JUSIKOM PRIMA (Journal of Information Systems and Computer Science Prima)
Vol. 6 No. 2, February 2023 E-ISSN: 2580-2879

OPTIMIZATION OF LUNG CANCER CLASSIFICATION METHOD USING EDA-BASED MACHINE LEARNING

Windania Purba[1] , Sumita Wardani[2], Diana Febrina Lumbantoruan[3], Fransiska Celia Ivo Silalahi[4], Thomas Leo Edison[5]
[1,2,3,4,5]Universitas Prima Indonesia
Jl. Sampul No.4, Medan Petisah, Medan, North Sumatra
E-mail : Windania@unprimdn.ac.id

ABSTRACT- Lung cancer is one of the three deadliest diseases in the world and has rapidly developed. Based on this, researchers conducted research to predict the factors that influence lung cancer. One method to identify this is using data mining methods and classification techniques. Researchers used several popular algorithms in classification to make comparisons of the most accurate algorithms for lung cancer classification. The algorithms used include K-Nearest Neighbor, Random Forest Classifier, Logistic Regression, Linear SVM, Naïve Bayes, Decision Tree, Random Forest, Gradient Boosting, Kernel SVM, and MLPClassifier. The researcher used this algorithm because, in the research that the researcher found on the Kaggle platform, he examined the comparison of the algorithm using the breast cancer dataset. In previous studies, their researchers used SVM, which obtained an accuracy of 96.47%, Neural Networks of 97.06%, and Naïve Bayes with an accuracy of 91.18% to study breast cancer. The difference from previous research is that this study uses several existing algorithms in Machine Learning such as K-Nearest Neighbor, Random Forest Classifier, Logistic Regression, Linear SVM, Naïve Bayes, Decision Tree, Random Forest, Gradient Boosting, Kernel SVM, and MLPClassifier. In addition, this research was conducted to see whether the results of the accuracy of the algorithm that the researchers carried out using the lung cancer dataset had different results.

The results of this study found that the more accurate algorithms were Random Forest and Gradient Boosting with an accuracy value of 100%, whereas in previous studies, it was the same. Still, Gradient Boosting had a higher accuracy value than Random Forest. Then, based on the data used in this study, the most influencing factors in predicting a diagnosis of lung cancer are obesity and coughing up blood. The results of this study found that the more accurate algorithms were Random Forest and Gradient Boosting with an accuracy value of 100%, whereas in previous studies, it was the same. Still, Gradient Boosting had a higher accuracy value than Random Forest. Then, based on the data used in this study, the most influencing factors in predicting a diagnosis of lung cancer are obesity and coughing up blood. The results of this study found that the more accurate algorithms were Random Forest and Gradient Boosting with an accuracy value of 100%, whereas in previous studies, it was the same. Still, Gradient Boosting had a higher accuracy value than Random Forest. Then, based on the data used in this study, the most influencing factors in predicting a diagnosis of lung cancer are obesity and coughing up blood.

Keywords: *Data Mining*, Classification, Lung Cancer, Accuracy

1. INTRODUCTION

The lungs are one of the organs of the respiratory system, which functions as a place for the exchange of carbon dioxide and oxygen in the blood. The problem that often occurs in the respiratory system is polluted air quality, so the inhaled air contains many bacteria that can attack the respiratory system, especially the lungs [1]. One disease that attacks the respiratory system is lung cancer [2].

Lung cancer is one of the three diseases which is the highest cause of death [3]. Several factors influence the rapid development of lung cancer worldwide, including long-term exposure to tobacco smoke, genetic factors, radon gas, and air pollution. [4]

Based on this, researchers conducted research to predict the factors that influence lung cancer. One method to identify this is to use data mining methods. Data mining is a branch of computer science with rapid development [5]. Data mining has been widely applied in various fields, such as science, engineering, and business. An example of the application of Data *mining* is by using machine learning techniques to see tumor behavior in breast cancer patients [6]. Machine learning is a branch of artificial intelligence and computer science that uses data and algorithms to mimic how humans learn to improve machine learning accuracy [7]. In Machine learning, many methods can handle complex decisions [12].

In previous studies, researchers used the SVM algorithm and obtained an accuracy of 96.47%, Neural Networks of 97.06%, and using Naïve Bayes with an accuracy of 91.18% to study breast cancer [8]. Whereas in research [9], the Naïve Bayes algorithm has an accuracy value of 72.61%, Random Forest 68.64%, and Random Tree 70.18%.

In this study, researchers will use several popular classification algorithms to compare which algorithm is more accurate in classifying lung cancer. The algorithms to be used are the K-Nearest Neighbor Algorithm, Random Forest Classifier, Logistic Regression, Linear SVM, Naïve Bayes, Decision Tree, Random Forest, Gradient Boosting, Kernel SVM, and MLP Classifier. Reason from p.s. The use of several algorithms that have been described is due to research [10]. Therefore, these algorithms are also used to see comparisons between these algorithms using breast cancer datasets.

Based on these problems, this study will conduct experiments that aim to compare algorithms based on the distribution of testing data and training data to find out the best results from these algorithms and also to find out which parameters have the most influence in predicting someone being diagnosed with lung cancer.

2. RESEARCH CONTENT

The contents of this research consist of a research methodology that explains the stages carried out in carrying out the research system as well as the results and discussion, which aims to explain and interpret the research results.

2.1 Research Methodology

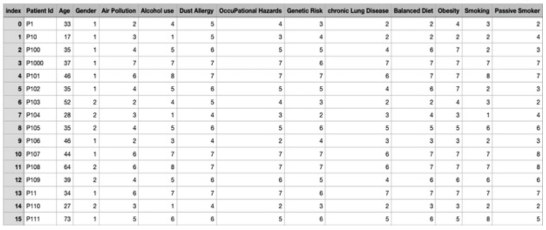
In this research, there are several stages, including:

1. **Preparation**
   Activities must be carried out at this stage to make planning preparations more effective.

2. **Literature Review**
   This stage is carried out to collect information through scientific journals and books.
   This stage is also carried out to identify the factors influencing lung cancer using existing datasets and the algorithms used.

3. **Data collection**
   In this stage, data collection is carried out using a literature study, namely tracing datasets from the Kaggle website [11].

4. **Data Mining Processing**
   Datamining is exploring and finding interesting patterns from large amounts of data. Data sources explored in data mining can come from databases, data warehouses, or other information stores. [5]
   This study's data mining process follows Knowledge Discovery in Database (KDD) steps. The following are the steps that will be carried out, namely:
   1. *Data Acquisition*
   This stage is the process of retrieving data from valid data sources that will be used to create models in machine learning. The data is a dataset found on the Kaggle website about lung cancer.
   Datasets consist of 1000 rows and 25 columns. The dataset can be seen in the image below.



index	Patient Id	Age	Gender	Air Pollution	Alcohol use	Dust Allergy	OccuPational Hazards	Genetic Risk	chronic Lung Disease	Balanced Diet	Obesity	Smoking	Passive Smoker
0	P1	33	1	2	4	5	4	3	2	2	4	3	2
1	P10	17	1	3	1	5	3	4	2	2	2	2	4
2	P100	35	1	4	5	6	5	5	4	6	7	2	3
3	P1000	37	1	7	7	7	7	6	7	7	7	7	7
4	P101	46	1	6	8	7	7	7	6	7	7	8	7
5	P102	35	1	4	5	6	5	5	4	6	7	2	3
6	P103	52	2	2	4	5	4	3	2	2	4	3	2
7	P104	28	2	3	1	4	3	2	3	4	3	1	4
8	P105	35	2	4	5	6	5	6	5	5	5	6	6
9	P106	46	1	2	3	4	2	4	3	3	3	2	3
10	P107	44	1	6	7	7	7	7	6	7	7	7	8
11	P108	64	2	6	8	7	7	7	6	7	7	7	8
12	P109	39	2	4	5	6	6	5	4	6	6	6	6
13	P11	34	1	6	7	7	7	6	7	7	7	7	7
14	P110	27	2	3	1	4	2	3	2	3	3	2	2
15	P111	73	1	5	6	6	5	6	5	6	5	8	5

Figure 2.1 *Lung cancer dataset*
(Source: 2023 research results)

2. *Data Preparation*
   Stage *preparation data* consists of all activities or processes performed to build datasets for the modeling phase, including cleaning data, aggregating data, and transforming data into more useful variables.

3. *Exploratory Data*
   At this stage, it is a data exploration method using simple arithmetic and graphical techniques to summarize observational data.

4. *Data Modeling*
   Modeling is a stage in data science methodology where *data scientists* create a model to answer the problem.

5. *Evaluation*
   Model evaluation-*machine learning* was carried out to see the results of the model used by using training data or testing data taken from the dataset used.

6. **Results and Discussion**
   The discussion at this stage explains the results of exploring the dataset to find parameters that determine the predicted results of lung cancer classification and the results of machine learning processes carried out using algorithms such as K-Nearest Neighbor, Random Forest Classifier, Logistic Regression, Linear SVM, Naïve Bayes, Decision Tree, Random Forest, Gradient Boosting, Kernel SVM and MLPClassifier to find the most optimal method or algorithm for classification.

7. **Conclusions and recommendations**
   Make conclusions from research results and provide suggestions for researchers to be even better.

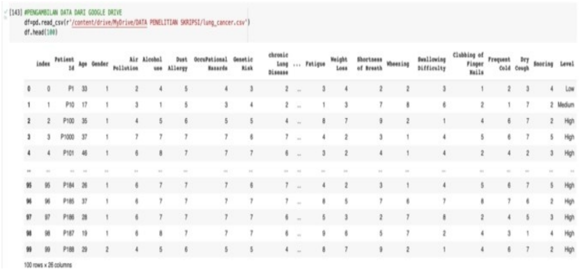
2.2 Results and Discussion

In this chapter, the researcher will explain in more detail the process that the researcher did in this study. Researchers use machine learning with Google collab software and use Python language. In this study, researchers followed a series of data science processes.

The steps that researchers took in conducting this research include:

1. *Data Acquisition*
   Data Acquisition is the process of retrieving data that will be used from useful data sources, where this data will later be used to create models in machine learning [13].
   In this study, secondary data was used about the lung cancer dataset taken from the Kaggle website, where there were 1000 data with the type of file in the form of .csv with 1000 rows and 26 columns. The code snippet used to display the dataset used is

as follows:



```
[143] #PENGAMBILAN DATA DARI GOOGLE DRIVE
      df=pd.read_csv(r'/content/drive/MyDrive/DATA PENELITIAN SKRIPSI/lung_cancer.csv')
      df.head(100)
```

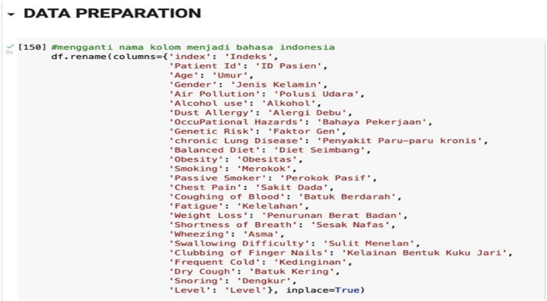
```
    index Patient Age Gender     Air Alcohol    Dust OccuPational Genetic  chronic ... Fatigue Weight Shortness Wheezing Swallowing Clubbing of Frequent  Dry Snoring  Level
               Id               Pollution   use Allergy      Hazards    Risk     Lung                 Loss of Breath           Difficulty      Finger     Cold Cough
                                                                              Disease                                                            Nails
0       0      P1  33      1        2       4       5            4       3        2 ...       3      4         2        2          3           1        2     3       4    Low
1       1     P10  17      1        3       1       5            3       4        2 ...       1      3         7        8          6           2        1     7       2 Medium
2       2    P100  35      1        4       5       6            5       5        4 ...       8      7         9        2          1           4        6     7       2   High
3       3   P1000  37      1        7       7       7            7       6        7 ...       4      2         3        1          4           5        6     7       5   High
4       4    P101  46      1        6       8       7            7       7        6 ...       3      2         4        1          4           2        4     2       3   High
..    ...     ... ...    ...      ...     ...     ...          ...     ...      ... ...     ...    ...       ...      ...        ...         ...      ...   ...     ...    ...
95     95    P184  26      1        6       7       7            7       6        7 ...       4      2         3        1          4           5        6     7       5   High
96     96    P185  37      1        6       7       7            7       7        7 ...       8      5         7        6          7           8        7     6       2   High
97     97    P186  28      1        6       7       7            7       7        6 ...       5      3         2        7          8           2        4     5       3   High
98     98    P187  19      1        6       8       7            7       7        6 ...       9      6         5        7          2           4        3     1       4   High
99     99    P188  29      2        4       5       6            5       5        4 ...       8      7         9        2          1           4        6     7       2   High
100 rows × 26 columns
```

Figure 2.2 Data view
(Source: 2023 research results)

2. *Data Preprocessing*

The next step is to preprocess the data. At this stage, it is carried out to check whether there is data that has missing values or duplicated data, normalizes the data, and also transforms the data into a form that is suitable for the mining process.

1.Rename column

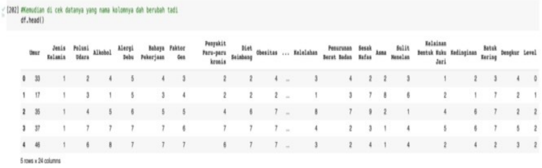


DATA PREPARATION

```
[150] #mengganti nama kolom menjadi bahasa indonesia
      df.rename(columns={'index': 'Indeks',
                         'Patient Id': 'ID Pasien',
                         'Age': 'Umur',
                         'Gender': 'Jenis Kelamin',
                         'Air Pollution': 'Polusi Udara',
                         'Alcohol use': 'Alkohol',
                         'Dust Allergy': 'Alergi Debu',
                         'OccuPational Hazards': 'Bahaya Pekerjaan',
                         'Genetic Risk': 'Faktor Gen',
                         'chronic Lung Disease': 'Penyakit Paru-paru kronis',
                         'Balanced Diet': 'Diet Seimbang',
                         'Obesity': 'Obesitas',
                         'Smoking': 'Merokok',
                         'Passive Smoker': 'Perokok Pasif',
                         'Chest Pain': 'Sakit Dada',
                         'Coughing of Blood': 'Batuk Berdarah',
                         'Fatigue': 'Kelelahan',
                         'Weight Loss': 'Penurunan Berat Badan',
                         'Shortness of Breath': 'Sesak Nafas',
                         'Wheezing': 'Asma',
                         'Swallowing Difficulty': 'Sulit Menelan',
                         'Clubbing of Finger Nails': 'Kelainan Bentuk Kuku Jari',
                         'Frequent Cold': 'Kedinginan',
                         'Dry Cough': 'Batuk Kering',
                         'Snoring': 'Dengkur',
                         'Level': 'Level'}, inplace=True)
```

Figure 2. 3 Rename column
(Source: 2023 research results)

And after changing the column names, the dataset will look like the image below:

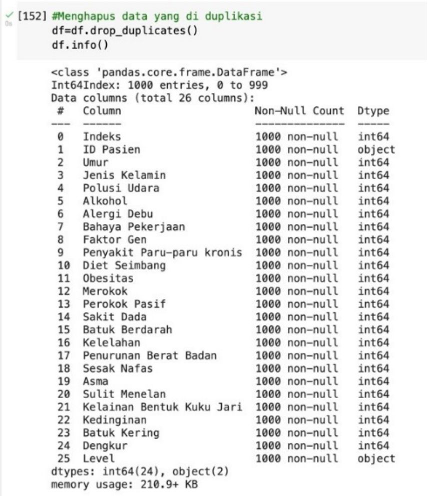


```
[202] #Kemudian di cek datanya yang nama kolomnya dah berubah tadi
      df.head()
```

```
  Umur   Jenis Polusi Alkohol Alergi    Bahaya Faktor  Penyakit    Diet Obesitas ... Kelelahan   Penurunan Sesak Asma  Sulit    Kelainan Kedinginan  Batuk Dengkur Level
       Kelamin  Udara          Debu Pekerjaan    Gen Paru-paru Seimbang                       Berat Badan Nafas       Menelan Bentuk Kuku              Kering
                                                         kronis                                                                    Jari
0   33       1      2       4     5         4      3         2       2        4 ...         3           4     2    2       3           1          2      3       4     0
1   17       1      3       1     5         3      4         2       2        2 ...         1           3     7    8       6           2          1      7       2     1
2   35       1      4       5     6         5      5         4       6        7 ...         8           7     9    2       1           4          6      7       2     2
3   37       1      7       7     7         7      6         7       7        7 ...         4           2     3    1       4           5          6      7       5     2
4   46       1      6       8     7         7      7         6       7        7 ...         3           2     4    1       4           2          4      2       3     2
5 rows × 24 columns
```

Figure 2.4 Display after the column name changed
(Source: 2023 research results)

2.Displays duplicate data and data types.



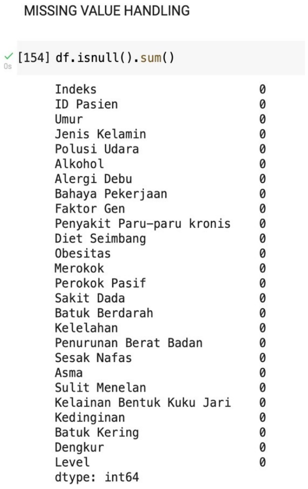
```
[152] #Menghapus data yang di duplikasi
      df=df.drop_duplicates()
      df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1000 entries, 0 to 999
Data columns (total 26 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Indeks                     1000 non-null   int64
 1   ID Pasien                  1000 non-null   object
 2   Umur                       1000 non-null   int64
 3   Jenis Kelamin              1000 non-null   int64
 4   Polusi Udara               1000 non-null   int64
 5   Alkohol                    1000 non-null   int64
 6   Alergi Debu                1000 non-null   int64
 7   Bahaya Pekerjaan           1000 non-null   int64
 8   Faktor Gen                 1000 non-null   int64
 9   Penyakit Paru-paru kronis  1000 non-null   int64
 10  Diet Seimbang              1000 non-null   int64
 11  Obesitas                   1000 non-null   int64
 12  Merokok                    1000 non-null   int64
 13  Perokok Pasif              1000 non-null   int64
 14  Sakit Dada                 1000 non-null   int64
 15  Batuk Berdarah             1000 non-null   int64
 16  Kelelahan                  1000 non-null   int64
 17  Penurunan Berat Badan      1000 non-null   int64
 18  Sesak Nafas                1000 non-null   int64
 19  Asma                       1000 non-null   int64
 20  Sulit Menelan              1000 non-null   int64
 21  Kelainan Bentuk Kuku Jari  1000 non-null   int64
 22  Kedinginan                 1000 non-null   int64
 23  Batuk Kering               1000 non-null   int64
 24  Dengkur                    1000 non-null   int64
 25  Level                      1000 non-null   object
dtypes: int64(24), object(2)
memory usage: 210.9+ KB
```

Figure 2.4 Duplicate data view
(Source: 2023 research results)

Figure 2.4 explains that there are 1000 data, which means that the data has no duplicates and the data types in the dataset are object and int64.

3.Displays missing values



MISSING VALUE HANDLING

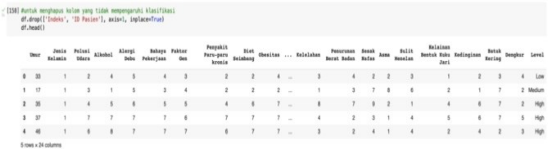
```
[154] df.isnull().sum()
```

```
Indeks                       0
ID Pasien                    0
Umur                         0
Jenis Kelamin                0
Polusi Udara                 0
Alkohol                      0
Alergi Debu                  0
Bahaya Pekerjaan             0
Faktor Gen                   0
Penyakit Paru-paru kronis    0
Diet Seimbang                0
Obesitas                     0
Merokok                      0
Perokok Pasif                0
Sakit Dada                   0
Batuk Berdarah               0
Kelelahan                    0
Penurunan Berat Badan        0
Sesak Nafas                  0
Asma                         0
Sulit Menelan                0
Kelainan Bentuk Kuku Jari    0
Kedinginan                   0
Batuk Kering                 0
Dengkur                      0
Level                        0
dtype: int64
```

Figure 2. 5 Displays missing values
(Source: 2023 research results)

The appearance of Figure 2.5 explains that there is no data that has a missing value.

4.Delete the column Which does not affect the classification

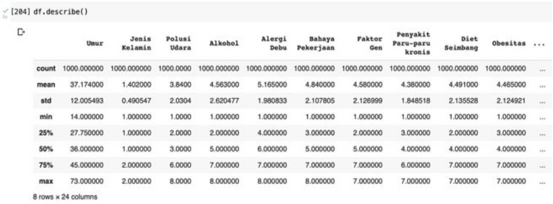


```
[158] #untuk menghapus kolom yang tidak mempengaruhi klasifikasi
      df.drop(['Indeks', 'ID Pasien'], axis=1, inplace=True)
      df.head()
```

```
  Umur   Jenis Polusi Alkohol Alergi    Bahaya Faktor  Penyakit    Diet Obesitas ... Kelelahan   Penurunan Sesak Asma  Sulit    Kelainan Kedinginan  Batuk Dengkur  Level
       Kelamin  Udara          Debu Pekerjaan    Gen Paru-paru Seimbang                       Berat Badan Nafas       Menelan Bentuk Kuku              Kering
                                                         kronis                                                                    Jari
0   33       1      2       4     5         4      3         2       2        4 ...         3           4     2    2       3           1          2      3       4    Low
1   17       1      3       1     5         3      4         2       2        2 ...         1           3     7    8       6           2          1      7       2 Medium
2   35       1      4       5     6         5      5         4       6        7 ...         8           7     9    2       1           4          6      7       2   High
3   37       1      7       7     7         7      6         7       7        7 ...         4           2     3    1       4           5          6      7       5   High
4   46       1      6       8     7         7      7         6       7        7 ...         3           2     4    1       4           2          4      2       3   High
5 rows × 24 columns
```

Figure 2. 6 Dataset view after deleting the column
(Source: 2023 research results)

5. Showing spread data descriptive statistics of the dataset



```
[204] df.describe()
```

	Umur	Jenis Kelamin	Polusi Udara	Alkohol	Alergi Debu	Bahaya Pekerjaan	Faktor Gen	Penyakit Paru-paru kronis	Diet Seimbang	Obesitas	...
count	1000.000000	1000.000000	1000.0000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	...
mean	37.174000	1.402000	3.8400	4.563000	5.165000	4.840000	4.580000	4.380000	4.491000	4.465000	...
std	12.005493	0.490547	2.0304	2.620477	1.980833	2.107805	2.126999	1.848518	2.135528	2.124921	...
min	14.000000	1.000000	1.0000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	...
25%	27.750000	1.000000	2.0000	2.000000	4.000000	3.000000	2.000000	3.000000	2.000000	3.000000	...
50%	36.000000	1.000000	3.0000	5.000000	6.000000	5.000000	5.000000	4.000000	4.000000	4.000000	...
75%	45.000000	2.000000	6.0000	7.000000	7.000000	7.000000	7.000000	6.000000	7.000000	7.000000	...
max	73.000000	2.000000	8.0000	8.000000	8.000000	8.000000	7.000000	7.000000	7.000000	7.000000	...

8 rows × 24 columns

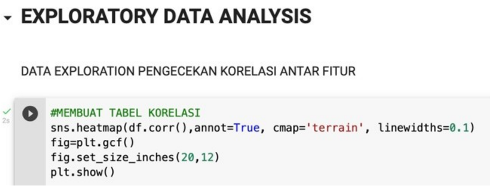
Figure 2.7 Distribution of descriptive data from the dataset
(Source: 2023 research results)

3. *Exploratory Data Analysis*

Exploratory Data Analysis(EDA) is a critical process in conducting initial investigations on data to find patterns, and anomalies, test hypotheses, and being able to check assumptions with the help of summary statistics and then graphical representation (visualization) [14].

Exploratory Data analysis allows analysts to understand the contents of the data used, starting from distribution, frequency, correlation, and others. This process is an essential process in data analysis because by doing Exploratory Data Analysis, the user will be able to save more time in the data analysis process, be able to find out some errors in the data such as missing values, outliers, duplications, encodings, noisy data, incomplete data, and others.

In this process, the researcher checks the correlation between the features in the dataset that the researcher uses. The code snippet used is as follows:



EXPLORATORY DATA ANALYSIS

DATA EXPLORATION PENGECEKAN KORELASI ANTAR FITUR

```
#MEMBUAT TABEL KORELASI
sns.heatmap(df.corr(),annot=True, cmap='terrain', linewidths=0.1)
fig=plt.gcf()
fig.set_size_inches(20,12)
plt.show()
```

Figure 2.8 The program list displays the correlation table
(Source: 2023 research results)

So the output of the code snippet in Figure 2.8 is as follows:

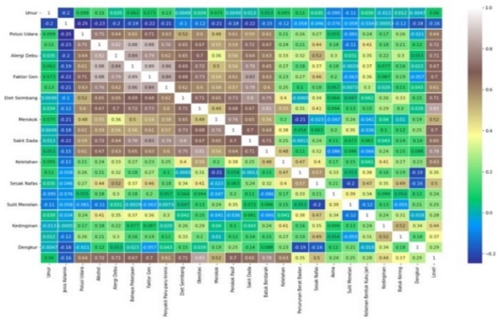
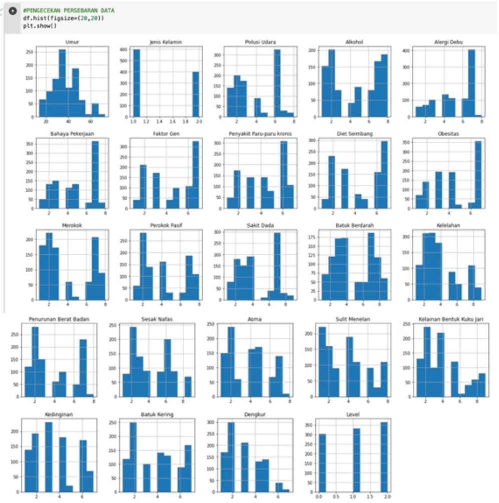


Figure 2. 9 Correlation between columns
(Source: 2023 Research Results)

The output results from the image above answer one of this study's objectives, which is to find out what factors or parameters most influence a person to be diagnosed with lung cancer.

So from the results of the correlation visualization above, it can be concluded that the most influential parameter for predicting a person will get lung cancer at a certain level is obesity, with the most considerable correlation value of 0.83, followed by coughing up blood with a correlation value of 0.78 then alcohol with a correlation value of 0.72, then dust allergies and a balanced diet which has the same correlation value of 0.71 and finally the gene factor and passive smoking which also has the same correlation value of 0.7.

Then the researcher also checked the distribution of data in each column of the dataset. The code snippet and its output is as follows:



```
#PENGECEKAN PERSEBARAN DATA
df.hist(figsize=(20,20))
plt.show()
```

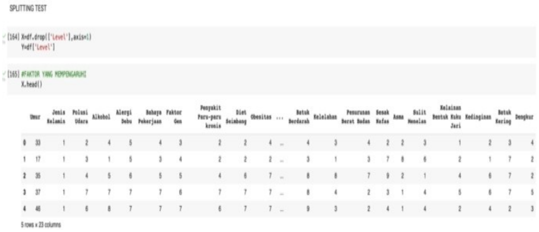
Figure 2. 10 Data distribution chart
(Source: 2023 Research Results)

The next step is to see the results of the accuracy of the classification algorithm used in this study. Which results of each accuracy of each of these algorithms will answer the first question of the problem formulation in this study, namely, which algorithm is more accurate in the classification of lung cancer?

4. *Data Modeling*

This stage will determine the outcome *accuracy* of each model or algorithm used in this study. But seeing before the seeing *accuracy* of each of its algorithms, first, perform the splitting test stage. The splitting test is carried out by dividing the dataset into two parts, namely the part that will be used for training data and testing data with a certain percentage. In this study, the data was divided into 800 training data and 200 test data where the ratio of the test data was 0.2% or 20%, the recommended rate during the splitting test.

However, before entering into the splitting test, the researcher first deleted the column that is not a factor or parameter in this classification, and in this study, what is not a parameter is the "Level" column because this column is the final result of the prediction or determining the Level of this classification. The code snippet to delete this Level column is as follows:

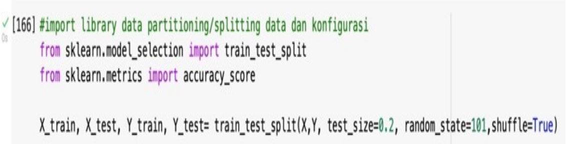


SPLITTING TEST

```
[164] X=df.drop(['Level'],axis=1)
      Y=df['Level']

[165] #FAKTOR YANG MEMPENGARUHI
      X.head()

                Jenis Polusi         Alergi    Bahaya Faktor  Penyakit     Diet                   Batuk           Penurunan  Sesak        Sulit      Kelainan             Batuk
     Umur Kelamin  Udara Alkohol    Debu Pekerjaan    Gen Paru-paru  Seimbang Obesitas ...  Berdarah Kelelahan Berat Badan Nafas Asma Menelan Bentuk Kuku Kedinginan Kering Dengkur
                                                              kronis                                                                                Jari
  0   33       1      2       4       5         4      3         2        2        4 ...         4         3           4     2    2       3           1          2      3       4
  1   17       1      3       1       5         3      4         2        2        2 ...         3         1           3     7    8       6           2          1      7       2
  2   35       1      4       5       6         5      5         4        6        7 ...         8         8           7     9    2       1           4          6      7       2
  3   37       1      7       7       7         7      6         7        7        7 ...         8         4           2     3    1       4           5          6      7       5
  4   46       1      6       8       7         7      7         6        7        7 ...         9         3           2     4    1       4           2          4      2       3

5 rows x 23 columns
```

Figure 2. 11 Display data after deleting the level column
(Source: 2023 research results)

After that, we enter the splitting test, and the code snippet for this splitting test is as follows:

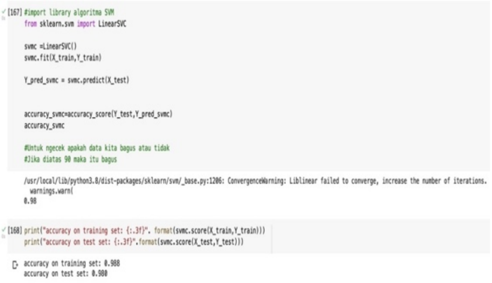


```
[166] #import library data partitioning/splitting data dan konfigurasi
      from sklearn.model_selection import train_test_split
      from sklearn.metrics import accuracy_score

      X_train, X_test, Y_train, Y_test= train_test_split(X,Y, test_size=0.2, random_state=101,shuffle=True)
```

Figure 2. 12 Program list for split test
(Source: 2023 research results)

After doing the splitting test and creating the test and training variables, the next step is for the researcher to check the accuracy of each algorithm. The results of the accuracy of each popular algorithm that researchers use include the following:

a. SVM Linear Algorithm
The accuracy results of this algorithm are as follows:

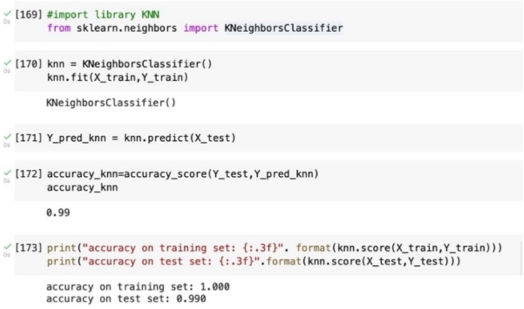


```
[167] #import library algoritma SVM
      from sklearn.svm import LinearSVC

      svmc =LinearSVC()
      svmc.fit(X_train,Y_train)

      Y_pred_svmc = svmc.predict(X_test)


      accuracy_svmc=accuracy_score(Y_test,Y_pred_svmc)
      accuracy_svmc

      #Untuk ngecek apakah data kita bagus atau tidak
      #Jika diatas 90 maka itu bagus

      /usr/local/lib/python3.8/dist-packages/sklearn/svm/_base.py:1206: ConvergenceWarning: Liblinear failed to converge, increase the number of iterations.
        warnings.warn(
      0.98

[168] print("accuracy on training set: {:.3f}". format(svmc.score(X_train,Y_train)))
      print("accuracy on test set: {:.3f}".format(svmc.score(X_test,Y_test)))

      accuracy on training set: 0.988
      accuracy on test set: 0.980
```

Figure 2. 13 Accuracy Linear SVM
(Source: 2023 research results)

From Figure 2.13, the results of the accuracy score are 0.98 or 98%, which is categorized as good. To see if the dataset used is good and the algorithm used is good, that is when the accuracy results are above 90 percent. On the other hand, if the accuracy results are below 90 percent, the algorithm is likely, not good, and the data could be better. Therefore to improve the accuracy, results must be optimized. However, because this research is to see comparisons, the researchers still need to carry out the optimization process.

b. K-Nearest Neighbor Algorithm
The accuracy results of this algorithm are as follows:

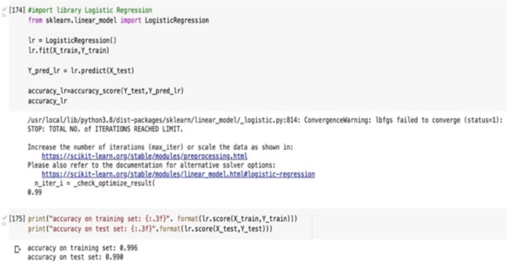


```
[169] #import library KNN
      from sklearn.neighbors import KNeighborsClassifier

[170] knn = KNeighborsClassifier()
      knn.fit(X_train,Y_train)

      KNeighborsClassifier()

[171] Y_pred_knn = knn.predict(X_test)

[172] accuracy_knn=accuracy_score(Y_test,Y_pred_knn)
      accuracy_knn

      0.99

[173] print("accuracy on training set: {:.3f}". format(knn.score(X_train,Y_train)))
      print("accuracy on test set: {:.3f}".format(knn.score(X_test,Y_test)))

      accuracy on training set: 1.000
      accuracy on test set: 0.990
```

Figure 2. 14 Accuracy K-Nearest Neighbor
(Source: 2023 research results)

From Figure 2.14, the results of the accuracy score are 0.99 or 99%. The accuracy results using the KNN algorithm are better than the previous algorithm, namely the Linear SVM algorithm.

c. Logistic Regression Algorithm
The accuracy results of this algorithm are as follows:

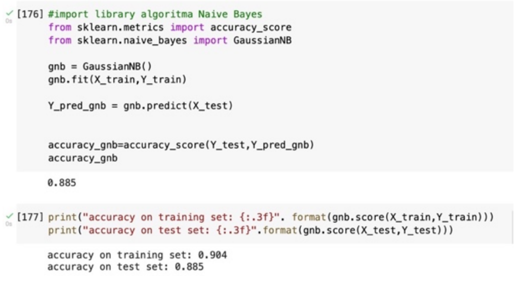


```
[174] #import library Logistic Regression
      from sklearn.linear_model import LogisticRegression

      lr = LogisticRegression()
      lr.fit(X_train,Y_train)

      Y_pred_lr = lr.predict(X_test)

      accuracy_lr=accuracy_score(Y_test,Y_pred_lr)
      accuracy_lr

      /usr/local/lib/python3.8/dist-packages/sklearn/linear_model/_logistic.py:814: ConvergenceWarning: lbfgs failed to converge (status=1):
      STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

      Increase the number of iterations (max_iter) or scale the data as shown in:
          https://scikit-learn.org/stable/modules/preprocessing.html
      Please also refer to the documentation for alternative solver options:
          https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
        n_iter_i = _check_optimize_result(
      0.99

[175] print("accuracy on training set: {:.3f}". format(lr.score(X_train,Y_train)))
      print("accuracy on test set: {:.3f}".format(lr.score(X_test,Y_test)))

      accuracy on training set: 0.996
      accuracy on test set: 0.990
```

Figure 2. 15 Accuracy Logistic Regression
(Source: 2023 research results)

From Figure 2.15, the results of the accuracy score are 0.99 or 99%. The results of the accuracy of the logistic regression algorithm are the same as the KNN algorithm. Still, the difference is the accuracy of the training data, which in the KNN algorithm is greater than the logistic regression.

d. Naïve Bayes Algorithm
The accuracy results of this algorithm are as follows:

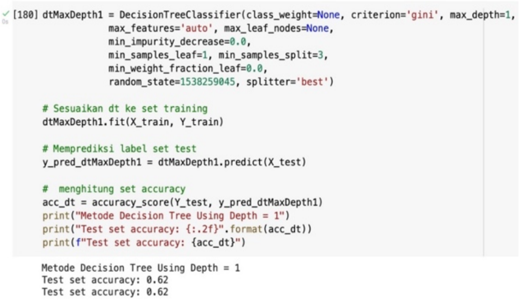


```
[176] #import library algoritma Naive Bayes
      from sklearn.metrics import accuracy_score
      from sklearn.naive_bayes import GaussianNB

      gnb = GaussianNB()
      gnb.fit(X_train,Y_train)

      Y_pred_gnb = gnb.predict(X_test)


      accuracy_gnb=accuracy_score(Y_test,Y_pred_gnb)
      accuracy_gnb

      0.885

[177] print("accuracy on training set: {:.3f}". format(gnb.score(X_train,Y_train)))
      print("accuracy on test set: {:.3f}".format(gnb.score(X_test,Y_test)))

      accuracy on training set: 0.904
      accuracy on test set: 0.885
```

Figure 2. 16 Accuracy Naïve Bayes
(Source: 2023 research results)

From Figure 2.16, the results of the accuracy score are 0.88 or 88%. The accuracy results are below 90%, which is not good, and to improve it, optimization should be done, but because this research is comparative, the researchers should have optimized this algorithm.

e. Decision Tree Algorithm
In this algorithm, the researcher checks accuracy with max depth = 1. The smaller the max depth, the smaller the accuracy results. The results of the accuracy of the max depth = 1 algorithm are as follows:

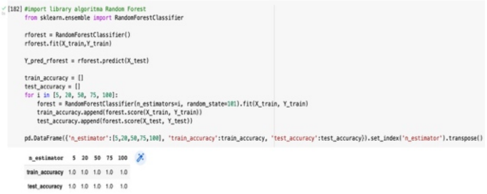


```
[180] dtMaxDepth1 = DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=1,
                  max_features='auto', max_leaf_nodes=None,
                  min_impurity_decrease=0.0,
                  min_samples_leaf=1, min_samples_split=3,
                  min_weight_fraction_leaf=0.0,
                  random_state=1538259045, splitter='best')

      # Sesuaikan dt ke set training
      dtMaxDepth1.fit(X_train, Y_train)

      # Memprediksi label set test
      y_pred_dtMaxDepth1 = dtMaxDepth1.predict(X_test)

      #  menghitung set accuracy
      acc_dt = accuracy_score(Y_test, y_pred_dtMaxDepth1)
      print("Metode Decision Tree Using Depth = 1")
      print("Test set accuracy: {:.2f}".format(acc_dt))
      print(f"Test set accuracy: {acc_dt}")

      Metode Decision Tree Using Depth = 1
      Test set accuracy: 0.62
      Test set accuracy: 0.62
```

Figure 2. 17 Accuracy Decision Tree Max Depth = 1
(Source: 2023 research results)

In Figure 2.17, the accuracy results are even far below 90 percent, which is only 0.62 or 62%, which could be better.

f. Random Forest Algorithm
The accuracy results of this algorithm are as follows:

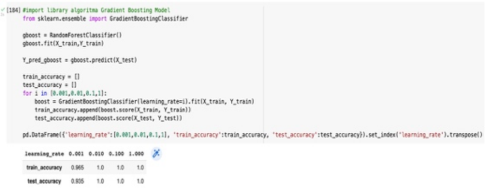


```
[182] #import library algoritma Random Forest
      from sklearn.ensemble import RandomForestClassifier

      rforest = RandomForestClassifier()
      rforest.fit(X_train,Y_train)

      Y_pred_rforest = rforest.predict(X_test)

      train_accuracy = []
      test_accuracy = []
      for i in [5, 20, 50, 75, 100]:
          forest = RandomForestClassifier(n_estimators=i, random_state=101).fit(X_train, Y_train)
          train_accuracy.append(forest.score(X_train, Y_train))
          test_accuracy.append(forest.score(X_test, Y_test))

      pd.DataFrame({'n_estimator':[5,20,50,75,100], 'train_accuracy':train_accuracy, 'test_accuracy':test_accuracy}).set_index('n_estimator').transpose()

      n_estimator      5    20   50   75   100
      train_accuracy  1.0  1.0  1.0  1.0  1.0
      test_accuracy   1.0  1.0  1.0  1.0  1.0
```

Figure 2. 18 *Accuracy Random Forest*
(Source: 2023 research results)

Figure 2.18 uses 5 estimators, namely 5, 20, 50, 75, and 100. The accuracy results of each estimator are the same as 1 or 100%, which is very good because even though the estimators are different, the results are still 100%.

g. Gradient Boosting Algorithm
The accuracy results of this algorithm are as follows:

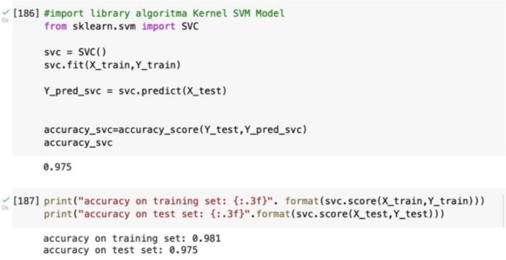


```
[184] #import library algoritma Gradient Boosting Model
      from sklearn.ensemble import GradientBoostingClassifier

      gboost = RandomForestClassifier()
      gboost.fit(X_train,Y_train)

      Y_pred_gboost = gboost.predict(X_test)

      train_accuracy = []
      test_accuracy = []
      for i in [0.001,0.01,0.1,1]:
          boost = GradientBoostingClassifier(learning_rate=i).fit(X_train, Y_train)
          train_accuracy.append(boost.score(X_train, Y_train))
          test_accuracy.append(boost.score(X_test, Y_test))

      pd.DataFrame({'learning_rate':[0.001,0.01,0.1,1], 'train_accuracy':train_accuracy, 'test_accuracy':test_accuracy}).set_index('learning_rate').transpose()

      learning_rate   0.001  0.010  0.100  1.000
      train_accuracy  0.965    1.0    1.0    1.0
      test_accuracy   0.935    1.0    1.0    1.0
```

Figure 2. 19 *Accuracy Gradient Boosting*
(Source: 2023 research results)

Figure 2.19 uses 4 learning rates, namely 0.001, 0.010, 0.100, and 1.000. The accuracy results for each learning rate are only one different, namely at a learning rate of 0.001, which is 0.93 or 93 percent, while based on the other learning rates, the same is equal to 1 or 100%.

h. AlgorithmSVC *kernels*
Results*accuracy*this algorithm is as follows:

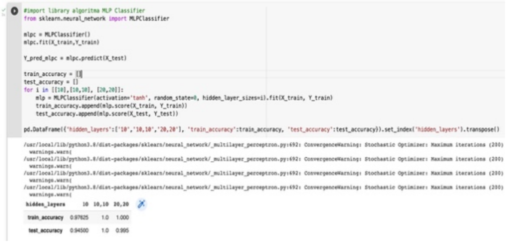


```
[186] #import library algoritma Kernel SVM Model
      from sklearn.svm import SVC

      svc = SVC()
      svc.fit(X_train,Y_train)

      Y_pred_svc = svc.predict(X_test)


      accuracy_svc=accuracy_score(Y_test,Y_pred_svc)
      accuracy_svc

      0.975

[187] print("accuracy on training set: {:.3f}". format(svc.score(X_train,Y_train)))
      print("accuracy on test set: {:.3f}".format(svc.score(X_test,Y_test)))

      accuracy on training set: 0.981
      accuracy on test set: 0.975
```

Figure 2. 20 *SVC Kernel Accuracy*
(Source: 2023 research results)

In Figure 2.20, the results of the accuracy score are 0.97 or 97%, which is categorized as good.

i. MLPClassifier Algorithm
The accuracy results of this algorithm are as follows:



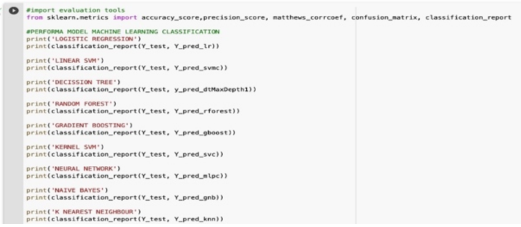
```
#import library algoritma MLP Classifier
from sklearn.neural_network import MLPClassifier

mlpc = MLPClassifier()
mlpc.fit(X_train,Y_train)

Y_pred_mlpc = mlpc.predict(X_test)

train_accuracy = []
test_accuracy = []
for i in [[10],[10,10], [20,20]]:
    mlp = MLPClassifier(activation='tanh', random_state=0, hidden_layer_sizes=i).fit(X_train, Y_train)
    train_accuracy.append(mlp.score(X_train, Y_train))
    test_accuracy.append(mlp.score(X_test, Y_test))

pd.DataFrame({'hidden_layers':['10','10,10','20,20'], 'train_accuracy':train_accuracy, 'test_accuracy':test_accuracy}).set_index('hidden_layers').transpose()

/usr/local/lib/python3.8/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:692: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200)
  warnings.warn(
/usr/local/lib/python3.8/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:692: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200)
  warnings.warn(
/usr/local/lib/python3.8/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:692: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200)
  warnings.warn(
/usr/local/lib/python3.8/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:692: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200)
  warnings.warn(
hidden_layers        10  10,10  20,20
train_accuracy  0.97625    1.0  1.000
test_accuracy   0.94500    1.0  0.995
```

Figure 2. 21 *accuracyMLPClassifier*
(Source: 2023 research results)

Figure 2.21 uses 3 hidden layers, namely 10, 10.10, and 20.20. The accuracy results of each hidden layer are different. Hidden layer 10 results in an accuracy of 0.94 or 94%, hidden layer 10.10 results in an accuracy of 1 or 100%, and hidden layer 20.20 results in an accuracy of 0.99 or 99%. The highest precision of the three hidden layers above is the hidden layer 10.10.

5. *Evaluation*

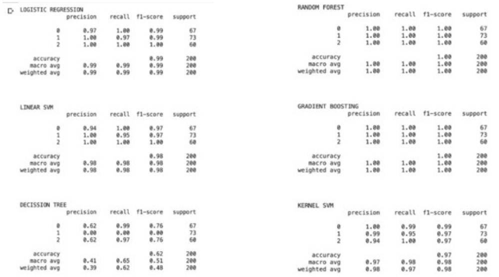
At this stage, we can also determine which model is the most optimal and see how much precision, f1 score, and recall each model or algorithm used. The results of precision, f1 score, and recall will later determine the accuracy of each model or algorithm. But first, before looking at the accuracy, precision, and recall values, import evaluation tools are carried out first, as shown below:



```
#import evaluation tools
from sklearn.metrics import accuracy_score,precision_score, matthews_corrcoef, confusion_matrix, classification_report

#PERFORMA MODEL MACHINE LEARNING CLASSIFICATION
print('LOGISTIC REGRESSION')
print(classification_report(Y_test, Y_pred_lr))

print('LINEAR SVM')
print(classification_report(Y_test, Y_pred_svmc))

print('DECISSION TREE')
print(classification_report(Y_test, y_pred_dtMaxDepth1))

print('RANDOM FOREST')
print(classification_report(Y_test, Y_pred_rforest))

print('GRADIENT BOOSTING')
print(classification_report(Y_test, Y_pred_gboost))

print('KERNEL SVM')
print(classification_report(Y_test, Y_pred_svc))

print('NEURAL NETWORK')
print(classification_report(Y_test, Y_pred_mlpc))

print('NAIVE BAYES')
print(classification_report(Y_test, Y_pred_gnb))

print('K NEAREST NEIGHBOUR')
print(classification_report(Y_test, Y_pred_knn))
```

Figure 2. 22 *Import library to display precision, recall, f1-score, and accuracy results*
(Source: 2023 research results)

Then, we will see the result of the above code snippet as follows:

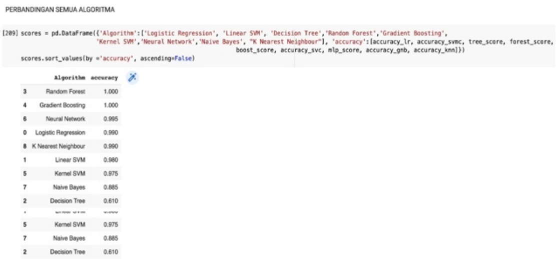


```
LOGISTIC REGRESSION
              precision    recall  f1-score   support

           0       0.97      1.00      0.99        67
           1       1.00      0.97      0.99        73
           2       1.00      1.00      1.00        60

    accuracy                           0.99       200
   macro avg       0.99      0.99      0.99       200
weighted avg       0.99      0.99      0.99       200

LINEAR SVM
              precision    recall  f1-score   support

           0       0.94      1.00      0.97        67
           1       1.00      0.95      0.97        73
           2       1.00      1.00      1.00        60

    accuracy                           0.98       200
   macro avg       0.98      0.98      0.98       200
weighted avg       0.98      0.98      0.98       200

DECISSION TREE
              precision    recall  f1-score   support

           0       0.62      0.99      0.76        67
           1       0.00      0.00      0.00        73
           2       0.62      0.97      0.76        60

    accuracy                           0.62       200
   macro avg       0.41      0.65      0.51       200
weighted avg       0.39      0.62      0.48       200
```

```
RANDOM FOREST
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        67
           1       1.00      1.00      1.00        73
           2       1.00      1.00      1.00        60

    accuracy                           1.00       200
   macro avg       1.00      1.00      1.00       200
weighted avg       1.00      1.00      1.00       200

GRADIENT BOOSTING
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        67
           1       1.00      1.00      1.00        73
           2       1.00      1.00      1.00        60

    accuracy                           1.00       200
   macro avg       1.00      1.00      1.00       200
weighted avg       1.00      1.00      1.00       200

KERNEL SVM
              precision    recall  f1-score   support

           0       1.00      0.99      0.99        67
           1       0.99      0.95      0.97        73
           2       0.94      1.00      0.97        60

    accuracy                           0.97       200
   macro avg       0.97      0.98      0.98       200
weighted avg       0.98      0.97      0.98       200
```

Figure 2. 23 Value results *accuracy, precision, and recall*
(Source: 2023 research results)

If sorted, the accuracy results of each algorithm or model used are as follows:



PERBANDINGAN SEMUA ALGORITMA

```
[209] scores = pd.DataFrame({'Algorithm':['Logistic Regression', 'Linear SVM', 'Decision Tree','Random Forest','Gradient Boosting',
                         'Kernel SVM','Neural Network','Naive Bayes', "K Nearest Neighbour"], 'accuracy':[accuracy_lr, accuracy_svmc, tree_score, forest_score,
                                                        boost_score, accuracy_svc, mlp_score, accuracy_gnb, accuracy_knn]})
      scores.sort_values(by ='accuracy', ascending=False)
```

	Algorithm	accuracy
3	Random Forest	1.000
4	Gradient Boosting	1.000
6	Neural Network	0.995
0	Logistic Regression	0.990
8	K Nearest Neighbour	0.990
1	Linear SVM	0.980
5	Kernel SVM	0.975
7	Naive Bayes	0.885
2	Decision Tree	0.610
[illegible]	[illegible]	[illegible]
5	Kernel SVM	0.975
7	Naive Bayes	0.885
2	Decision Tree	0.610

Figure 2. 24 Results of comparison of models or algorithms
(Source: 2023 research results)

3. CONCLUSION

The conclusions from this study are as follows:

1. That the most optimal model or algorithm is Random Forest and Gradient Boosting with an accuracy percentage of 100%.
2. Based on the results of exploratory data analysis, the most influential parameter in predicting the classification of lung cancer is obesity, with the largest correlation value of 0.83, followed by coughing up blood with a correlation value of 0.78, because when the value is 6, the patient immediately enters to the high-risk category for lung cancer. However, for other parameters, if the value is at number 7, it is still included in the medium-risk category for lung cancer.

4. CLOSING

It is hoped that further research can get discoveries about algorithms that are more optimal using data that may be more than 1000 because, in some existing studies, the results of the accuracy turned out to be different after the researchers conducted this research and could also create a system to assist medical doctors in making predictions. To the patient whether the patient has lung cancer or not or even for other cases.

BIBLIOGRAPHY

[1] M. Yusril, 2022, "Classification of Lung Disease Using the Naïve Bayes Classifier Method," Journal of Tekinkom, Vol , No 2, 2022, doi:10.37600/tekinkom.v5i2.649

[2] US. Pratama, Safrizal, J.Iriani, "Expert System for Diagnosing Respiratory Disorders Due to Cigarette Smoke Using the Dempster Shafer Method," Vol 9, No 1 (2021): IT Journal April 2021.

[3] K. Indra, 2022, "Classification of Lung Cancer Using Convolutional Neural Networks with Efficientnet-B0 Architecture".https://repository.uin-suska.ac.id/57854/

[4] Martini Made (Ed). 2022. Treatment of Cancer Patients. Bandung. 5.[Online]. Available:https://www.google.co.id/books/edition/Perawatan_Pasien_Kanker/eIuaEAAAQBAJ?hl=id&gbpv=1&pg=PA5&printsec=frontcover

[5] Pradnyana, GA, Darmawiguna, IGM., Wijaya, INS 2020. "Data Mining: Finding Knowledge in Data". Depok. 1st, 1st printing. Available in the ePerpusdikbud application.

[6] Mohammed, SA, Darrab, S., Noaman, SA, Saake, G. 2020. "Analysis Of Breast Cancer Detection Using Different Machine Learning Techniques.", Communications in Computer and Information Science., Vol 1234 CCIS, 108–117, doi :10.1007/978-981-15-7205-0_10

[7] Angkasa, V and Pangaribuan, JJ 2022. "Comparison of Random Forest and Knn Accuracy Levels for Diagnosing Breast Cancer"., Journal Information System Development (ISD)., Vol 7 No 1, 50.

[8] DA Omondiagbe, S. Veeramani, and AS Sidhu,

"Machine Learning Classification Techniques for Breast Cancer Diagnosis," in IOP Conference Series: Materials Science and Engineering, 2019, vol. 495, no. 1, doi: 10.1088/1757- 899X/495/1/012033

[9] Kirubha, A and Priya, SM 2017. "Comparison of Classification Algorithms in Lung Cancer Risk Factor Analysis". International Journal of Science and Research (IJSR). Vol 6 Issue 2

[10] UCI Machine Learning, "Breast Cancer Prediction Data Set | Kaggle," https://www.kaggle.com/code/surajsingh07/breast-cancer-dataset

[11] UCI Machine Learning, "Lung Cancer Prediction Data Set | Kaggle," https://www.kaggle.com/datasets/thedevastator/cancer-patients-and-air-pollution-a-new-link

[12] R. Wajhillah, S. Bahri, A. Wibowo, "Comparison of Machine Learning Methods in Diagnosing Depression Mental Disorders," Syntax Journal of Informatics Vol. 9, No. 1, 2020.

[13] T. Wei, K. Liu, J. Shi, X. Liu, and L. Liu, "Research on Data Acquisition of Communication Equipment Based On Machine Learning," in 2019 International Conference on Computer Network, Electronic and Automation (ICCNEA), Sept. 2019, doi: 10.1109/ICCNEA.2019.00030.

[14] N. Verbeeck, RM Caprioli, and R. van de Plas, "Unsupervised machine learning for exploratory data analysis in imaging mass spectrometry," Mass Spectrometry Reviews, vol. 39, no. 3, 2020, doi: 10.1002/mas.21602.