KLASIFIKASI KASUS COVID-19 MENGGUNAKAN MODEL *NAIVE BAYES CLASSIFIER*

Ahmad Firdaus, Miftahul Walid, Anwari
Program Studi Sistem Informasi S1, Fakultas Teknik
Universitas Islam Madura
ahmadfirdaus290497@gmail.com

ABSTRAK

Covid-19 dengan berbagai macam penanganannya membuat banyak pertentangan menyertainya. Untuk mengetahui apakah benar kasus pasien terkonfirmasi positif covid-19 tinggi atau rendah. Dalam tugas akhir ini, model *Naive Bayes Classifier* memprediksi bahwa kasus covid-19 di Pamekasan tinggi (>= 30 orang) dan model *Gaussian Naive Bayes* menunjukkan kinerja yang sangat baik dengan nilai akurasi model sebesar 0,9688 serta nilai *precision* (0,97), *recall* (1,00) dan *f-1 score* (0,98). menunjukkan bahwa data yang menyatakan kasus covid-19 di Kabupaten Pamekasan termasuk tinggi dengan pengambilan data dalam rentang waktu Bulan Januari 2021 sampai dengan Desember 2021. Diharapkan untuk penelitian selanjutnya sistem dapat dikembangkan lagi karena data yang didapat saat ini masih terbatas.

Kata kunci : *classifier, covid-19, gaussian, naive bayes.*

1. PENDAHULUAN

Perkembangan suatu informasi yang akurat membutuhkan sebuah tambang data agar dapat memperoleh data baru yang valid. Seperti halnya perkembangan covid-19 yang sudah berlangsung selama 2 tahun lebih, data pasien belum terekam sepenuhnya. Menyebabkan para pengambil keputusan tidak bisa secepatnya menentukan langkah dan strategi yang dibutuhkan terkait masalah virus covid-19 ini. Perkembangan penyakit ini yang sudah masuk kategori wabah sangat cepat, sehingga dalam pendataan dan ketersediaan data juga semakin cepat berkembangnya. Karena itulah dibutuhkan sebuah analisa tambang data yang akurat sebagai informasi pendukung keputusan para pengambil kebijakan. Di samping itu, perkembangan internet memiliki peranan yang cukup besar dalam akumulasi data. Data yang banyak, namun pengelolaannya tidak maksimal memunculkan suatu kondisi yang sering disebut " rich of data but poor of information " karena data yang ada tidak dapat digunakan untuk aplikasi yang berguna. Tidak jarang data – data tersebut dibiarkan begitu saja seakan – akan " kuburan data " [1].

Berdasarkan latar belakang diatas, dibutuhkan suatu analisis dan pengolahan data untuk klasifikasi mengenai kasus pasien covid-19 yang terekam di RSUD Kabupaten Pamekasan. Dengan tujuan untuk mendapatkan data yang akurat mengenai kasus penyebaran covid-19 tergolong kasus tinggi atau rendah.

2. TINJAUAN PUSTAKA

Adapun penelitian terdekat adalah penelitian mengenai penerapan *data mining* untuk klasifikasi kasus covid-19 menggunakan algoritma *Naïve Bayes* dengan studi kasus di daerah Sulawesi Selatan [2]. Dengan hasil penelitian akurasi klasifikasi sebesar 91% dari 23 data latih dan 2 diantaranya tidak diklasifikasikan secara tepat. Software yang digunakan untuk menganalisis data dalam penelitian ini adalah *Jupyter Notebook*.

.

2.1. *Knowledge Discovery in Database* (KDD)

Knowledge Discovery In Database merupakan proses yang menentukan tentang suatu informasi yang bermanfaat serta dapat menentukan suatu pola di dalam sebuah data. Informasi ini terkandung di dalam basis data yang berukuran besar yang sebelumnya tidak diketahui [3].

Hasil dari penelitian, yaitu tahapan proses di dalam KDD (*Knowledge Data Discovery*) antara lain sebagai berikut [4]:

1. Seleksi Data, tahapan ini dilakukan untuk memilih data yang sesuai variabel yang dibutuhkan di dalam penelitian. yaitu dengan menentukan atribut dari data mana saja yang akan digunakan dalam penelitian dari sekelompok data operasional.
2. Proses *cleaning*, dilakukan terhadap keseluruhan data yang jumlah yang di teliti.
3. Transformasi, proses yang satu ini menghasilkan satu rekaman data set.
4. Proses analisis data.

2.2. *Data Mining*

Data mining merupakan suatu proses yang paling pertama yang akan dilakukan, yaitu mencari suatu informasi yang benar-benar baru [5]. *Data mining* banyak digunakan untuk mengambil intisari pengetahuan dari suatu kumpulan data yang dapat dengan mudah dimengerti oleh manusia. Meliputi *database* dan manajemennya, pra-pemrosesan data, pertimbangan model dan inferensi, ukuran ketertarikan, pertimbangan kompleksitas, serta pasca pemrosesan terhadap struktur data yang telah ditemukan, visualisasi data, dan *online updating* [6].

Sedangkan Gartner Group [7] mengatakan bahwa *Data mining* merupakan suatu proses untuk menemukan hubungan baru yang memiliki arti, pola dan kebiasaan dengan memilah sebagian besar data yang disimpan dalam media penyimpanan dengan menggunakan teknik pengenalan pola seperti halnya teknik statistik dan nodel matematika. *Data mining* merupakan gabungan dari beberapa disiplin ilmu yang menyatukan teknik dari pembelajaran mesin, pengenalan pola, statistik, *database*, dan visualisasi data untuk menangani permasalahan pengambilan informasi dari suatu basis data yang besar [8].

Data mining dapat dikelompokkan berdasarkan fungsinya, yaitu:

1. Deskripsi. Peneliti dan analis mencoba mencari cara untuk menggambarkan pola dan kecenderungan yang terdapat dalam sekumpulan data.
2. Estimasi. Estimasi hampir sama dengan klasifikasi, kecuali variabel target estimasi lebih ke arah numerik daripada ke arah kategori.
3. Prediksi. Prediksi hampir sama dengan klasifikasi dan estimasi, bedanya bahwa dalam prediksi nilai dari hasil akan ada dimasa mendatang. Contoh prediksi dalam bisnis dan penelitian adalah :
   a. Prediksi harga minyak dalam tiga bulan yang akan datang.
   b. Prediksi tingkat penganguran lima tahun akan datang.
   c. Prediksi persentase kenaikan kecelakaan lalu lintas tahun depan jika batas bawah kecepatan dinaikan.

Beberapa metode dan teknik yang digunakan dalam klasifikasi dan estimasi dapat juga digunakan (untuk keadaan yang tepat) pada prediksi.

2.3. *Classification* (Klasifikasi)

Classification merupakan sebuah cara untuk menemukan properti yang sama pada himpunan objek di dalam sebuah data dan mengklasifikasikannya ke dalam kelas-kelas yang berbeda menurut model klasifikasi yang di tetapkan. Bertujuan untuk menemukan model dari *training set* yang membedakan kelas yang sesuai, model tersebut selanjutnya digunakan untuk mengklasifikasikan atribut yang kelasnya belum di ketahui. Beberapa metode klasifikasi di gunakan di dalam *Data Mining*. Dimana di dalam klasifikasi, terdapat target variabel kategori.

2.4. Metode *Naïve Bayes*

Metode *Naive Bayes* merupakan gabungan Teorema *Bayes* dengan "*Naïve*" yang berarti setiap atribut atau variabelnya bersifat bebas (*independent*). *Naïve Bayes Classifier* dapat dilatih dengan efisien dalam pembelajaran terawasi (*supervised learning*). Klasifikasi memiliki keuntungan hanya butuh sejumlah kecil data pelatihan untuk dapat memperkirakan parameter (varians dan sarana dari variabel) yang diperlukan untuk klasifikasi. Karena variabel yang independen diasumsikan, hanya saja variasi dari variabel pada masing-masing kelas harus ditentukan, bukan semua matriks kovarians [9].

Metode *Bayesian Learning* yang paling mudah, cepat, dan sederhana. oleh dikarena itu berasal dari teorema *Bayes* dan hipotesis kebebasan, menghasilkan *classifier statistic* berdasarkan peluang. *Naïve Bayes* merupakan pengklasifikasian dengan metode *probabilitas* dan *statistic* yang di kemukakan oleh ilmuan inggris Thomas Bayes [10].

Probabilitas yang di dalam memproduksi perkiraan akhir dihitung sebagai jumlah frekuensi dari "master" tabel keputusan. Teorema Bayes berasal dari rumus:

$$\mathbf{P(A \mid B) = P(B \cap A)/P(B)} \quad (1)$$

Dimana P(A| B) artinya peluang A dan keadaan B sebagai persamaan (1)

$$\mathbf{P(A \cap B) = P(B \mid A)\,P(A)} \quad (2)$$

Sehingga di dapatkan teorema Bayes:

$$\mathbf{P(A \mid B) = P(A \mid B)P(A) \,/\, P(B)} \quad (3)$$

Keterangan:
P : Probabilitas
A : Kriteria
B : Keadaan

Teorema Bayes memiliki rumus :

$$\mathbf{P(H \mid X) = P(X \mid H)P(H) \,/\, P(X)}$$

Keterangan :
X : Data dengan kelas yang belum diketahui
H : Hipotesis data X merupakan suatu kelas spesifik
P(H|X) : Probabilitas hipotesis H berdasarkan kondisi x (posteriori prob.)
P(H) : Probabilitas hipotesis H (prior prob.)
P(X|H) : Probabilitas X berdasarkan kondisi tersebut
P(X) : Probabilitas dari X

2.5. Bahasa Pemrograman *Python*

Python adalah *scripting language* yang berorientasi objek. Bahasa pemrograman ini dapat digunakan untuk pengembangan perangkat lunak dan bisa dijalankan melalui berbagai sistem operasi. Saat ini, *Python* juga merupakan bahasa yang populer bagi bidang *data science* dan analisis. Hal ini dikarenakan oleh dukungan bahasa *Python* terhadap *library – library* yang didalamnya menyediakan fungsi analisis data dan fungsi *machine learning*, *data preprocessing tools*, serta visualisasi data [11].

Berikut ini adalah beberapa alasan *Python* menjadi bahasa yang populer, khususnya dalam ranah analisis data dan *data science*:

1. *Open-source library*, *frameworks*, dan tools untuk *data mining.*
2. Relatif lebih mudah dipahami.
3. Multifungsi, tidak hanya untuk *data processing*, namun juga bisa untuk tugas lain seperti membuat website dan tampilan GUI (*Graphical User Interface*).

Dalam membuat sebuah proyek *data mining* dengan *Python*, dapat menggunakan *Anaconda*, dimana *Anaconda* telah menyediakan berbagai kelengkapan *Python* yang lebih dikhususkan untuk kebutuhan analisis data.

IDE (*Integrated Development Enviroment*) yang dapat digunakan antara lain *Jupyter Notebook* dengan *extension. ipynb* yang sudah merupakan bawaan dari *Anaconda Navigator*.

Berikut adalah contoh proses yang dilakukan oleh Python untuk proyek data mining sederhana:

1. Data mentah, dapat diimpor ke Python dengan menggunakan library Library ini mempunyai fungsi untuk mengimpor data dengan format csv ke Python.
2. Eksplorasi data dan data *preprocessing* dapat dilakukan dengan lebih mudah, karena Python telah memiliki fungsi untuk melihat persebaran data dan melakukan manipulasi data untuk handle data yang tidak sesuai. Dalam kasus ini library yang berfungsi adalah sklearn (Sci – kit learn) dan NumPy. Untuk visualisasi data, salah satu library yang terkenal adalah Matplotlib, dimana dapat membuat visualisasi dari persebaran data termasuk plot dan chart.
3. Pembuatan model *data mining* dapat dilakukan dengan cepat dengan tersedianya fitur – fitur machine learning yang cukup lengkap dari Sci – kit learn. Setelah memilih metode yang diinginkan, dapat langsung menggunakan fungsi didalam library.
4. Evaluasi model yang telah dibuat, seperti menghitung akurasi, spesifisitas, sensitivitas, presisi, error rate dapat ditampilkan dengan mudah dengan menggunakan fungsi classification_score yang tersedia pada Sci – kit learn. Hanya dengan satu baris, seluruh komponen evaluasi akan ditampilkan. Tidak hanya itu, fungsi pada Sci-kit learn juga dapat menampilkan *confusion matrix* yang berisi nilai prediksi dan aktual yang dilakukan dari data testing.

Merepresentasikan hasil dari model dapat divisualisasikan dalam bentuk plot ataupun hasil dalam bentuk web. Ketika permintaan penampilan hasil dalam bentuk web, maka hasil dari model dapat disinkronkan karena *Python* juga mendukung untuk pembuatan web. Selain itu, untuk menampilkan model, dapat menggunakan library – library lain, contohnya adalah graphviz untuk menampilkan *decision tree*.

3. METODE PENELITIAN

3.1. Pengumpulan Data

Metode pengumpulan data yang akan digunakan dalam penelitian ini adalah sebagai berikut:

a. Observasi
b. Metode *Interview*.
c. Studi Pustaka.

Objek penelitian adalah RSUD Kabupaten Pamekasan sebagai pusat perawatan darurat Covid-19 di Kabupaten Pamekasan.

Data yang akan dianalisa merupakan data pasien covid-19 pada Bulan Januari sampai dengan bulan Desember 2021. berikut ini contoh tabel struktur data penelitian:

Tabel 1. Tabel contoh struktur data penelitian

No	Bulan	Kecamatan	JK	Positif	Rawat	Sembuh	Mati
1	januari	PAMEKASAN	P	152	4	3	1
2	januari	TLANAKAN	L	122	2	2	0
3	januari	PADEMAWU	P	117	3	3	0
4	januari	BATU MARMAR	L	132	3	1	2
…	….	…..	…	….	….	…	….
156	Desember	GALIS	L	83	1	1	0

3.2. Tahapan analisis data

Adapun tahapan - tahapan analisis dataset yang akan dilakukan, sebagai berikut :

1. Mengkoneksikan database
2. Memisah dataset
3. Menentukan vektor fitur dan variabel target
4. Memisahkan dataset menjadi train set dan test set
5. Rekayasa Fitur
6. Pengskalaan fitur
7. Analisis data dengan model Naive Bayes
8. Menghitung confusion matrix dan nilai akurasi model
9. Visualisasi model dengan histogram

4. HASIL DAN PEMBAHASAN

Tugas akhir ini menganalisis kasus *covid-19* di Kabupaten Pamekasan. Dengan menghitung seberapa banyak kasus pasien *covid-19* yang positif, sembuh, dirawat dan meninggal dalam kurun waktu 1 tahun di Kabupaten Pamekasan. Analisa dilakukan dengan menggunakan metode *Naive Bayes Classification*, dalam hal ini menggunakan model *GaussianNB* yang telah tersedia dalam *library* bahasa pemrograman *Python*.

Dengan menggunakan data yang didapat dari RSUD Pamekasan, maka proses analisa dilakukan sesuai tahapan berikut ini :

4.1. Mengkoneksikan struktur data ke dalam *Jupyter Notebook*, dengan perintah :

```
import pandas as pd
import numpy as np
# input data
data_covid=pd.read_excel('data_covid.xlsx')
# Menampilkan data
data_covid.head(10)
```

Dan tampilan *dataframe*, seperti tampak pada gambar 1 berikut ini :

	NO.	Bulan	Kecamatan	gender	positif	rawat	Sembuh	Meninggal
0	1	januari	PAMEKASAN	P	152	4	3	1
1	2	januari	TLANAKAN	L	122	2	2	0
2	3	januari	PADEMAWU	P	117	3	3	0
3	4	januari	BATU MARMAR	L	132	3	1	2
4	5	januari	WARU	P	144	1	2	0
5	6	januari	KADUR	P	33	1	1	0
6	7	januari	LARANGAN	L	28	0	0	0
7	8	januari	PAKONG	L	27	0	0	0
8	9	januari	PASEAN	L	42	0	0	0
9	10	januari	PALENGAAN	L	39	0	0	0

Gambar 1. *Dataframe* kasus pasien covid-19 di Pamekasan

4.2. Melihat detail informasi mengenai tipe data yang ada di *dataframe* sebelum data diolah.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 156 entries, 0 to 155
Data columns (total 8 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   NO.        156 non-null    int64
 1   Bulan      156 non-null    object
 2   Kecamatan  156 non-null    object
 3   gender     156 non-null    object
 4   positif    156 non-null    int64
 5   rawat      156 non-null    int64
 6   Sembuh     156 non-null    int64
 7   Meninggal  156 non-null    int64
dtypes: int64(5), object(3)
memory usage: 9.9+ KB
```

Gambar 2. Informasi tipe data pada *Dataframe*

Pada gambar 2 terlihat informasi mengenai tipe data dari masing-masing kolom, ada 2 tipe data yaitu : *integer* dan *text*. Dan tidak ada *missing value* (*non null*).

4.3. Memisah dataset.

Pada tahapan ini akan memisahkan dataset menjadi variabel kategorikal dan numerik. Variabel kategorikal memiliki tipe data objek dan variabel numerik memiliki tipe data int64.

```
# mencari variabel kategorikal
categorical = [var for var in data_covid.columns if data_covid[var].dtype=='O']
print('Berikut ini {} variabel kategorikal \n'.format(len(categorical)))
print('Variabel kategori adalah :\n\n', categorical)
```

```
Berikut ini 3 variabel kategorikal

Variabel kategori adalah :

 ['Bulan', 'Kecamatan', 'gender']
```

```
# mencari variabel numerikal
numerical = [var for var in data_covid.columns if data_covid[var].dtype!='O']
print('Berikut ini {} varuabel numerikal \n'.format(len(numerical)))
print('Varuabel numerik adalah :', numerical)
```

```
Berikut ini 5 varuabel numerikal

Varuabel numerik adalah : ['NO.', 'positif', 'rawat', 'Sembuh', 'Meninggal']
```

Gambar 3. Variabel kategorikal dan variabel numerikal

4.4. Mendeklarasikan vektor fitur dan variabel target, dengan perintah :

```
# menentukan vektor fitur dan variabel target
X = data_covid.drop(['positif'], axis=1)
y = data_covid['positif']
```

Menampilkan variabel kategori dan numerik secara terpisah, berikut ini baris perintah dan hasilnya pada gambar 4 :

```
# Tampilkan variabel kategorikal
categorical = [col for col in X_train.columns if X_train[col].dtypes == 'O']
categorical
```

```
['Bulan', 'Kecamatan', 'gender']
```

```
# tampilkan variabel numerikal
numerical = [col for col in X_train.columns if X_train[col].dtypes != 'O']
numerical
```

```
['NO.', 'rawat', 'Sembuh', 'Meninggal']
```

Gambar 4. Variabel kategorikal dan variabel numerikal

4.5. Memisahkan dataset menjadi *training* dan *testing set* dengan perintah :

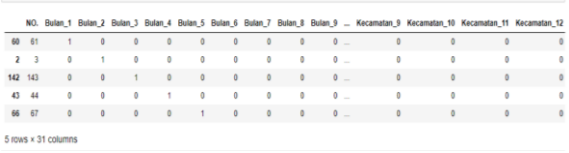
```
# Membagi dataset menjadi training set dan testing set
from sklearn. model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y,
test_size = 0.3, random_state = 0)
```

Perintah diatas membagi dataset menjadi 70% *train set* dan 30% *test set*. Sehingga dari 156 data, 109 adalah *train set* dan 37 merupakan *test set*.

4.6. Rekayasa fitur

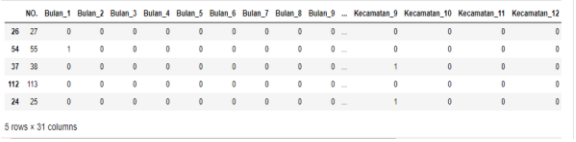
Rekayasa fitur merupakan suatu proses pengolahan data mentah menjadi suatu fitur yang berguna agar dapat memahami model dengan lebih baik dan meningkatkan daya prediksinya. Berikut ini langkah-langkahnya:

1. *Encode* variabel kategorikal (X_train)



Gambar 5. Hasil *encode* variabel kategorikal (X_train)

2. *Encode* variabel kategorikal X_test

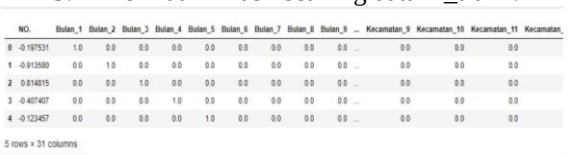


Gambar 6. Hasil *encode* variabel kategorikal (X_test)

Setelah selesai menentukan *train set* dan *test set* yang siap untuk pembuatan model. Langkah selanjutnya memetakan semua variabel fitur ke dalam skala yang sama. Ini disebut penskalaan fitur.

4.7. Pengskalaan fitur (*Feature Scaling*), adapun tahapan prosesnya sebagai berikut :

1. Mendefinisikan kolom X_train
2. Melakukan *scalling* terhadap X_train dan X_test
3. Berikut ini hasil scalling data X_train :



Gambar 7. Hasil *scalling* X_train

4.8. Analisis data menggunakan *Naive Bayes Classifier*

```
# input data
data_covid=pd.read_excel('data_train.xlsx')
# Menampilkan data
data_covid.head(10)
```

	positif	rawat	Sembuh	Meninggal	kasus
0	152	4	3	1	>=30
1	122	2	2	0	<30
2	117	3	3	0	>=30
3	132	3	1	2	>=30
4	144	1	2	0	<30
5	33	1	1	0	<30
6	28	0	0	0	>=30
7	27	0	0	0	>=30
8	42	0	0	0	>=30
9	39	0	0	0	>=30

Gambar 8. Dataset yang akan diolah

Kemudian menentukan variabel independen dan dependen

```
# Variabel independen
x = data_covid.drop(["kasus"], axis = 1)
x.head()
```

	positif	rawat	Sembuh	Meninggal
0	152	4	3	1
1	122	2	2	0
2	117	3	3	0
3	132	3	1	2
4	144	1	2	0

```
# Variabel dependen
y = data_covid["kasus"]
y.head()
```

```
0    >=30
1     <30
2    >=30
3    >=30
4     <30
Name: kasus, dtype: object
```

Gambar 9. Variabel independen dan dependen

Menghitung model gaussian NB(), menentukan x_train dan x_test dan mencari nilai prediksi.

```
# Import train_test_split function
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, random_state = 123)
```

```
# Import Gaussian Naive Bayes model
from sklearn.naive_bayes import GaussianNB
# Mengaktifkan/memanggil/membuat fungsi klasifikasi Naive bayes
modelnb = GaussianNB()
# Memasukkan data training pada fungsi klasifikasi naive bayes
nbtrain = modelnb.fit(x_train, y_train)
nbtrain.class_count_
```

```
array([ 11., 113.])
```

```
# Menentukan hasil prediksi dari x_test
y_pred = nbtrain.predict(x_test)
y_pred
```

```
array(['>=30', '>=30', '>=30', '>=30', '>=30', '>=30', '>=30',
       '>=30', '>=30', '>=30', '>=30', '>=30', '>=30', '>=30',
       '>=30', '>=30', '>=30', '>=30', '>=30', '>=30', '>=30',
       '>=30', '>=30', '>=30', '>=30', '>=30', '>=30', '>=30'],
      dtype='<U4')
```

Gambar 10. Hitung model gaussianNB ()

Selanjutnya menghitung nilai matrix model

```
# import confusion_matrix model
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test, y_pred)
```

```
array([[ 0,  1],
       [ 0, 31]], dtype=int64)
```

```
# Merapikan hasil confusion matrix
y_actual1 = pd.Series([1, 0,1,0,1,0,1,0,1,0,0,1,1,0,1,1,0,0], name = "actual")
y_pred1 = pd.Series([1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1], name = "prediction")
data_covid_confusion = pd.crosstab(y_actual1, y_pred1)
data_covid_confusion
```

prediction	0	1
actual		
0	7	2
1	1	8

Gambar 11. Menghitung matrix model

4.9. Merapikan nilai matrix model dan menghitung nilai akurasi pada Naive *Bayes*

```
# Merapikan hasil confusion matrix
y_actual1 = pd.Series([1, 0,1,0,1,0,1,0,1,0,0,1,1,0,1,1,0,0], name = "actual")
y_pred1 = pd.Series([1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1], name = "prediction")
data_covid_confusion = pd.crosstab(y_actual1, y_pred1)
data_covid_confusion
```

actual \ prediction	0	1
0	7	2
1	1	8

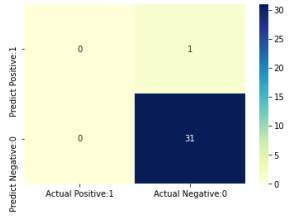
```
# Menghitung nilai akurasi dari klasifikasi naive bayes
from sklearn.metrics import classification_report
print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
<30	0.00	0.00	0.00	1
>=30	0.97	1.00	0.98	31
accuracy			0.97	32
macro avg	0.48	0.50	0.49	32
weighted avg	0.94	0.97	0.95	32

Gambar 12. Menghitung nilai akurasi klasifikasi *Naive Bayes*

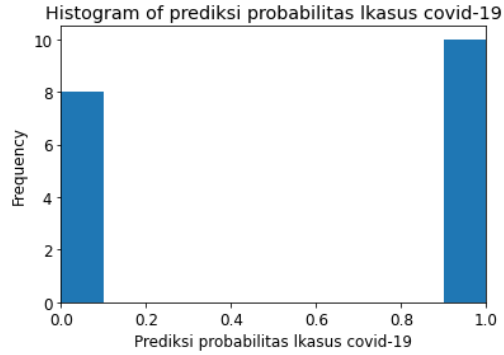
```
# visualize confusion matrix with seaborn heatmap

cm_matrix = pd.DataFrame(data=cm, columns=['Actual Positive:1', 'Actual Negative:0'],
                                  index=['Predict Positive:1', 'Predict Negative:0'])

sns.heatmap(cm_matrix, annot=True, fmt='d', cmap='YlGnBu')
```

```
<AxesSubplot:>
```



Gambar 13. C*onfusion matrix* akurasi klasifikasi *Naive Bayes*

```
Text(0, 0.5, 'Frequency')
```



Gambar 14. Histogram prediksi klasifikasi kasus covid

5. KESIMPULAN DAN SARAN

Dalam tugas akhir ini, model *Naive Bayes Classifier* memprediksi bahwa kasus covid-19 di Pamekasan tinggi (>= 30 orang) dan model *Gaussian Naive Bayes* menunjukkan kinerja yang sangat baik dengan nilai akurasi model sebesar 0,9688 serta nilai *precision* (0,97), *recall* (1,00) dan *f-1 score* (0,98).

DAFTAR PUSTAKA

[1] Bengnga, A., & Ishak, R. (2018). Prediksi Jumlah Mahasiswa Registrasi Per Semester Menggunakan Linier Regresi Pada Universitas Ichsan Gorontalo. ILKOM Jurnal Ilmiah, 10(2), 136-143. doi:https://doi.org/10.33096/ilkom.v10i2.274.136-143

[2] Samsir, S., Ambiyar, A., Verawardina, U., Edi, F., & Watrianthos, R. (2021). Analisis Sentimen Pembelajaran Daring Pada Twitter di Masa Pandemi COVID-19 Menggunakan Metode Naïve Bayes. Jurnal Media Informatika Budidarma, 5(1), 157-163.

[3] Hartini, D., & Informasi, S. (n.d.). Implementasi Data Mining Rought Set Dalam Menganalisis Kinerja Dosen. Jurnal Ilmiah Binary STMIK Bina Nusantara Jaya. Vol 1 No 2.2019. DOI: https://doi.org/10.52303/jb.v1i2.11

[4] Suarpurningsih, N., Utami, N., & Estiyanti, N. (2022). Klasifikasi Penentuan Kelayakan Pemberian Kredit Menggunakan Metode Naive Bayes Classifier (Kasus: Koperasi Simpan Pinjam Artha Segara). J-SAKTI (Jurnal Sains Komputer dan Informatika), 6(1), 391-404. doi:http://dx.doi.org/10.30645/j-sakti.v6i1.454

[5] tan, F., & ELISA, E. Pemanfaatan Data Mining Clustering Dalam Penentuan Media Promosi Umkm Di Kota Batam. Computer and Science Industrial Engineering (COMASIE), 4(1), 59–65. 2021.

[6] Laia, Seberlina., Pane, Deski Helsa., Affandi Egi. Implementasi Metode K-Means Untuk Mengelompokkan Kawasan Potensi Pertanian Karet Produktif. Jurnal Ilkom. Vol 1, No 4 (2022). DOI: https://doi.org/10.53513/jursi.v1i4.5224

[7] Nemati, H. R., & Barko, C. D. (Eds.). (2004). Organizational data mining: leveraging enterprise data resources for optimal performance. IGI Global.

[8] Idris, Muhammad. Implementasi Data Mining Dengan Algoritma Naïve Bayes Untuk Memprediksi Angka Kelahiran. Pelita Informatika. Vol 7, No 3. 2019.

[9] Hikmah, N. (2017). Pemanfaatan Metode Naïve Bayes Classifier dalam Pembuatan Sistem Pakar untuk Diagnosa Penyakit Kelamin. Energy-Jurnal Ilmiah Ilmu-Ilmu Teknik, 7(2), 50-55.

[10] Idris, M. 2019. Implementasi Data Mining Dengan Algoritma Naïve Bayes Untuk Memprediksi Angka Kelahiran. Pelita Informatika: Informasi dan Informatika, 7(3), 421-428.)

[11] Socs.binus. Diakses pada tanggal : 23 Mei 2022. dari:https://socs.binus.ac.id/2012/03/12/python-untuk-scientific-computing/