CERDAS MENGUASAI GIT

CERDAS MENGUASAI GIT
Dalam 24 Jam

Rolly M. Awangga
Informatics Research Center

Kreatif Industri Nusantara

Penulis:
Rolly Maulana Awangga

ISBN : 978-602-53897-0-2

Editor:
M. Yusril Helmi Setyawan

Penyunting:
Syafrial Fachrie Pane
Khaera Tunnisa
Diana Asri Wijayanti

Desain sampul dan Tata letak:
Deza Martha Akbar

Penerbit:
Kreatif Industri Nusantara

Redaksi:
Jl. Ligar Nyawang No. 2
Bandung 40191
Tel. 022 2045-8529
Email : awangga@kreatif.co.id

Distributor:
Informatics Research Center
Jl. Sariasih No. 54
Bandung 40151
Email : irc@poltekpos.ac.id

Cetakan Pertama, 2019

'Jika Kamu tidak dapat
menahan lelahnya
belajar, Maka kamu harus
sanggup menahan
perihnya Kebodohan.'
Imam Syafi'i

CONTRIBUTORS

ROLLY MAULANA AWANGGA, Informatics Research Center., Politeknik Pos Indonesia, Bandung, Indonesia

CONTENTS IN BRIEF

DAFTAR ISI

FOREWORD

Sepatah kata dari Kaprodi, Kabag Kemahasiswaan dan Mahasiswa

KATA PENGANTAR

Buku ini diciptakan bagi yang awam dengan git sekalipun.

R. M. AWANGGA

Bandung, Jawa Barat
Februari, 2019

ACKNOWLEDGMENTS

Terima kasih atas semua masukan dari para mahasiswa agar bisa membuat buku ini lebih baik dan lebih mudah dimengerti.

Terima kasih ini juga ditujukan khusus untuk team IRC yang telah fokus untuk belajar dan memahami bagaimana buku ini mendampingi proses Intership.

R. M. A.

ACRONYMS

ACGIH	American Conference of Governmental Industrial Hygienists
AEC	Atomic Energy Commission
OSHA	Occupational Health and Safety Commission
SAMA	Scientific Apparatus Makers Association

GLOSSARY

git	Merupakan manajemen sumber kode yang dibuat oleh linus torvald.
bash	Merupakan bahasa sistem operasi berbasiskan *NIX.
linux	Sistem operasi berbasis sumber kode terbuka yang dibuat oleh Linus Torvald

SYMBOLS

A Amplitude

& Propositional logic symbol

a Filter Coefficient

$\mathcal{B}$ Number of Beats

INTRODUCTION

ROLLY MAULANA AWANGGA, S.T., M.T.

Informatics Research Center
Bandung, Jawa Barat, Indonesia

Pada era disruptif saat ini. git merupakan sebuah kebutuhan dalam sebuah organisasi pengembangan perangkat lunak. Buku ini diharapkan bisa menjadi penghantar para programmer, analis, IT Operation dan Project Manajer. Dalam melakukan implementasi git pada diri dan organisasinya.

Rumusnya cuman sebagai contoh aja biar keren[1].

$$ABC\mathcal{D}\mathcal{E}\mathcal{F}\alpha\beta\Gamma\Delta\sum_{def}^{abc} \tag{I.1}$$

BAB 1

CHAPTER 1

BAB 2

CHAPTER 2

BAB 3

CHAPTER 3

BAB 4

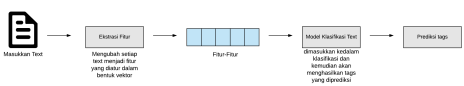
CHAPTER 4

4.1 Evietania - 1174051

4.1.1 Teori

1. Jelaskan apa itu klasifikasi teks, sertakan gambar ilustrasi buatan sendiri.

   klasifikasi teks adalah cara untuk memilah-milah teks berdasarkan parameter tertentu baik itu jenis teks atau jenis dari dokumen yang terdapat kumpulan teks didalamnya, sedangkan teks itu sndiri merupakan sekumpulan kata yang dapat dibaca. bisa berupa buku, majalah, rambu-rambu dan lain sebagainya.



Gambar 4.1 contoh klasifikasi teks

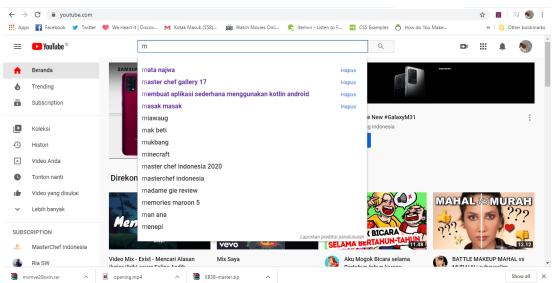
2. Jelaskan mengapa klasifikasi bunga tidak bisa menggunakan machine learning, sertakan ilustrasi sendiri.

   Klasifikasi bunga tidakdapat menggunakan mesin learning dikarenakan jenis-jenis bunga banyak yang mirip bahkan banyak bunga yang serupa tetapi tidak sama. oleh karena itu klasifikasi bunga tidakbisa di gunakan oleh mesin learning dikarenakan jika salah satu inputan ciri-ciri dari siatu bunga di inputkan kemungkinan jawaban dari mesin learning itu tidak tepat contoh dimasukan inputan ciri ciri bunga mawar putih kemudian mesin learning menjawab bahwa itu bunga mawar merah.

Gambar 4.2 contoh klasifikasi bunga

3. Jelaskan bagaimana teknik pembelajaran mesin pada teks pada kata-kata yang digunakan di youtube,jelaskan arti per atribut data csv dan sertakan ilustrasi buatan sendiri.

   cara pembelajaran teks yang di gunakan youtube yaitu dengan cara merekam data yang sering di inputkan oleh user pada menu pencarian youtube. sehingga pada saat user akan mencari data yang serupa seringkali youtube menyediakan opsi atau rekomendasi-rekomendasi dari pencaharian. contoh saya menuliskan m maka muncul opsi pilihan master chep dan lainya yang berawalan m rekomendasi yang muncul merupakan kata-kata yang sering di cari oleh banyak user atau sering di buka oleh user itu sendiri.

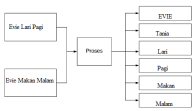


Gambar 4.3 contoh teknik pembelajaran mesin

4. Jelaskan apa yang dimaksud vektorisasi data.

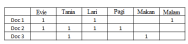
vektorisasi data merupakan pemechan data menjadi bagian bagian yang lebih sederhana contoh pada satu paragraf terdiri dari 200 kata kemudian dilakukan vektorisasi dengancara membagi-bagi kata dalam paragraf tersebut ke dalam kalimat-kalimat yang terpisah kemudian di pecah lagi menjadi data dalam perkata selanjutnya kata kata tersebut di terjemahkan.

5. Jelaskan apa itu bag of words dengan kata-kata yang sederhana dan ilustrasi sendiri. bag of words merupakan peroses penyederhanaan kata-kata yang asalnya tersiri dalam satu kalimat atau satu paragraf di ubah menjadi perkata kemudian kata-kata tersebut di kumpulkan menjadi satu kelompok tanpa ada arti dari kata-kata yang telah di kumpulkan tersebut lalu di hitung frekuensi kemunculan dari kata tersebut.



Gambar 4.4 contoh bag of words

6. Jelaskan apa itu TF-IDF, ilustrasikan dengan gambar sendiri. TF-IDF merupakan metode untuk menghitung bobot dari kata yang sering muncul pada suatu kalimat. metode ini menghitung nilai TF atau Term Frequency dan IDF atau Inverse Document Frequency pada setiap kata pada kalimat yang dijadikan acuan kata pada metode ini sering di sebut token adapun rumus dari metode ini.



	Evie	Tanis	Lari	Pagi	Makan	Malam
Doc 1	1		1			1
Doc 2	1	1	1	1		
Doc 3		1			1	

Gambar 4.5 contoh TF-IDF

4.1.2 Praktek

1. Buat aplikasi sederhana menggunakan pandas, buat data dummy format csv sebanyak 500 baris dan melakukan load ke dataframe pandas. Jelaskan arti setiap baris kode yang dibuat.

```
# In[1]: Import library dari pandas
import pandas as pan
#Membaca file csv menggunakan pandas
data_forest = pan.read_csv('praktek1.csv')
# In[2]: untuk melihat jumlah dari baris data yang telah di
    import
print(len(data_forest))
# In[3]: untuk melihat lima baris pertama data yang telah di
    import
print(data_forest.head())
# In[4]: untuk mengetahui banyak baris dan kolom dari data yang
```

```
# telah di import.
print(data_forest.shape)
```

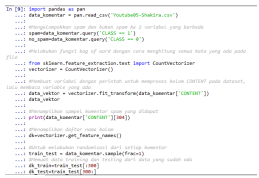
2. Dari dataframe tersebut, dipecah menjadi dua dataframe yaitu 450 row pertama dan 50 row sisanya(Harus beda dengan teman sekelas)

```
#%%
data_450 = data_forest[:450]
#%%
data_50 = data_forest[450:]
```

3. Praktekkan vektorisasi dan klasifikasi dari data Shakira dengan decision tree. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud dari setiap luaran yang didapatkan.

```
npm = 1174039%4
print(npm)
#%%
#Import pandas dan meload file Youtube05-Shakira
import pandas as pan
data_komentar = pan.read_csv('Youtube05-Shakira.csv')

#Mengelompokkan spam dan bukan spam ke 2 variabel yang berbeda
spam=data_komentar.query('CLASS == 1')
no_spam=data_komentar.query('CLASS == 0')

#Melakukan fungsi bag of word dengan cara menghitung semua kata
    yang ada pada file
from sklearn.feature_extraction.text import CountVectorizer
vectorizer = CountVectorizer()

#Membuat variabel dengan perintah untuk memproses kolom CONTENT
    pada dataset, lalu membaca variable yang ada
data_vektor = vectorizer.fit_transform(data_komentar['CONTENT'])
data_vektor

#Menampilkan sampel komentar spam yang didapat
print(data_komentar['CONTENT'][304])

#Menampilkan daftar nama kolom
dk=vectorizer.get_feature_names()

#Untuk melakukan randomisasi dari setiap komentar
train_test = data_komentar.sample(frac=1)
#Memuat data training dan testing dari data yang sudah ada
dk_train=train_test[:300]
dk_test=train_test[300:]

#Melakukan training pada data training dan memvektorisasi data
    tersebut
dk_train_att = vectorizer.fit_transform(dk_train['CONTENT'])
dk_train_att

#Melakukan Testing pada data testing dan memvektorisasi data
    tersebut
```

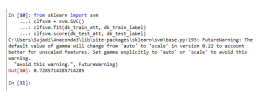
```
dk_test_att=vectorizer.transform(dk_test['CONTENT'])
dk_test_att

#Mengambil label spam dan bukan spam
dk_train_label=dk_train['CLASS']
dk_test_label = dk_test['CLASS']
```



Gambar 4.6 Melakukan vektorisasi dan mengambil salah satu contoh

4. Cobalah klasifikasikan dari data vektorisasi yang ditentukan di nomor sebelumnya dengan klasifikasi SVM. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.

```
from sklearn import svm
clfsvm = svm.SVC()
clfsvm.fit(dk_train_att, dk_train_label)
clfsvm.score(dk_test_att, dk_test_label)
```



Gambar 4.7 Melakukan prediksi CVM berdasarkan nilai

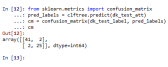
5. Coba klasifikasikan dari data vektorisasi yang ditentukan di nomor sebelumnya dengan klasifikasi Decision Tree. Tunjukkan kelaurannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan.

```
#Melakukan klasifikasi Decision Tree
from sklearn import tree
clftree = tree.DecisionTreeClassifier()
clftree.fit(dk_train_att, dk_train_label)
clftree.score(dk_test_att, dk_test_label)
```

Gambar 4.8 Klasifikasi data dari vektorisasi yang ada dengan decision tree

6. Plotlah confusion matrix dari praktek modul ini menggunakan matplotlib. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan

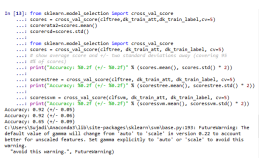
```
#%%
#Melakukan confusion matrix
from sklearn.metrics import confusion_matrix
pred_labels = clftree.predict(dk_test_att)
cm = confusion_matrix(dk_test_label, pred_labels)
cm
```



Gambar 4.9 Penggunaan confusion matrix

7. Jalankan program cross validation pada bagian teori bab ini. Tunjukkan keluarannya dari komputer sendiri dan artikan mnaksud setiap luaran yang didapatkan.

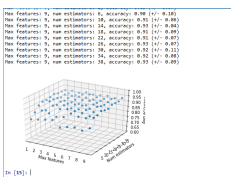
```
from sklearn.model_selection import cross_val_score
scores = cross_val_score(clftree,dk_train_att,dk_train_label,cv
    =5)
scorerata2=scores.mean()
scorersd=scores.std()

from sklearn.model_selection import cross_val_score
scores = cross_val_score(clftree, dk_train_att, dk_train_label,
    cv=5)
# show average score and +/- two standard deviations away (
    covering 95
#% of scores)
print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std
    () * 2))

scorestree = cross_val_score(clftree, dk_train_att,
    dk_train_label, cv=5)
print("Accuracy: %0.2f (+/- %0.2f)" % (scorestree.mean(),
    scorestree.std() * 2))

scoressvm = cross_val_score(clfsvm, dk_train_att, dk_train_label,
     cv=5)
print("Accuracy: %0.2f (+/- %0.2f)" % (scoressvm.mean(),
    scoressvm.std() * 2))
```



Gambar 4.10 Mengambil tingkat akurasi dari klasifikasi data dan prediksi

8. Buatlah program pengamatan komponen informasi pada bagian teori bab ini. Tunjukkan keluarannya dari komputer sendiri dan artikan maksud setiap luaran yang didapatkan

```
#%%
from sklearn.ensemble import RandomForestClassifier
import numpy as np
max_features_opts = range(1, 10, 1)
n_estimators_opts = range(2, 40, 4)
rf_params = np.empty((len(max_features_opts)*len(
    n_estimators_opts),4), float)
i = 0
for max_features in max_features_opts:
    for n_estimators in n_estimators_opts:
        clf = RandomForestClassifier(max_features=max_features,
    n_estimators=n_estimators)
        scores = cross_val_score(clf, dk_train_att,
    dk_train_label, cv=5)
        rf_params[i,0] = max_features
        rf_params[i,1] = n_estimators
        rf_params[i,2] = scores.mean()
        rf_params[i,3] = scores.std() * 2
        i += 1
        print("Max features: %d, num estimators: %d, accuracy:
    %0.2f (+/- %0.2f)"
%   (max_features, n_estimators, scores.mean(), scores.std() * 2)
    )

import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm
fig = plt.figure()
fig.clf()
ax = fig.gca(projection='3d')
x = rf_params[:,0]
y = rf_params[:,1]
z = rf_params[:,2]
ax.scatter(x, y, z)
ax.set_zlim(0.6, 1)
ax.set_xlabel('Max features')
ax.set_ylabel('Num estimators')
ax.set_zlabel('Avg accuracy')
plt.show()
```



Gambar 4.11 Melakukan regresi data dengan numpy dan mengambil fitur - fitur yang ada

4.1.3 Penanganan Error

4.1.3.1 Sreenshoot Error

1. Error 1

```
FileNotFoundError: [Errno 2] File b'praktek.csv' does not exist: b'prak
```

Gambar 4.12 error 1

4.1.3.2 Penanganan untuk Error

1. penanganan 1, kesalahan pada nama file yang dituju, dimana cara penanganannya adalah memasukkan nama file dengan benar

```
# In[1]: Import library dari pandas
import pandas as pan
#Membaca file csv menggunakan pandas
data_forest = pan.read_csv('praktek1.csv')
# In[2]: untuk melihat jumlah dari baris data yang t
```

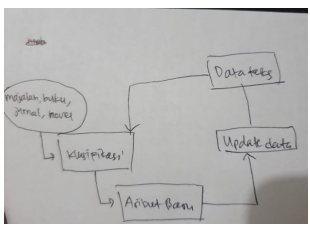
Gambar 4.13 penanganan1

4.2 1174008 - Arjun Yuda Firwanda

4.2.1 Teori

1. Klasifikasi Text

   Pengolahan teks sangat dibutuhkan pada aplikasi yang memiliki jumlah dokumen yang berukuran besar. klasifikasi teks sendiri merupakan cara dalam memilah data teks berdasarkan parameter serta dengan data yang bersifat dokumen. Data tersbut dapat berupa char atau string.



Gambar 4.14 Klasifikasi Text

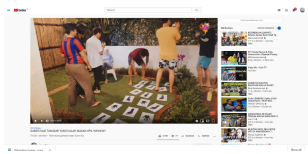
2. Mengapa Klasifikasi Bunga tidak dapat menggunakan Machine Learning

Dikarenakan klasifikasi itu menggunakan tipe data yang dimana atributnya memiliki nilai data yang berupa vektor dengan perbandingan masing-masing data yang memiliki sedikit perbedaan. Dapat dilihat pada gambar berikut.

Gambar 4.15 Klasifikasi Bunga

3. Teknik Pembelajaran maachine learning pada teks kata-kata di youtube

   Contohnya pada saat kita membuka sebuah video maka di sebelah kanan ada list ”berikutnya”, pada saat itu mesin melakukan ujicoba dan apabila anda menekan salah satu dari video tersebut maka hal tersebut akan direkam dan disimpan oleh mesin tersebut



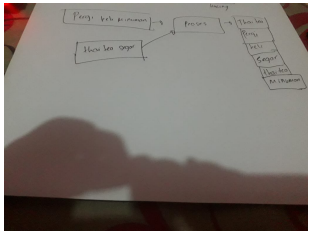
Gambar 4.16 Machine Learning Youtube

4. Jelaskan apa yang dimaksud vektorisasi data.

   vektorisasi data merupakan pemechan data menjadi bagian bagian yang lebih sederhana contoh ada sebuah paragraf, dari paragraf tersebut akan dibagi bagi menjadi kalimat, yang nantinya akan dibagi bagi kembali menjadi perkata.

5. Jelaskan apa itu bag of words dengan kata-kata yang sederhana dan ilustrasi sendiri

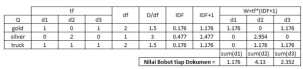
   Bag of words merupakan penyajian sederhana yang biasanya digunakan pada aplikasi text mining pada saat mengenalkan strukturke sebuah kumpulan dokumen yang berbasis teks untuk diklasifikasikan kembali menjadi dua atau lebih kelas yang telah di tentukan.



Gambar 4.17 Bag of Words

6. Jelaskan apa itu TF-IDF, ilustrasikan dengan gambar sendiri.

   TF-IDF merupakan metode untuk menghitung bobot dari kata yang sering muncul pada suatu kalimat. metode ini menghitung nilai Term Frequency dan IDF atau Inverse Document Frequency pada setiap kata pada kalimat yang dijadikan acuan kata pada metode ini sering di sebut token adapun rumus dari metode ini dapat dilihat pada gambar.



	tf			df	D/df	IDF	IDF+1	W=tf*(IDF+1)		
Q	d1	d2	d3					d1	d2	d3
gold	1	0	1	2	1.5	0.176	1.176	1.176	0	1.176
silver	0	2	0	1	3	0.477	1.477	0	2.954	0
truck	1	1	1	2	1.5	0.176	1.176	0	1.176	1.176
								sum(d1)	sum(d2)	sum(d3)
							Nilai Bobot tiap Dokumen =	1.176	4.13	2.352

Gambar 4.18 TF-IDF

4.2.2 Praktek

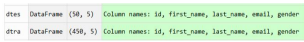
1. import data pandas dan 500 baris data dumy kemudian di jelaskan tiap barisnya.

```
# In[1]: Import library dari pandas
import pandas as pan
#Membaca file csv menggunakan pandas
data_forest = pan.read_csv('1174008.csv')
# In[2]: untuk melihat jumlah dari baris data yang telah di
    import
print(len(data_forest))
```

Gambar 4.19 Data Dummy

2. memecah data prame menjadi dua yag pertama 450 dan kedua sisanya

```

#Mengelompokkan spam dan bukan spam ke 2 variabel yang berbeda
spam=data_komentar.query('CLASS == 1')
```



Gambar 4.20 Pisah Data

3. praktek vektorisasi

```
no_spam=data_komentar.query('CLASS == 0')

#Melakukan fungsi bag of word dengan cara menghitung semua kata
    yang ada pada file
from sklearn.feature_extraction.text import CountVectorizer
vectorizer = CountVectorizer()

#Membuat variabel dengan perintah untuk memproses kolom CONTENT
    pada dataset, lalu membaca variable yang ada
data_vektor = vectorizer.fit_transform(data_komentar['CONTENT'])
data_vektor

#Menampilkan sampel komentar spam yang didapat
print(data_komentar['CONTENT'][304])

#Menampilkan daftar nama kolom
dk=vectorizer.get_feature_names()

#Untuk melakukan randomisasi dari setiap komentar
train_test = data_komentar.sample(frac=1)
#Memuat data training dan testing dari data yang sudah ada
dk_train=train_test[:300]
dk_test=train_test[300:]

#Melakukan training pada data training dan memvektorisasi data
    tersebut
dk_train_att = vectorizer.fit_transform(dk_train['CONTENT'])
dk_train_att

#Melakukan Testing pada data testing dan memvektorisasi data
    tersebut
dk_test_att=vectorizer.transform(dk_test['CONTENT'])
dk_test_att

#Mengambil label spam dan bukan spam
```

lakukan import library pandas yang di inisialisasi menjasi pd setelah itu ada dibuat variable data dengan method read_csv untuk membaca file berekstensikan csv yang di masukan alamatnya pada kurung, lakukan klasifikasi atau pemilihan komentar yang berisi spam atau bukan spam dengan parameter class samadengan 1 merupakan spam dan class samadengan 0 bukan spam setelah itu masukan librari CountVektorizer yang digunakan untuk vektorisasi data kemudian dilanjutkan pada bagian In[103] dibuat variabel yang berisi vektorisasi dari data pada data di field content setelah itu variabel tersebut di running hasilnya menunjukan 350 baris di kali 1738 kolom selanjutnya dicoba untuk memunculkan isi recod pada baris ke 345 maka akan muncul isian dari baris tersebut.

selanjutnya dibuat variabel dk atau daftar yang berisi data hasil vektorisasi setelah yang terdiri dari variabel dshuf yang berisi data komen yang di dalamnya di buat random yang nantinya akan dibut data training dan data testing dengan ketentuan data training 300 dan data testing sebanyak 50 setelah itu data training di lakukan vektorisasi dan data testing juga dilakukan vektorisasi setelah itu kedua data training dan testing tersebut dibuat label dengan parameter field CLASS pada tabel.

Gambar 4.21 Vektorisasi

4. klasifikasi SVM

```
dk_test_label = dk_test['CLASS']

#%%
#Coding untuk melakukan klasifikasi SVM
from sklearn import svm
```

import librari svm dari sklearn kemudian membuat variabel clfsvm berisikan method svc setelah itu variabel tersebut di berikan method fit dengan isian data train vektorisasi dan data training label yang berguna untuk melatih data tersebut setelah itu di coba untuk memunculkan score atau akurasi dari data tersebut menggunakan data testing vektorisasi dan data testing label.

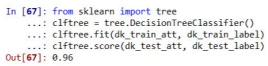
Gambar 4.22 SVM

5. klasifikasi decision tree

```
clfsvm.fit(dk_train_att, dk_train_label)
clfsvm.score(dk_test_att, dk_test_label)
```

import librari tree dari sklearn kemudian membuat variabel clftree berisikan method DecisionTreeClasifier setelah itu variabel tersebut di berikan method fit dengan isian data train vektorisasi dan data training label yang berguna untuk

melatih data tersebut agar dapat digunakan pada codingan selanjutnya setelah itu di coba untuk memunculkan score atau akurasi dari data tersebut menggunakan data testing vektorisasi dan data testing label.

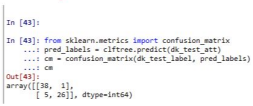


Gambar 4.23 Desicion Tree

6. plot comfusion matrix

```
from sklearn import tree
clftree = tree.DecisionTreeClassifier()
clftree.fit(dk_train_att, dk_train_label)
clftree.score(dk_test_att, dk_test_label)
```

import library comfusion matrix selanjutnya dilakukan prediksi pada pada data tes nya kemudian data tersebut di masukan kedalam variabel cm dengan method confusion matrix yang di dalamnya terdapat data dari variabel perd label dan dk test label setelah itu variabel cm tersebut di running maka akan memunculkan nilai matrixnya.



Gambar 4.24 Confussion Matrix

7. cross valodation

```
#%%
#Melakukan confusion matrix
from sklearn.metrics import confusion_matrix
pred_labels = clftree.predict(dk_test_att)
cm = confusion_matrix(dk_test_label, pred_labels)
cm

#%%
from sklearn.model_selection import cross_val_score
scores = cross_val_score(clftree,dk_train_att,dk_train_label,cv
    =5)
scorerata2=scores.mean()
scorersd=scores.std()

from sklearn.model_selection import cross_val_score
```

```
scores = cross_val_score(clftree, dk_train_att, dk_train_label,
    cv=5)
# show average score and +/- two standard deviations away (
    covering 95
#% of scores)
print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std
    () * 2))

scorestree = cross_val_score(clftree, dk_train_att,
    dk_train_label, cv=5)
```

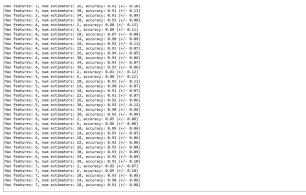
memunculkan nilai akurasi dari tiga metode yaitu random forest, decision tree, dan klasifikasi svm (suport vector machine) diamana akan di bandingkan tingkat akurasi dari semua hasil akurasiya mana yang terbaik dan lebih akurat pada hasilnya data yang paling akurat yaitu random forest.

Gambar 4.25 Cross Validation

8. Pengamatan program

```
scoressvm = cross_val_score(clfsvm, dk_train_att, dk_train_label,
     cv=5)
print("Accuracy: %0.2f (+/- %0.2f)" % (scoressvm.mean(),
    scoressvm.std() * 2))

#%%
from sklearn.ensemble import RandomForestClassifier
import numpy as np
max_features_opts = range(1, 10, 1)
n_estimators_opts = range(2, 40, 4)
rf_params = np.empty((len(max_features_opts)*len(
    n_estimators_opts),4), float)
i = 0
for max_features in max_features_opts:
    for n_estimators in n_estimators_opts:
        clf = RandomForestClassifier(max_features=max_features,
    n_estimators=n_estimators)
        scores = cross_val_score(clf, dk_train_att,
    dk_train_label, cv=5)
        rf_params[i,0] = max_features
        rf_params[i,1] = n_estimators
        rf_params[i,2] = scores.mean()
```

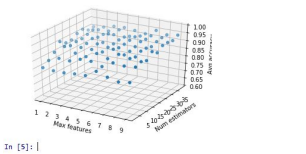
terdapat grafik data yang terdapat dari grafik tersebut di dapat dari codingan dengan cara pengulangan data masing masing 10 kali setelah itu di eksekusi menjadi grafik berbentuk 3D pada gambar tersebut menunjukan rasio dari yang terrendah yaitu data SVM kemudian data decision tree dan hasil random forest.



Gambar 4.26 Pengamatan Program

Berikut adalah untuk grafik

```
% (max_features, n_estimators, scores.mean(), scores.std() * 2)
    )

import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm
fig = plt.figure()
fig.clf()
ax = fig.gca(projection='3d')
x = rf_params[:,0]
y = rf_params[:,1]
z = rf_params[:,2]
ax.scatter(x, y, z)
ax.set_zlim(0.6, 1)
ax.set_xlabel('Max features')
ax.set_ylabel('Num estimators')
```



Gambar 4.27 Grafik

4.2.3 Penanganan Error

4.2.3.1 Sreenshoot Error

1. Error 1

Gambar 4.28 Error1

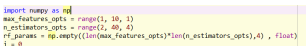
4.2.3.2 Kode Error dan Jenisnya

1. Kode Error 1 jenis Name Error

Gambar 4.29 Error1

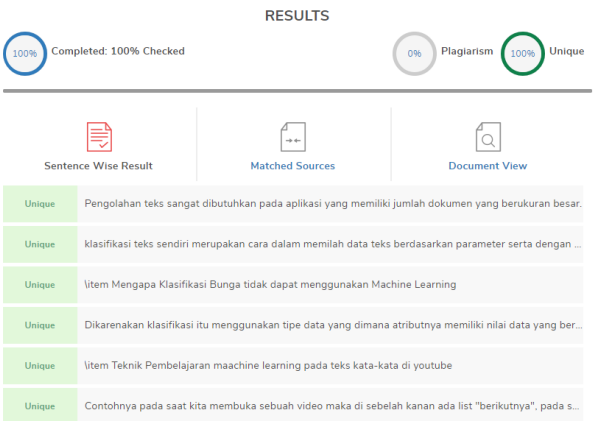
4.2.3.3 Solusi

1. Mengimport liibrary numpy sebagai np



Gambar 4.30 Solusi 1

4.2.4 Bukti Tidak Plagiat



Gambar 4.31 Bukti Tidak Plagiat.

BAB 5

CHAPTER 5

BAB 6

CHAPTER 6

BAB 7

CHAPTER 7

DAFTAR PUSTAKA

[1] R. Awangga, “Sampeu: Servicing web map tile service over web map service to increase computation performance,” in *IOP Conference Series: Earth and Environmental Science*, vol. 145, no. 1. IOP Publishing, 2018, p. 012057.

Index