score*, yang mewakili nilai harmonik atau rata-rata perhitungan antara *recall* dan *precision*. Rincian perhitungan untuk *precision*, *recall*, dan *f1-score* dijelaskan pada gambar poin 1,2 dan 3 [13].

$$\text{Precision} = \frac{TP}{(TP+FP)} \quad (2)$$

$$\text{Recall} = \frac{TP}{(TP+FN)} \quad (3)$$

$$\text{F1-Score} = \frac{2 * (\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})} \quad (4)$$

2.5 Random Oversampling

Random Oversampling merupakan teknik transfer data dari kelas minoritas ke data training secara acak. Proses pemberian data diulang sampai jumlah data dari kelas minoritas sama rata dengan jumlah data kelas mayoritas. Langkah pertama yang dilakukan adalah dengan menghitung selisih antara data kelas mayoritas dan data kelas minoritas. Setelah itu, dilakukan perulangan sebanyak hasil penghitungan data sambil membaca data kelas minoritas secara acak dan dimasukkan ke dalam data training [14].

2.6 Dataset

Dataset adalah sebuah kumpulan data yang berasal dari informasi-informasi pada masa lalu dan siap untuk dikelola menjadi sebuah informasi baru. Kumpulan data yang ada di dataset bisa di-load dari sumber data apa pun yang valid. *Dataset* tetap ada di memori dan data di dalamnya bisa dimanipulasi dan di-update tanpa bergantung pada *database* asalnya [15]. *Dataset* yang digunakan pada penelitian ini adalah dataset survey lung cancer yang dikumpulkan dari website sistem prediksi kanker paru online, diperoleh dari website *Kaggle* yaitu : <https://www.kaggle.com/datasets/nancyalaswad90/lung-cancer>. Dataset *survey lung cancer* terdiri dari 310 data dan 16 atribut. Atribut dataset *survey lung cancer* ditunjukkan pada gambar 3.

```
df.info()

[26]
... <class 'pandas.core.frame.DataFrame'>
RangeIndex: 309 entries, 0 to 308
Data columns (total 16 columns):
#   Column                Non-Null Count  Dtype
---  -
0   GENDER                 309 non-null    object
1   AGE                    309 non-null    int64
2   SMOKING                 309 non-null    int64
3   YELLOW_FINGERS         309 non-null    int64
4   ANXIETY                 309 non-null    int64
5   PEER_PRESSURE          309 non-null    int64
6   CHRONIC_DISEASE        309 non-null    int64
7   FATIGUE                 309 non-null    int64
8   ALLERGY                309 non-null    int64
9   WHEEZING               309 non-null    int64
10  ALCOHOL_CONSUMING      309 non-null    int64
11  COUGHING               309 non-null    int64
12  SHORTNESS OF BREATH    309 non-null    int64
13  SWALLOWING DIFFICULTY  309 non-null    int64
14  CHEST PAIN             309 non-null    int64
15  LUNG_CANCER            309 non-null    object
dtypes: int64(14), object(2)
memory usage: 38.8+ KB
```

Gambar 3. Atribut yang digunakan dalam Prediksi Kanker Paru-paru

3. HASIL DAN PEMBAHASAN

3.1 Data Preprocessing

Data Preprocessing merupakan salah satu tahapan dalam melakukan mining data. Sebelum menuju ke tahap pemrosesan. Data mentah akan diolah terlebih dahulu. *Data Preprocessing* atau praproses data biasanya dilakukan melalui cara eliminasi data yang tidak sesuai. Selain itu dalam proses ini data akan diubah dalam bentuk yang akan lebih dipahami oleh sistem [16]. Pada tahap ini sebelum masuk ketahap modeling dataset terlebih dahulu diproses melalui tahap *data cleaning*, *Handling data imbalance* dan *data transformation*.

1. Data cleaning

Pada proses ini dilakukan pengecekan *missing value*, yaitu memastikan bahwa tidak ada data yang hilang. Hasil dari eksekusi bahwa dataset ini sudah tidak ditemukan adanya *missing value* ditunjukkan pada gambar 4.

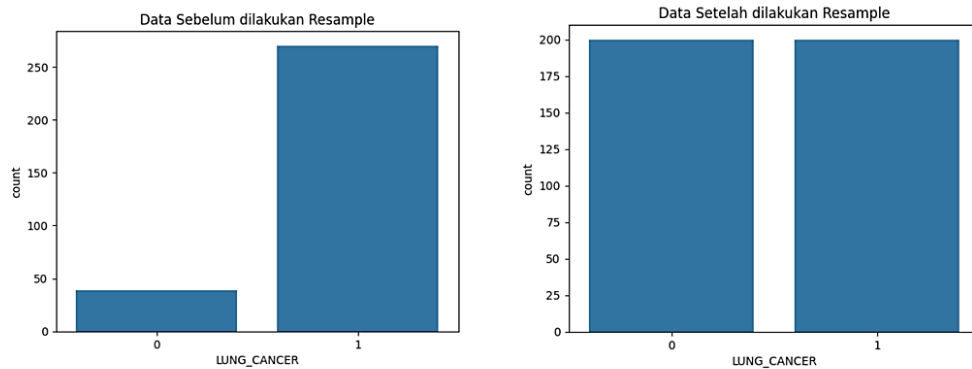
```
#Check data missing
df.isnull().sum()

[7]
... GENDER                0
AGE                0
SMOKING            0
YELLOW_FINGERS     0
ANXIETY            0
PEER_PRESSURE      0
CHRONIC_DISEASE    0
FATIGUE            0
ALLERGY            0
WHEEZING           0
ALCOHOL_CONSUMING  0
COUGHING           0
SHORTNESS OF BREATH 0
SWALLOWING DIFFICULTY 0
CHEST PAIN         0
LUNG_CANCER        0
dtype: int64
```

Gambar 4. Check Missing Value

2. Handling data imbalance

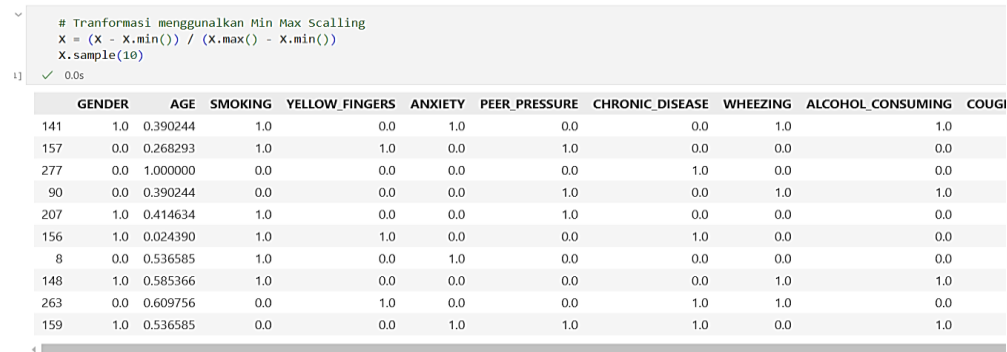
Pada tahap ini untuk mengatasi ketidakseimbangan data, digunakan teknik *random overampling* dengan perintah *resample* dari modul *sklearn.utils*. Fungsi *resample* memungkinkan untuk mengambil sampel acak dari data yang sudah ada. Pada gambar 5 dapat dilihat persebaran data sebelum dan sesudah dilakukan resample data.



Gambar 5. Random overampling

3. Data Transformation

Proses selanjutnya adalah Transformasi data atau Normalisasi Data. Normalisasi dilakukan agar setiap fitur pada dataset memiliki rentang yang sama yaitu 0 sampai 1 menggunakan perintah *MinMaxScaler* agar tidak memiliki ketimpangan rentang fitur. Pada Gambar 6 dapat dilihat data yang sudah ternormalisasi. data yang sudah di transformasikan menggunakan rumus *MinMax Scaling*, sebagai berikut:



Gambar 6. Transformasi Data

3.2 Data Modeling

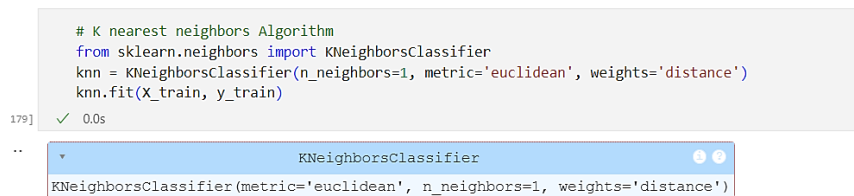
3.2.1 Pemodelan Data Tanpa Teknik Random Oversampling (ROS)

Untuk melihat dampak dari penggunaan teknik *Random oversampling* dalam penelitian ini, langkah awalnya adalah membuat model tanpa menerapkan teknik ROS untuk menilai kinerja model awal. Untuk membuat model *K-Nearest Neighbor (KNN)* dengan 16 variabel yang sudah disiapkan dapat dilihat pada Tabel 1. Setelah melakukan *preprocessing data*, langkah selanjutnya yaitu melakukan proses *data transformation* atau normalisasi data. Tujuan dari normalisasi data adalah untuk mengubah nilai numerik di setiap variabel sehingga setiap variabel memiliki rata-rata nilai 0 dan 1 dan variansinya menjadi 0 dan 1 juga. Efek positif dari normalisasi data adalah untuk mengurangi besarnya nilai variabel dan mengubahnya menjadi skala yang lebih proporsional seperti pada Gambar 6. Setelah melakukan *preprocessing data*, tahap selanjutnya adalah melakukan proses *Data Splitting*. Pada tahap *Data splitting*, dilakukan pembagian data menjadi 70% Data Training dan 30% Data Testing .

```
### Train Test Split
from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test=train_test_split(X,y,test_size=0.30,random_state=0)
```

Gambar 7. Splitting Data Train & Data Test



```
# K nearest neighbors Algorithm
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=1, metric='euclidean', weights='distance')
knn.fit(X_train, y_train)
```

179] ✓ 0.0s

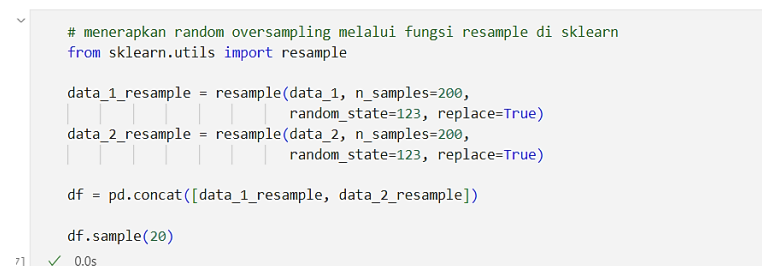
KNeighborsClassifier

KNeighborsClassifier(metric='euclidean', n_neighbors=1, weights='distance')

Gambar 8. Import library

3.2.1 Pemodelan Data Menggunakan Teknik *Random Oversampling* (ROS)

Random Oversampling merupakan suatu teknik transfer data dari kelas minoritas ke data training secara acak. Proses pemberian data diulang sampai jumlah data dari kelas minoritas sama rata dengan jumlah data kelas mayoritas. Setelah melakukan *Data Preprocessing* menggunakan data yang sudah siap dilakukan *Transformasi data* atau normalisasi data sama seperti proses yang dilakukan tanpa menggunakan Teknik ROS, dapat dilihat pada Gambar 9 Selanjutnya melakukan teknik *Random Oversampling* dengan menggunakan perintah *resample* dari modul *sklearn.utils* di Visual Studio Code. Jadi, setelah melakukan perintah tersebut, jumlah data akan mengikuti dengan label data yang terbanyak. Jumlah data jadi 400 data, 200 baris untuk label 0 dan 200 untuk label 1.



```
# menerapkan random oversampling melalui fungsi resample di sklearn
from sklearn.utils import resample

data_1_resample = resample(data_1, n_samples=200,
                           random_state=123, replace=True)
data_2_resample = resample(data_2, n_samples=200,
                           random_state=123, replace=True)

df = pd.concat([data_1_resample, data_2_resample])

df.sample(20)
```

7] ✓ 0.0s

Gambar 9. Syntax *Random oversampling*

Setelah melakukan *preprocessing data*, tahap selanjutnya adalah melakukan proses *Data Splitting*. Pada tahap *Data splitting*, dilakukan pembagian data menjadi 70% Data Training dan 30% Data Testing, sama seperti proses yang dilakukan tanpa menggunakan Teknik ROS. Proses split data dapat dilihat pada gambar 7. Selanjutnya adalah *import* model *K-Nearest Neighbor (KNN)* dengan rumus *euclidean distance* dan menggunakan *extension* dalam *Visual Studio Code* yaitu *Scikit-learn* sama seperti proses yang dilakukan tanpa menggunakan Teknik ROS, seperti Gambar 8.

3.3 Pengujian

Pada pengujian model *K-Nearest Neighbor (KNN)* dilakukan menggunakan data yang sudah di normalisasi, data diuji dengan nilai *k* dari 1 sampai dengan 30, Dari pengujian tersebut akan menghasilkan nilai *K* terbaik yang memiliki jarak paling minimum dan memiliki kemiripan yang tinggi.



```
# Plotting a graph for n_neighbors
from sklearn import metrics
from sklearn.neighbors import KNeighborsClassifier

X_axis = list(range(1, 31))
acc = pd.Series()
x = range(1,31)

for i in list(range(1, 31)):
    knn_model = KNeighborsClassifier(n_neighbors = i)
    knn_model.fit(X_train, y_train)
    prediction = knn_model.predict(X_test)

    acc = pd.concat([acc, pd.Series(metrics.accuracy_score(prediction, y_test))])

plt.plot(X_axis, acc)
plt.xticks(x)
plt.title("Finding best value for n_estimators")
plt.xlabel("n_estimators")
plt.ylabel("Accuracy")
plt.grid()
plt.show()
print('Highest value: ',acc.values.max())
```

8] ✓ 0.7s

Gambar 1. Pengujian nilai *K* terbaik

Pengujian selanjutnya dilakukan menggunakan Metode *Confusion Matrix* dengan menggunakan model algoritma *K-Nearest Neighbor (KNN)* dilakukan perbandingan tanpa teknik *Random oversampling* dan menggunakan teknik *Random oversampling*. Pada Perhitungan *Confusion Matrix* dari data testing pada Visual

Studio Code menggunakan fungsi *confusion_matrix* dari *scikit-learn.metrics* dan di visualisasi menggunakan fungsi *plt* dari *matplotlib.pyplot* dan *sn* dari *seaborn* seperti pada Gambar 11.

```
import matplotlib.pyplot as plt
import seaborn as sn
plt.figure(figsize=(7,5))
sn.heatmap(cm, annot=True)
plt.xlabel('Predicted')
plt.ylabel('Truth')
Text(58.22222222222214, 0.5, 'Truth')
```

Gambar 11. Import *matplotlib.pyplot* & *seaborn*

Kemudian metode yang di evaluasi menggunakan *confusion matrix* sehingga dapat diketahui nilai performa dari masing masing metode klasifikasi. Adapun parameternya untuk mengukur performa tersebut adalah presisi, *recall*, dan akurasi dan *f1-score*.

```
# Evaluating using accuracy_score metric
from sklearn.metrics import accuracy_score
accuracy_knn = accuracy_score(y_test, y_pred_knn)

from sklearn.metrics import confusion_matrix
y_pred = knn.predict(X_test)
cm = confusion_matrix(y_test, y_pred)

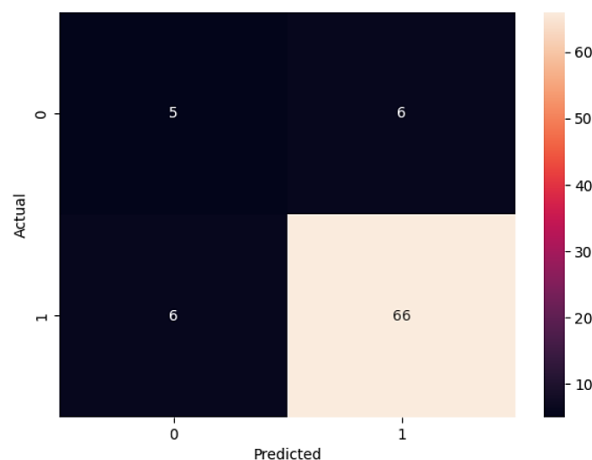
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

Gambar 12. Syntax *Confusion Matrix* dan *Classification Report*

Selanjutnya dengan dari nilai *Confusion Matrix* ini dapat diperoleh nilai akurasi, presisi, *recall* dan *f1-score* dengan *Classification Report*, yang memberikan informasi lebih rinci tentang kinerja model setiap kelas, sehingga membantu dalam evaluasi kemampuan model dalam mengklasifikasikan data untuk setiap kelas secara terpisah. Pada perhitungan *Classification Report* dilakukan pada Visual Studio Code dengan menggunakan fungsi *classification report* dari *scikit learn metrics*, seperti pada Gambar 12.

3.3.1 Pengujian Tanpa Menggunakan Teknik *Random Oversampling (ROS)*

Pada Gambar memperlihatkan Hasil pengujian pada Algoritma *K-Nearest Neighbor (KNN)* tanpa teknik *Random oversampling (ROS)*. Hasil pengujiannya dapat dilihat pada Gambar 13.



Gambar 2. *Confusion Matrix (Non ROS)*

Gambar 13 menunjukan *Confusion Matrix* yang mempresentasikan setiap kelas klasifikasi positif dan negatif. Parameter *True Positif (TP)* merupakan data Positif yang diprediksi benar yaitu 66 data. Parameter *True Negative (TN)* merupakan data negatif yang diprediksi benar yaitu 5 data. Parameter *False Positive (FP)*

merupakan data negative namun diprediksi sebagai data positif yaitu 6 data. Parameter *False Negative* (FN) merupakan data positif namun diprediksi sebagai data negative yaitu 6 data. Dari nilai *Confusion Matrix*, dengan fungsi *classification_report* didapat informasi yang lebih rinci pada Gambar 14 dengan menampilkan *Accuracy*, *Precision*, *Recall* dan *f1-score*.

```
from sklearn.metrics import classification_report
print(classification_report(y_test,knn.predict(X_test)))
```

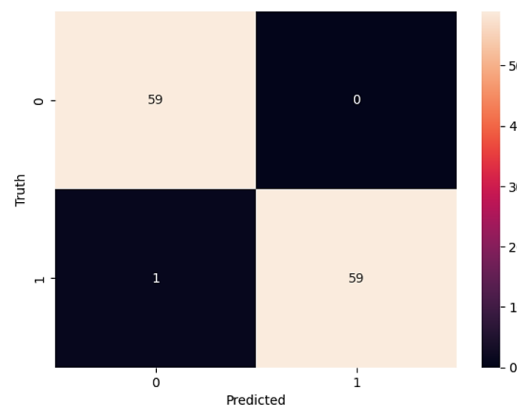
0.0s

	precision	recall	f1-score	support
0	0.45	0.45	0.45	11
1	0.92	0.92	0.92	72
accuracy			0.86	83
macro avg	0.69	0.69	0.69	83
weighted avg	0.86	0.86	0.86	83

Gambar 3. *Classification Report (Non ROS)*

3.3.2 Pengujian Tanpa Menggunakan Teknik *Random Oversampling* (ROS)

Hasil pengujian Algoritma *K-Nearest Neighbor* (KNN) menggunakan teknik *Random oversampling* (ROS) dapat dilihat pada Gambar 15.



Gambar 4. *Confusion Matrix (Apply ROS)*

Gambar 15 menunjukkan *Confusion Matrix* yang mempresentasikan setiap kelas klasifikasi positif dan negatif. Parameter *True Positif* (TP) merupakan data Positif yang diprediksi benar yaitu 59 data. Parameter *True Negative* (TN) merupakan data negatif yang diprediksi benar yaitu 59 data. Parameter *False Positive* (FP) merupakan data negative namun diprediksi sebagai data positif yaitu 0 data. Parameter *False Negative* (FN) merupakan data positif namun diprediksi sebagai data negative yaitu 1 data. Dari nilai *Confusion Matrix*, dengan fungsi *classification_report* didapat informasi yang lebih rinci pada Gambar 16 dengan menampilkan *Accuracy*, *Precision*, *Recall* dan *f1-score*.

```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

0.0s

	precision	recall	f1-score	support
0	0.98	1.00	0.99	59
1	1.00	0.98	0.99	60
accuracy			0.99	119
macro avg	0.99	0.99	0.99	119
weighted avg	0.99	0.99	0.99	119

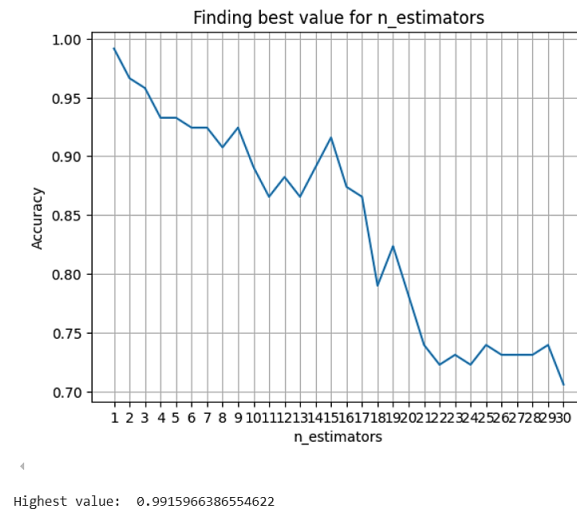
Gambar 5. *Classification Report (Apply ROS)*

3.4 Hasil Penelitian

Hasil penelitian dari model klasifikasi yang dilakukan dengan Algoritma *K-Nearest Neighbors* (KNN) sebagai berikut.

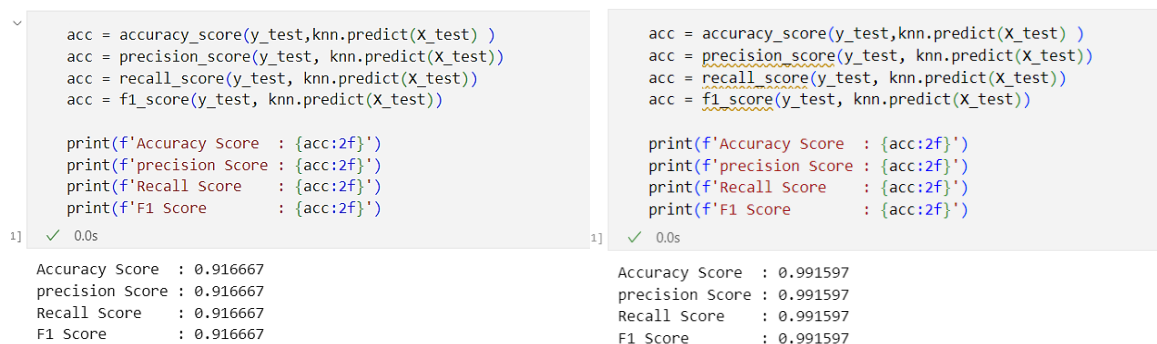
1. Pengujian nilai K terbaik

Pengujian model dilakukan menggunakan data yang sudah di normalisasi, data diuji dengan nilai k dari 1 sampai dengan 30. Dari pengujian yang dilakukan menghasilkan nilai K terbaik ada pada K=1 dengan nilai akurasi sebesar 99,15%. Hasil pengujian Nilai K terbaik dapat dilihat pada Gambar 17.



Gambar 17. Nilai K terbaik

2. Perbandingan Accuracy, Precision, Recall dan F1-Score



Gambar 18. Perbandingan *Accuracy*, *Precision*, *Recall* dan *F1-Score* (a: Tanpa Teknik Ros dan b: Menggunakan Teknik Ros)

Berdasarkan Gambar 18 dilihat dari nilai *Accuracy*, *Precision*, *Recall* dan *F1-Score* menggunakan Dataset yang tidak seimbang menghasilkan nilai *Accuracy*, *Precision*, *Recall* dan *F1-Score* yang rendah dan setelah Data diseimbangi menggunakan Teknik ROS (*Random oversampling*) mengalami peningkatan nilai *Accuracy* menjadi 0.99, *Precision* menjadi 0.99, *Recall* menjadi 0.99 dan *F1-Score* menjadi 0.99. Hasil evaluasi dapat dilihat pada Tabel 2 berikut ini.

Tabel 2. Evaluasi Model Klasifikasi

Algoritma KNN	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
Tanpa Teknik Random Oversampling	0.91	0.91	0.91	0.91
Menggunakan Teknik Random Oversampling	0.99	0.99	0.99	0.99

Jika dilihat pada Tabel 2, model yang dihasilkan dari data yang tidak diseimbangi memiliki nilai *Accuracy*, *Precision*, *Recall* dan *F1-Score* yang lebih rendah dari pada model yang dibuat dengan data yang diseimbangi dengan Teknik *Random Oversampling*. Maka dapat dikatakan kalau Teknik *Random Oversampling* ini dapat meningkatkan performa dari Algoritma *K-Nearest Neighbors* (KNN) karna hasil *Accuracy*, *Precision*, *Recall* dan *F1-Score* lebih tinggi dari hasil sebelumnya.

3.4 Pembahasan Hasil Penelitian

Pada penelitian ini menggunakan model klasifikasi Algoritma *K-Nearest Neighbors* (KNN) dengan bahasa pemrograman *python* dan tools *Visual studio code*. Setelah mendapatkan dataset, kemudian dilakukan tahapan

preprocessing data dengan melakukan normalisasi data. Setelah data dinormalisasi, data diuji dengan nilai k dari 1 sampai dengan 30, Dari pengujian tersebut akan menghasilkan nilai K terbaik. Dari pengujian yang dilakukan menghasilkan nilai K terbaik ada pada $K=1$ dengan nilai akurasi sebesar 99,15%.

Berdasarkan analisis yang sudah dilakukan, dapat disimpulkan bahwa Teknik Random oversampling sangat berpengaruh untuk meningkatkan nilai *Accuracy*, *Precision*, *Recall* dan *F1-Score*. Peningkatan nilai *accuracy* untuk Algoritma *K-Nearest Neighbors* (KNN) sebesar 0.8. Jadi, ini memungkinkan pada masalah klasifikasi dataset Kanker paru paru ini Algoritma *K-Nearest Neighbors* (KNN) menangannya dengan baik. Kemudian menggunakan teknik ROS (*Random oversampling*) menciptakan sampel dataset yang baru untuk kelas minoritas dengan mengubah data interpolasi pada data pelatihan yang baru. Dengan cara ini, jumlah dataset meningkat sehingga membantu menyeimbangi ketidakseimbangan dan memberikan model lebih banyak informasi. Maka dapat disimpulkan Algoritma *K-Nearest Neighbors* (KNN) menggunakan teknik ROS adalah pilihan terbaik untuk klasifikasi dataset kanker paru paru ini.

4. KESIMPULAN

Berdasarkan hasil yang penelitian yang dilakukan oleh penulis, kesimpulan yang didapat yaitu Algoritma *K-Nearest Neighbor* (KNN) dapat dengan baik dalam klasifikasi penyakit kanker paru-paru. Hasil pengujian yang dilakukan diketahui bahwa algoritma *K-Nearest Neighbor* dengan rasio splitting data 70:30 menghasilkan nilai K terbaik ada pada $K=1$ dengan nilai akurasi sebesar 99,15%. Hasil pengujian nilai performa yang digunakan untuk mengetahui klasifikasi penyakit kanker paru paru. Algoritma *K-Nearest Neighbor* (KNN) setelah menyeimbangi data menggunakan teknik ROS (*random oversampling*), mendapatkan nilai performa yang lebih tinggi dibandingkan tanpa menggunakan teknik ROS (*random oversampling*). Hasil pengujian yang dilakukan diketahui bahwa algoritma *K-Nearest Neighbor* (KNN) memiliki nilai *accuracy* 0.99, nilai *precision* 0.99, nilai *recall* 0.99, dan *f1score* 0.99. Berdasarkan hasil dari pengukuran nilai performa *Accuracy*, *Precision*, *Recall* dan *f1-score* menggunakan algoritma *K-Nearest Neighbor* (KNN) dengan teknik ROS (*random oversampling*) dapat disimpulkan bahwa algoritma *K-Nearest Neighbor* (KNN) dapat dengan baik mengklasifikasikan penyakit kanker paru paru.

REFERENCES

- [1] dr. Vincent Lim and dr. Salvirah, "5 Jenis Penyakit Penyebab Kematian Tertinggi di Indonesia," siloamhospitals. Accessed: Mar. 18, 2024. [Online]. Available: <https://www.siloamhospitals.com/en/informasi-siloam/artikel/waspada-5-jenis-penyakit-penyebab-kematian-tertinggi-di-indonesia>
- [2] Pittara, "Kanker Paru paru," alodokter. Accessed: Mar. 18, 2024. [Online]. Available: <https://www.alodokter.com/kanker-paru-paru>
- [3] Rizaty and Monavia Ayu, "Data Persentase Perokok di Indonesia (2015-2023)," dataindonesia. Accessed: Mar. 18, 2024. [Online]. Available: <https://dataindonesia.id/kesehatan/detail/data-persentase-perokok-di-indonesia-20152023>
- [4] J. Homepage *et al.*, "Implementasi Algoritma Decision Tree dan Support Vector Machine untuk Klasifikasi Penyakit Kanker Paru," vol. 3, pp. 15–19, 2023.
- [5] A. Desiani *et al.*, "Perbandingan Klasifikasi Penyakit Kanker Paru-Paru menggunakan Support Vector Machine dan K-Nearest Neighbor," *Jurnal PROCESSOR*, vol. 18, no. 1, Apr. 2023, doi: 10.33998/processor.2023.18.1.700.
- [6] T. Abdi Mangun, O. Nurdiawan, and A. Irma Purnamasari, "LUNG CANCER ANALYSIS USING K-NEAREST NEIGHBOR ALGORITHM," 2023
- [7] W. Ramdhani, D. Bona, R. B. Musyaffa, and C. Rozikin, "Klasifikasi Penyakit Kanker Payudara Menggunakan Algoritma K-Nearest Neighbor," *Jurnal Ilmiah Wahana Pendidikan*, 2022, pp. 445–452, doi: 10.5281/zenodo.6968420.
- [8] A. Amalia *et al.*, "PREDIKSI KUALITAS UDARA MENGGUNAKAN ALGORITMA K-NEAREST NEIGHBOR," 2022
- [9] A. Pratama Agustin and A. Charis Fauzan, "Implementation Of K-Nearest Neighbor With Minkowski Distance For Early Detection Of Covid-19 In CT-Scan Images Of The Lungs Abstrak," 2022. [Online]. Available: <https://ui.adsabs.harvard.edu/abs/2020arXiv200313865Y/abstract>.
- [10] A. Desiani *et al.*, "Perbandingan Klasifikasi Penyakit Kanker Paru-Paru menggunakan Support Vector Machine dan K-Nearest Neighbor," *Jurnal PROCESSOR*, vol. 18, no. 1, Apr. 2023, doi: 10.33998/processor.2023.18.1.700.
- [11] J. Khatib Sulaiman, S. Tegar Kusuma, and T. Bayu Sasongko, "Optimasi K-Nearest Neighbor dengan Grid Search CV pada Prediksi Kanker Paru-Paru," *Indonesian Journal of Computer Science Attribution*, vol. 12, no. 4, p. 2162, 2023
- [12] M. Rahmadiyah and P. Suparman, "PENERAPAN METODE K-NEAREST NEIGHBOUR UNTUK SISTEM PENENTUAN PEMINJAMAN MODAL NASABAH BANK SYARIAH INDONESIA

- CABANG CIKARANG BERBASIS WEBSITE,” *Jurnal informasi dan Komputer*, vol. 10, no. 2, 2022.
- [13] B. B. Tangkere, “Analisis Performa Logistic Regression dan Support Vector Classification untuk Klasifikasi Email Phising,” *Jurnal Ekonomi Manajemen Sistem Informasi (JEMSI)*, vol. Vol. 5, pp. 442–450, 2024, doi: 10.38035/jemsi.v5i4.
- [14] S. Diantika, “PENERAPAN TEKNIK RANDOM OVERSAMPLING UNTUK MENGATASI IMBALANCE CLASS DALAM KLASIFIKASI WEBSITE PHISHING MENGGUNAKAN ALGORITMA LIGHTGBM,” 2023.
- [15] Kabar Harian, “Pengertian Dataset dan Jenis-jenisnya,” Kumparan. Accessed: Mar. 19, 2024. [Online]. Available: <https://kumparan.com/kabar-harian/pengertian-dataset-dan-jenis-jenisnya-1wtM6xNlqpQ/full>
- [16] Sripto, Rr Nurul Rahmanita, and Ajeng Sekar Kirana, “Teknik pre-processing dan classification dalam data science,” binus. Accessed: Mar. 19, 2024. [Online]. Available: <https://mie.binus.ac.id/2022/08/26/teknik-pre-processing-dan-classification-dalam-data-science/>