

Analisis Tingkat Akurasi Prediksi Gejala COVID-19 Dengan Menggunakan Metode Logistic Regression dan Support Vector Machine

Briandy Tri Putra¹, Evi Yulianingsih², Fatmasari³, Ferdiansyah⁴

^{1,2,3,4}Sistem Informasi, Sains Teknologi, Universitas Bina Darma

¹191410085@student.binadarma.ac.id, ²ev_yulianingsih@binadarma.ac.id, ³fatmasari@binadarma.ac.id,
⁴ferdi@binadarma.ac.id

Abstract

One of the computer science technologies that are programmed to learn and perform human-like activities is artificial intelligence. Artificial intelligence technology has been used in several fields, one of which is in the health sector. In the health sector, artificial intelligence is used as a tool to detect diseases in humans, one example is predicting the early symptoms of COVID-19 is one of the SARS-CoV2 infectious diseases that caused a worldwide pandemic, and the virus was first detected from wild animals in the Wuhan City market, China in late 2019. In previous research entitled "Classification Method of Coronavirus Disease 19 (COVID-19) Symptoms Using Neural Network Algorithm" by Rahmi, et al, using data on COVID-19 symptoms to get an accuracy rate in predicting COVID-19 using Neural Network and Logistic Regression methods. The results of this study obtained an accuracy rate of 95% with the Neural Network method, and 94% with the Logistic Regression method. In this study, the authors want to compare the Logistic Regression method with Support Vector Machine in predicting the early symptoms of COVID-19. The result of this research is to get the highest level of accuracy of the two methods to predict the symptoms of COVID-19. Logistic Regression has an accuracy rate of 0.95%, while the Support Vector Machine Kernel Linear method has an accuracy of 0.95%, Support Vector Machine Kernel Sigmoid 0.91%, and Support Vector Machine Kernel RBF has an accuracy of 0.95%. In this study, the Sigmoid Kernel Support Vector Machine has a low level of accuracy, so it is not recommended for predicting COVID-19 symptoms.

Keywords: covid19, logistic regression, SVM

Abstrak

Salah satu teknologi ilmu komputer yang deprogram untuk mempelajari dan melakukan aktivitas seperti manusia adalah kecerdasan buatan. Teknologi kecerdasan buatan telah dipakai pada beberapa bidang, salah satunya di bidang kesehatan. Dibidang kesehatan, kecerdasan buatan digunakan sebagai alat untuk mendeteksi penyakit pada manusia, salah satu contohnya adalah memprediksi gejala awal COVID-19 merupakan salah satu penyakit menular SARS-CoV2 yang menyebabkan pandemi di seluruh dunia, dan virus tersebut terdeteksi pertama kali dari hewan-hewan liar di pasar Kota Wuhan, China pada akhir 2019. Pada penelitian sebelumnya yang berjudul Metode Klasifikasi Gejala Penyakit *Coronavirus Disease 19 (COVID-19)* Menggunakan Algoritma *Neural Network* oleh Rahmi, dkk, menggunakan data gejala-gejala COVID-19 untuk mendapatkan tingkat akurasi dalam prediksi COVID-19 menggunakan metode *Neural Network* dan *Logistic Regression*. Hasil penelitian tersebut mendapatkan tingkat akurasi sebesar 95% dengan metode *Neural Network*, dan 94% dengan metode *Logistic Regression*. Pada penelitian ini, penulis ingin membandingkan metode *Logistic Regression* dengan *Support Vector Machine* dalam memprediksi gejala awal COVID-19. Hasil dari penelitian ini adalah mendapatkan akurasi dengan tingkat yang tertinggi dari kedua metode tersebut untuk memprediksi gejala COVID-19. *Logistic Regression* memiliki tingkat akurasi sebesar 0.95%, sementara metode *Support Vector Machine Kernel Linear* memiliki akurasi sebesar 0.95%, *Support Vector Machine Kernel Sigmoid* 0.91%, dan *Support Vector Machine Kernel RBF* memiliki akurasi sebesar 0.95%. Pada penelitian ini, *Support Vector Machine Kernel Sigmoid* memiliki tingkat akurasi yang rendah, sehingga tidak direkomendasikan untuk prediksi gejala COVID-19.

Kata kunci: covid19, logistic regression, SVM

©This work is licensed under a Creative Commons Attribution - ShareAlike 4.0 International

1. Pendahuluan

Kecerdasan buatan merupakan ilmu komputer dengan pengalaman dan pengetahuan sama seperti manusia. Kecerdasan buatan telah banyak diterapkan diberbagai teknologi seperti robot, alat-alat membantu keseharian manusia, hingga aplikasi. Saat ini, para ilmuwan berlomba-lomba untuk membuat sebuah penelitian terkait mesin yang dapat memprediksi sesuatu akan terjadi, contohnya adalah dapat memprediksi harga saham akan naik atau turun. Pada penelitian yang berjudul “A LSTM-Method for Bitcoin Price Prediction: A Case Study Yahoo Finance Stock

Market” oleh Ferdiansyah, dkk, melakukan penelitian kecerdasan buatan terkait dengan prediksi harga *bitcoin* menggunakan metode *long-short-term-memory*. Penelitian tersebut menjelaskan bahwa *bitcoin* merupakan fluktuasi harga harian dan selalu berubah setiap hari, yang terkadang nilai tukarnya terus naik hingga turun secara drastis. Hal ini karena pasar saham dipengaruhi oleh banyak ketidakpastian faktor seperti masalah ekonomi dan politik memiliki dampak ke tingkat global. Untuk mengatasi masalah tersebut membutuhkan sebuah alat prediksi yang dapat membantu investor untuk memutuskan memilih saham *bitcoin* atau *cryptocurrency*. Hasil dari penelitian ini

adalah peneliti berhasil membuat sebuah model yang menyediakan hasil prediksi *bitcoin* dari pasar saham *Yahoo Finance* [1].

Selain itu, kecerdasan buatan telah banyak dikembangkan di berbagai bidang, salah satunya adalah bidang kesehatan yang digunakan sebagai alat pendekripsi penyakit. Seluruh rumah sakit saat ini berteknologi kecerdasan buatan untuk mendiagnosis dan memprediksi kondisi terhadap pasien, jika nantinya akan terjadi kondisi yang lebih parah. Selain itu, kecerdasan buatan merupakan solusi untuk meningkatkan layanan kesehatan yang lebih efektif dan efisien. Hingga tahun 2019, teknologi kecerdasan buatan dikembangkan untuk mendeteksi penyakit *COVID-19*. Hingga tahun 2019, teknologi kecerdasan buatan dikembangkan untuk mendeteksi penyakit *COVID-19* [2]. *Corona disease virus 2019*, atau *COVID-19* adalah salah satu *coronavirus* dengan penyebabnya *Severe Acute Respiratory Syndrom-CoV-2 (SARS-CoV-2)*. Pasar Huanan merupakan tempat pertama kali diduga adanya virus ini, yang terletak pada wilayah Huanan, di China. Setiap orang memiliki kasus yang berbeda-beda ketika terserang penyakit *COVID-19*, dimulai dari gejala ringan hingga gejala parah. Gejala-gejala umum *COVID-19* yaitu kelelahan, demam, batuk, hingga kehilangan penciuman dan perasa [3].

Pada penelitian sebelumnya yang berjudul “Metode Klasifikasi Gejala Penyakit *Coronavirus Disease 19 (COVID-19)* Menggunakan Algoritma *Neural Network*” oleh Rahmi, dkk, menggunakan *dataset* tersebut sebagai penelitian untuk mendapatkan tingkat akurasi untuk memprediksi *COVID-19* berdasarkan dari gejala-gejala di dalam dataset tersebut. Metode yang digunakan adalah metode *Neural Network* merupakan salah satu metode yang digunakan untuk klasifikasi *machine learning*. Hasil dari penelitian yang didapatkan adalah hasil tingkat akurasi metode *Neural Network* sebesar 95% [4]. Sementara dari penelitian tersebut, peneliti ingin melakukan penelitian baru untuk melakukan perbandingan antara metode *Logistic Regression* dengan *Support Vector Machine*. Metode *Logistic Regression* merupakan salah satu metode statisika yang digunakan sebagai algoritma klasifikasi analisis data yang digunakan sebagai bentuk model untuk membuat sistem prediksi [5]. Sementara, metode *Support Vector Machine* merupakan metode yang sama dengan *Logistic Regression*, namun perbedaannya adalah *Support Vector Machine* menggunakan sebuah *Kernel Trick* untuk memaksimalkan hasil tingkat akurasi yang akan didapatkan [6].

Tujuan untuk melakukan penelitian ini adalah untuk melakukan perbandingan antara metode *Logistic Regression* dan *Support Vector Machine*. Namun, setelah didapatkan hasil dari kedua metode tersebut akan dilakukan perbandingan lagi dengan penelitian terdahulu yang menggunakan metode *Neural Network*. *Dataset* yang digunakan dalam penelitian ini adalah dataset yang sama digunakan pada penelitian terdahulu,

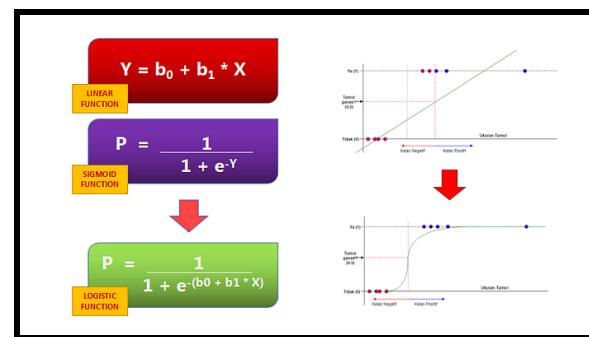
yakni *dataset* berisi pasien mengalami gejala-gejala *COVID-19* yang bersumber dari website <http://github.com/nshomron/covidpred>. Dari kedua metode tersebut, penulis ingin melakukan sebuah penelitian yang berjudul “Analisis Tingkat Akurasi Prediksi Gejala *Covid-19* Dengan Menggunakan Metode *Logistic Regression* Dan *Support Vector Machine*”. Hasil penelitian ini adalah untuk mendapatkan dan membandingkan tingkat akurasi tertinggi diantara metode *Logistic Regression* dan *Support Vector Machine* dalam memprediksi gejala-gejala awal *COVID-19*.

2. Metode Penelitian

Metode *Logistic Regression* merupakan salah satu metode statisika yang digunakan sebagai algoritma klasifikasi analisis data yang digunakan sebagai bentuk model untuk membuat sistem prediksi [5]. Sementara, metode *Support Vector Machine* merupakan metode yang sama dengan *Logistic Regression*, namun perbedaannya adalah *Support Vector Machine* menggunakan sebuah *Kernel Trick* untuk memaksimalkan hasil tingkat akurasi yang akan didapatkan [6].

2.1. Logistic Regression

Logistic Regression merupakan sebuah metode atau algoritma yang digunakan sebagai klasifikasi yang menggabungkan variabel respon dan variabel prediksi dengan menghasilkan *output* probabilitas tertentu [7].



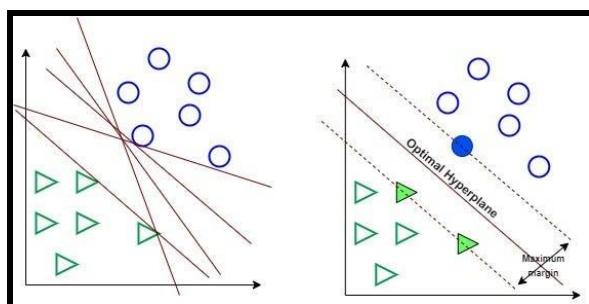
Gambar 1. Rumus *Logistic Regression* (*Logistic Function*)

Logistic Regression memiliki rumus yang diberi nama sebagai “*Logistic Function*” yang merupakan sebuah fungsi dimana fungsi ini dibentuk dari *Linear Function* dan *Sigmoid Function* dengan menyamakan nilai Y dari *Linear Function* dan nilai Y dari *Sigmoid Function* [7]. Bentuk dari *Logistic Function* adalah sebagai berikut:

1. Melakukan invers pada *Sigmoid Function* sehingga berubah bentuk menjadi $Y = \ln(p/(1-p))$.
2. Menyamakan fungsi *Linear Function* $Y = b_0 + b_1 * X$ menjadi bentuk $\ln(p/(1-p)) = b_0 + b_1 * X$.
3. Melakukan persamaan $\ln(p/(1-p)) = b_0 + b_1 * X$ menjadi bentuk logaritmik sehingga menjadi persamaan $P = 1/(1+e^{-(b_0+b_1*X)})$.

2.2. Support Vector Machine

Support Vector Machine merupakan sebuah model atau algoritma *machine learning* yang digunakan sebagai klasifikasi dan regresi untuk data yang tidak beraturan. Metode ini merupakan bagian dari *supervised learning* yang sering juga digunakan sebagai analisis data [8]. Tujuan dari *SVM* adalah untuk menemukan *hyperplane* yang terbaik sebagai pemisah data input di dalam ruang berdimensi. *Hyperplane* merupakan sebuah garis batas yang memisahkan data tidak beraturan [6].



Gambar 2. Grafik Support Vector Machine

Pada gambar 2, gambar sebelah kiri terdapat beberapa *hyperplane* yang terlihat memisahkan data segitiga dan lingkaran secara tidak beraturan. Sehingga, pada gambar sebelah kanan, setelah menggunakan algoritma *Support Vector Machine* yang akan membuat *hyperplane* memisahkan kedua data secara beraturan. Untuk memaksimalkan *hyperplane*, digunakan sebuah *Kernel Trick* yang dapat mengubah ruangan dimensi rendah menjadi dimensi tinggi. Fungsi *Kernel Trick* yang dapat digunakan diantaranya yakni *Linear*, *Sigmoid*, dan *RBF* [9].

2.3. Dataset

Dataset merupakan sekumpulan data yang berisi berbagai informasi terkait topik tertentu, yang sering kali digunakan sebagai data untuk membuat sebuah model atau algoritma dari *machine learning* [10]. Dataset yang digunakan dalam penelitian ini adalah dataset berisi pasien dengan gejala-gejala *COVID-19* yang digunakan pada penelitian sebelumnya yang berjudul “Metode Klasifikasi Gejala Penyakit Coronavirus Disease 19 (*COVID-19*) Menggunakan Algoritma *Neural Network*” oleh Rahmi, dkk, bersumber dari website <http://github.com/nshomron/covidpred>.

Tabel 1. Dataset Pasien Berdasarkan Gejala-Gejala COVID-19

Tahun 2022

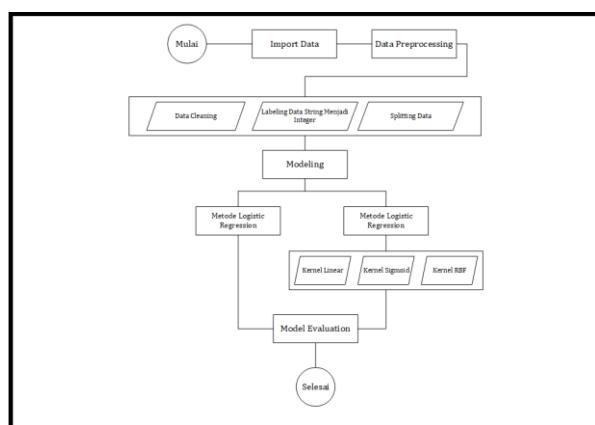
Gejala	Negatif	Positif
Batuk	236.368 pasien	42.228 pasien
Demam	256.844 pasien	21.752 pasien
Sakit Tenggorokan	276.291 pasien	1.926 pasien
Sesak Nafas	277.270 pasien	1.577 pasien
Sakit Kepala	276.613	2.414 pasien

Pasien yang berumur 60 tahun atau lebih	127.703 pasien	25.825 pasien
Jenis Kelamin	130.158 pasien	129.127 pasien
Tes Indikasi	242.741 pasien	36.107 pasien
Hasil Tes Corona	260.277 pasien	14.729 pasien

Pada tabel 1, *dataset* yang digunakan merupakan dataset berasal dari website *Israeli Ministry of Health* merupakan sebuah website Pemerintahan Israel yang menyediakan berbagai *dataset* tentang kesehatan, dapat diakses oleh seluruh orang untuk mengembangkan sebuah *machine learning*. Namun, *dataset* yang secara langsung diunduh dari website tersebut masih memiliki Bahasa Ibrani sehingga harus diubah menjadi Bahasa Inggris atau Bahasa Indonesia untuk memudahkan klasifikasi pada penelitian ni. Oleh karena itu, terdapat sebuah situs menyediakan *dataset* tersebut yang telah diterjemahkan ke dalam Bahasa Inggris dari website *GitHub* dengan alamat <http://github.com/nshomron/covidpred>.

2.4. Tahapan Penelitian

Metode *Logistic Regression* dan *Support Vector Machine* memiliki tahapan-tahapan yang sama, yang terdiri dari *Import Data*, *Data Preprocessing*, *Modelling*, dan *Model Evaluation*.



Gambar 3. Tahapan Penelitian

1. *Import Data* merupakan tahapan awal untuk memasukkan *dataset* ke dalam alat pengolahan data.

2. *Data Preprocessing* merupakan tahapan kedua untuk mengolah *dataset* yang tidak beraturan menjadi lebih beraturan. Tahapan *data preprocessing* terbagi menjadi tiga bagian sebagai berikut:

a. Data Cleaning

Proses ini dilakukan untuk pembersihan data dibagi menjadi tiga proses yang terdiri dari penghapusan data kosong, dan penghapusan kolom yang tidak digunakan.

b. Labeling Data String Menjadi Integer

Proses ini melakukan perubahan pada kolom yang memiliki tipe data *string* menjadi tipe data *integer*, karena kecerdasan buatan hanya dapat membaca isi data yang bertipe *integer* terdiri dari angka “0” dan “1”.

c. Splitting Data

Splitting Data merupakan proses untuk melakukan pemisahan *dataset* terdiri dari *Train Data* dan *Test Data*.

3. *Modeling* merupakan tahapan ketiga untuk memasukkan metode sebagai pola prediksi dari *dataset* yang digunakan. Dalam penelitian ini menggunakan metode *Logistic Regression* dan *Support Vector Machine*. *Support Vector Machine* terdiri dari tiga *kernel* yang digunakan yaitu *Kernel Linear*, *Sigmoid*, dan *RBF*.

4. *Model Evaluation* merupakan tahapan akhir untuk melakukan pengujian dari *dataset* dan metode yang telah digunakan. Tahapan ini menggunakan *Confusion Matrix* untuk evaluasi tingkat akurasi yang didapatkan.

3. Hasil dan Pembahasan

3.1. Dataset

Dataset yang digunakan merupakan *dataset* pasien yang pernah mengalami gejala-gejala *COVID-19* diunduh dari situs *Israeli Ministry of Health* di <https://data.gov.il/dataset/covid-19>.

#	test_date	cough	fever	sore_throat	shortness_of_breath	head_ache	corona_result	age_60_and_above	gender	test_indication
0	2020-04-30	0	0	0	0	0	negative	None	female	Other
1	2020-04-30	1	0	0	0	0	negative	None	female	Other
2	2020-04-30	0	1	0	0	0	negative	None	male	Other
3	2020-04-30	1	0	0	0	0	negative	None	female	Other
4	2020-04-30	1	0	0	0	0	negative	None	male	Other
...
27843	2020-03-11	0	0	0	0	0	negative	None	None	Other
27844	2020-03-11	0	0	0	0	0	negative	None	None	Other
27845	2020-03-11	0	0	0	0	0	positive	None	None	Contact with confirmed
27846	2020-03-11	0	0	0	0	0	other	None	None	Other
27847	2020-03-11	0	0	0	0	0	negative	None	None	Other

Gambar 4. Isi Dataset

Pada gambar 4, *dataset* ini berisi 10 kolom yakni *test_date* (waktu tes), *cough* (batuk), *fever* (demam), *sore_throat* (sakit tenggorokan), *shortness_of_breath* (sesak napas), *head_ache* (sakit kepala), *corona_result* (hasil tes *COVID-19*), *age_60_and_above* (pasien yang berumur 60 tahun atau lebih), *gender* (jenis kelamin), dan *test_indication* (tes indikasi).

```
[87] # menampilkan ukuran isi dataset
df.shape

(278848, 10)

[88] # menampilkan tipe data pada setiap atribut
df.dtypes

test_date          object
cough             object
fever             object
sore_throat       object
shortness_of_breath    object
head_ache         object
corona_result     object
age_60_and_above   object
gender            object
test_indication   object
dtype: object
```

Gambar 5. Total Pasien Dan Tipe Data Pada Dataset

Pada gambar 5, *dataset* ini memiliki total pasien sebanyak 278.848 pasien, dengan tipe data yang digunakan yakni tipe data *object*. Tipe data ini terdiri dari kolom *test_date* (waktu tes), *corona_result* (hasil tes *COVID-19*), *age_60_and_above* (pasien yang berumur 60 tahun atau lebih), *gender* (jenis kelamin),

dan *test_indication* (tes indikasi), kolom *cough* (batuk), *fever* (demam), *sore_throat* (sakit tenggorokan), *shortness_of_breath* (sesak napas), dan *head_ache* (sakit kepala).

df.cough.value_counts()	df.shortness_of_breath.value_counts()	df.age_60_and_above.value_counts()
0 121649	0 261079	None 127320
1 114719	0 16191	No 125703
2 25875	1 1065	Yes 25025
None 16259	None 512	Name: age_60_and_above, dtype: int64
Name: cough, dtype: int64	Name: shortness_of_breath, dtype: int64	

df.fever.value_counts()	df.head_ache.value_counts()	df.gender.value_counts()
0 133942	0 260079	female 130158
1 122992	0 16354	male 129127
2 13582	1 2065	other gender, dtype: int64
None 8170	None 349	
Name: fever, dtype: int64	Name: head_ache, dtype: int64	

df.sore_throat.value_counts()	df.corona_result.value_counts()	df.test_indication.value_counts()
0 208664	negative 269237	Other 242741
1 10257	positive 14729	Absent 2568
2 1480	other 3892	Contact with confirmed 10639
None 446	Name: corona_result, dtype: int64	Name: test_indication, dtype: int64
Name: sore_throat, dtype: int64		

Gambar 6. Data Kosong Dan Duplikasi Data Pada Dataset

Pada gambar 6, *dataset* ini memiliki data yang tidak beraturan, yakni terdapat data “None” pada beberapa kolom yakni kolom *cough* (batuk), *fever* (demam), *sore throat* (sakit tenggorokan), *shortness of breath* (sesak napas), *head ache* (sakit kepala), *age 60 and above* (pasien berumur 60 tahun atau lebih), dan *gender* (jenis kelamin), dan kolom yang memiliki data “other” terdiri dari kolom *corona result* (hasil tes *COVID-19*). Data yang tidak beraturan tersebut sering terjadi di dalam dataset, karena terdapat banyaknya data di dalam dataset. Sementara, setiap kolomnya juga memiliki 4 data, yang seharusnya menjadi 2 data diantara 0 (negatif), dan 1 (positif). Hal ini terjadi karena tipe data pada kolom *cough* (batuk), *fever* (demam), *sore throat* (sakit tenggorokan), *shortness of breath* (sesak napas), *head ache* (sakit kepala) merupakan tipe data *object*, yang seharusnya *int64*.

[98] df.isnull().sum()
test_date 0
cough 0
fever 0
sore_throat 0
shortness_of_breath 0
head_ache 0
corona_result 0
age_60_and_above 0
gender 0
test_indication 0
dtype: int64

[99] df.duplicated().sum()
272068

Gambar 7. Duplikasi Data

Pada gambar 7, *dataset* ini tidak memiliki data yang kosong, namun memiliki duplikasi data sebanyak 272.068. Namun, duplikasi data tersebut tidak perlu dibersihkan karena akan mengurangi banyaknya data di dalam *dataset* ini, sehingga tingkat akurasi yang akan didapatkan memiliki angka yang rendah.

3.2. Pengolahan Data

Proses pengolahan data dilakukan pada tahapan *Data Preprocessing*, merupakan tahapan untuk mengolah dataset yang tidak beraturan menjadi lebih beraturan. Tujuan untuk melakukan *data preprocessing* ini adalah

untuk meningkatkan kualitas data yang lebih akurat. Tahapan ini terdiri dari 3 proses yang harus dilakukan, yaitu proses *data cleaning*, proses *labeling*, *splitting data* dan standarisasi fitur.

3.2.1. Data Cleaning

1) Penghapusan Kolom Yang Tidak Digunakan

Proses penghapusan kolom yang tidak digunakan ini dilakukan pada kolom *test_date* (waktu tes data), karena kolom ini tidak diperlukan untuk melakukan prediksi *COVID-19*.

Program Penghapusan Kolom “*test_date*”

```
# membuang kolom yang tidak terpakai
df.drop(['test_date'], axis=1, inplace=True)
```

	cough	fever	sore_throat	shortness_of_breath	head_ache	corona_result	age_60_and_above	gender	test_indication
0	0	0	0	0	negative	None	female	Other	
1	1	0	0	0	negative	None	female	Other	
2	0	1	0	0	negative	None	male	Other	
3	1	0	0	0	negative	None	female	Other	
4	1	0	0	0	negative	None	male	Other	
...
278843	0	0	0	0	negative	None	None	Other	
278844	0	0	0	0	negative	None	None	Other	
278845	0	0	0	0	positive	None	None	Contact with confirmed	
278846	0	0	0	0	other	None	None	Other	
278847	0	0	0	0	negative	None	None	Other	

Gambar 8. Hasil Setelah Dilakukan Penghapusan Kolom

Pada gambar 8, setelah dilakukan penghapusan kolom *test_date*, maka tersisa 9 kolom yakni *cough* (batuk), *fever* (demam), *sore_throat* (sakit tenggorokan), *shortness_of_breath* (sesak napas), *head_ache* (sakit kepala), *corona_result* (hasil tes *COVID-19*), *age_60_and_above* (pasien yang berumur 60 tahun atau lebih), *gender* (jenis kelamin), dan *test_indication* (tes indikasi).

2) Penghapusan Data “*None*” dan “*other*”

Proses penghapusan data “*None*” dan “*other*” dilakukan pada seluruh kolom yang memiliki kedua data ini, karena data ini merupakan data yang tidak beraturan. Data yang tidak beraturan ini sering terjadi pada dataset, karena banyaknya data di dalam *dataset*. Data yang tidak beraturan ini perlu dihilangkan karena akan menyebabkan tingkat akurasi yang tidak akurat.

Program Penghapusan Data “*None*” dan “*other*”

```
df = df.replace("other", np.nan)
df = df.replace("None", np.nan)
df = df.dropna()
df = df.reset_index(drop=True)
```

Pada code diatas, merupakan *code* untuk melakukan penghapusan data “*None*” dan “*other*” menggunakan perintah *Replace* dengan menargetkan isi datanya. Lalu kemudian menggunakan perintah *Dropna* untuk menghapus data “*None*” dan “*other*” serta menambahkan *Reset_Index* untuk mengatur indeks di dalam *dataset*. Hasil setelah penghapusan data “*None*” dan “*other*” dapat dilihat pada gambar di bawah ini.

[30] df.cough.value_counts()	[31] df.shortness_of_breath.value_counts()	[32] df.age_60_and_above.value_counts()
0 104996	0 13200	No 112593
1 23200	1 2863	Yes 23701
0 6453	1 1027	Name: age_60_and_above, dtype: int64
1 1588	1 34	
Name: cough, dtype: int64	Name: shortness_of_breath, dtype: int64	Name: age_60_and_above, dtype: int64
[33] df.fever.value_counts()	[34] df.head_ache.value_counts()	[35] df.gender.value_counts()
0 116291	0 131398	Male 69153
1 11962	1 2039	Female 67141
0 7342	1 2089	Name: gender, dtype: int64
1 499	1 68	
Name: fever, dtype: int64	Name: head_ache, dtype: int64	Name: gender, dtype: int64

Gambar 9. Hasil Setelah Dilakukan Penghapusan Data “*None*” dan “*other*”

3) Perubahan Tipe Data

Proses ini melakukan perubahan tipe data pada kolom *cough* (batuk), *fever* (batuk), *sore throat* (sakit tenggorokan), *shortness of breath* (sesak napas), *head ache* (sakit kepala) dari *object* menjadi *int64*. Hal ini karena kolom-kolom tersebut memiliki 4 data yang seharusnya menjadi 2 data diantara 0 (negatif) dan 1 (positif). Pada gambar 10 merupakan *code* untuk melakukan proses perubahan tipe data ke *int64*, menggunakan perintah *Astype*.

Program Perubahan Tipe Data

```
df['cough'] = df['cough'].astype(int)
df['fever'] = df['fever'].astype(int)
df['sore_throat'] =
df['sore_throat'].astype(int)
df['shortness_of_breath'] =
df['shortness_of_breath'].astype(int)
df['head_ache'] =
df['head_ache'].astype(int)
```

[40] df.cough.value_counts()	[42] df.sore_throat.value_counts()
0 111443	0 134821
1 24851	1 1473
Name: cough, dtype: int64	Name: sore_throat, dtype: int64
[41] df.fever.value_counts()	[43] df.shortness_of_breath.value_counts()
0 123633	0 135233
1 12661	1 1061
Name: fever, dtype: int64	Name: shortness_of_breath, dtype: int64
[42] df.sore_throat.value_counts()	[44] df.head_ache.value_counts()
0 134821	0 134219
1 1473	1 2075
Name: sore_throat, dtype: int64	Name: head_ache, dtype: int64

Gambar 10. Hasil Setelah Dilakukan Perubahan Tipe Data

Pada gambar 10 merupakan hasil setelah dilakukan perubahan Tipe Data, yang sebelumnya dalam setiap kolom memiliki 4 data tergabung menjadi 2 data diantara angka “0” merupakan negatif, dan “1” merupakan positif.

3.2.2. Labeling Data String Menjadi Integer

Proses ini dilakukan perubahan pada kolom yang memiliki isi data dengan tipe *string* menjadi tipe *integer*. Pada dasarnya kecerdasan buatan hanya dapat membaca isi data yang bertipe integer terdiri dari angka “0” yang diartikan sebagai status negatif dan “1” diartikan sebagai status positif. Pada gambar 11 merupakan *code* untuk melabel kolom *age 60 and above* (pasien berumur 60 tahun atau lebih), *gender* (jenis kelamin), *test indication* (hasil tes indikasi), dan *corona result* (hasil *COVID-19*), dengan menggunakan perintah *Replace*.

Program Labeling Data

```
# memindahkan kolom corona_result ke kolom terakhir
last_column = df.pop('corona_result')
df.insert(8, 'corona_result', last_column)

# melabel ulang kolom age_60_and_above dengan angka 0 dan 1
df['age_60_and_above'].replace({'No':0, 'Yes':1}, inplace=True)

# melabel ulang kolom gender dengan angka 0 dan 1
df['gender'].replace({'female':0, 'male':1}, inplace=True)

# melabel ulang kolom test_indication dengan angka 0 dan 1
df['test_indication'].replace({'Other':0, 'Contact with confirmed':1, 'Abroad':1}, inplace=True)

# melabel ulang kolom corona_result dengan angka 0 dan 1
df['corona_result'].replace({'negative':0, 'positive':1}, inplace=True)
```

```
[48] df.age_60_and_above.value_counts() [50] df.test_indication.value_counts()
0    112593          0    114358
1     23701          1    21936
Name: age_60_and_above, dtype: int64      Name: test_indication, dtype: int64

[49] df.gender.value_counts() [51] df.corona_result.value_counts()
1     69153          0    125668
0     67141          1    10626
Name: gender, dtype: int64                Name: corona_result, dtype: int64
```

Gambar 11. Hasil Setelah Dilakukan *Labeling Data*

Pada gambar 11 merupakan hasil setelah dilakukan *Labeling Data* pada 4 kolom dengan isi datanya menjadi angka “0” yang diartikan sebagai status negatif dan “1” diartikan sebagai status positif.

3.2.3. Splitting Data

Splitting Data merupakan proses untuk memisahkan dataset menjadi *Train Data* dan *Test Data* menggunakan *Train Test Split*. *Train Data* merupakan bagian dataset yang digunakan sebagai pola prediksi, sementara *Test Data* merupakan bagian data yang digunakan sebagai target prediksinya.

Program Splitting Data

```
# Memisahkan dataset menjadi Training Data dan Test Data
feature = ['cough', 'fever', 'sore_throat', 'shortness_of_breath', 'head_ache', 'age_60_and_above', 'test_indication']
labels = ['corona_result']
x = df[feature].values
y = df.corona_result.values

x_train, x_test, y_train, y_test =
train_test_split(x, y, test_size=0.33, random_state=42, stratify=y)
```

Code diatas merupakan code untuk memisahkan dataset yang terdiri dari *Train Data* menggunakan 80% dari dataset yang berisi kolom *cough* (batuk), *fever* (demam), *sore_throat* (sakit tenggorokan), *shortness_of_breath* (sesak napas), *head_ache* (sakit kepala), *age_60_and_above* (pasien yang berumur 60 tahun atau lebih), *gender* (jenis kelamin), dan

test_indication (tes indikasi). Sementara, *Test Data* menggunakan sebanyak 20% dari *dataset* yang berisi kolom *corona_result* (hasil tes *COVID-19*).

3.2.4. Standarisasi Fitur

Standarisasi fitur dilakukan normalisasi terhadap *dataset* agar tidak terjadi penyimpangan. Standarisasi fitur ini menggunakan Standard Scaler dari library *Scikit-Learn* yang sudah tersedia di *Google Colaboratory*. Sebelum melakukan standarisasi fitur, dataset yang digunakan harus dilakukan splitting data menjadi dua bagian yaitu *Train Data* dan *Test Data*. *Train Data* menggunakan 80% dari dataset yang berisi kolom *cough* (batuk), *fever* (demam), *sore_throat* (sakit tenggorokan), *shortness_of_breath* (sesak napas), *head_ache* (sakit kepala), *age_60_and_above* (pasien yang berumur 60 tahun atau lebih), *gender* (jenis kelamin), dan *test_indication* (tes indikasi). Sementara, *Test Data* menggunakan sebanyak 30% dari *dataset* yang berisi kolom *corona_result* (hasil tes *COVID-19*). Setelah melakukan pembagian data, selanjutnya melakukan standarisasi fitur dengan *Standard Scaler* menggunakan code seperti di bawah ini.

Program Standarisasi Fitur

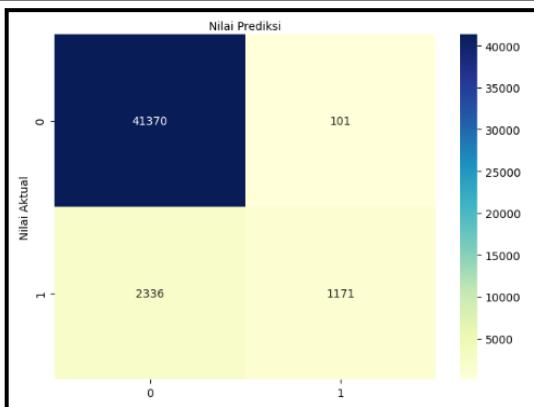
```
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)
```

3.3. Hasil Tingkat Akurasi

Setelah dilakukan proses pengolahan *data preprocessing*, *dataset* yang telah diolah akan dilakukan pemodelan dengan metode *Logistic Regression* dan *Support Vector Machine*. Selanjutnya dilakukan pengujian tingkat akurasi, menggunakan *tools Confusion Matrix*, merupakan sebuah tabel klasifikasi yang melakukan uji benar dan uji jumlah data yang salah. Tabel *Confusion Matrix* sering digunakan untuk menghitung akurasi pada suatu model kecerdasan buatan [11].

1) Logistic Regression

Pengujian tingkat akurasi dilakukan dengan *dataset* yang berisi pasien gejala-gejala *COVID-19* dengan total data sebanyak 136.294 data dengan 9 kolom, yakni kolom *cough* (batuk), *fever* (demam), *sore_throat* (sakit tenggorokan), *shortness_of_breath* (sesak napas), *head_ache* (sakit kepala), *corona_result* (hasil *COVID-19*), *age_60_and_above* (pasien yang berumur 60 tahun atau lebih), *gender* (jenis kelamin), dan *test_indication* (tes indikasi). Peneliti membagi *dataset* menjadi 80% untuk *Train Data* dan 20% untuk *Test Data*. Kemudian digunakan metode *Logistic Regression* sehingga mendapatkan hasil nilai *Confusion Matrix* pada gambar dibawah ini.



Gambar 12. Confusion Matrix Logistic Regression

Pada gambar 12 merupakan diagram *Confusion Matrix* yang didapatkan dari metode *Logistic Regression* dan *dataset* pasien berdasarkan gejala-gejala COVID-19, yang terdiri dari:

1. *True Positive (TP)* merupakan total data prediksinya benar dan realitanya benar, yang berjumlah sebanyak 1.171 data.
2. *True Negative (TN)* merupakan total data prediksinya tidak benar dan realitanya tidak benar, yang berjumlah sebanyak 41.370 data.
3. *False Positive (FP)* merupakan total data prediksinya benar dan realitanya tidak benar yang berjumlah sebanyak 2.336 data.
4. *False Negative (FN)* merupakan jumlah data prediksinya tidak benar dan realitanya benar, yang berjumlah sebanyak 101 data.

Setelah diketahui nilai, tingkat akurasi dapat dihitung dengan rumus *Confusion Matrix*. Pada rumus *Confusion Matrix*, jumlah nilai *True Positive (TP)* dan *True Negative (TN)* dijumlahkan, kemudian dibagi dengan jumlah nilai *True Positive (TP)*, *False Positive (FP)*, *False Negative (FN)*, dan *True Negative (TN)*.

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN} \quad (1)$$

$$= \frac{1171+41370}{1171+2336+101+41370} \quad (2)$$

$$= \frac{42541}{44978} \quad (3)$$

Setelah dilakukan perjumlahan *True Positive (TP)* dan *True Negative (TN)*, didapatkan nilai sebanyak 42541 data, sedangkan perjumlahan *True Positive (TP)*, *False Positive (FP)*, *False Negative (FN)*, dan *True Negative (TN)*, memiliki nilai sebanyak 44978 data. Dari kedua nilai tersebut, akan dibagi dua sehingga akan menghasilkan tingkat akurasi sebagai berikut:

$$= 0.945817955 \quad (4)$$

	precision	recall	f1-score	support
negative covid	0.95	1.00	0.97	41471
positive covid	0.92	0.33	0.49	3507
accuracy			0.95	44978
macro avg	0.93	0.67	0.73	44978
weighted avg	0.94	0.95	0.93	44978

Gambar 13. Hasil Tingkat Akurasi *Logistic Regression* Pada *Google Colaboratory*

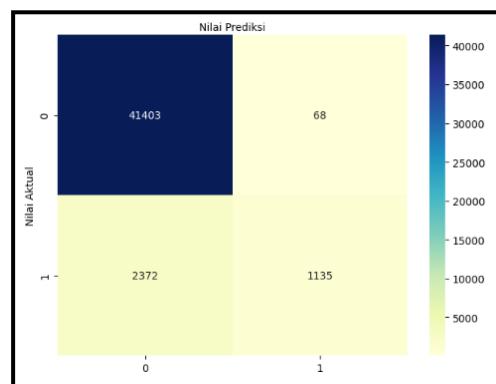
Hasil dari perhitungan diatas mendapatkan nilai sebesar 0.945817955 berdasarkan perhitungan dengan menggunakan rumus *Confusion Matrix*. Pada gambar 13 merupakan hasil tingkat akurasi metode *Logistic Regression* pada *tools Google Colaboratory*, yang dikonversi dalam bentuk pecahan persen menjadi 0.95%, dataset yang digunakan dari *website Israeli Ministry of Health* pada tahun 2022.

2) Support Vector Machine

Pengujian tingkat akurasi dilakukan dengan *dataset* yang berisi pasien gejala-gejala COVID-19 dengan total data sebanyak 136.294 data dengan 9 kolom, yakni kolom *cough* (batuk), *fever* (demam), *sore_throat* (sakit tenggorokan), *shortness_of_breath* (sesak napas), *head_ache* (sakit kepala), *corona_result* (hasil COVID-19), *age_60_and_above* (pasien yang berumur 60 tahun atau lebih), *gender* (jenis kelamin), dan *test_indication* (tes indikasi). Peneliti membagi *dataset* menjadi 80% *Train Data* dan 20% *Test Data*. Kemudian digunakan metode *Support Vector Machine* dengan tiga kernel yakni *Kernel Linear*, *Sigmoid* dan *RBF*.

a. Kernel Linear

Tingkat akurasi yang didapatkan dari metode *Support Vector Machine* dengan menggunakan *Kernel Linear* berdasarkan hasil hitungan dari rumus *Confusion Matrix* pada gambar dibawah ini.



Gambar 14. Confusion Matrix Support Vector Machine Kernel Linear

Pada gambar 14 merupakan diagram *Confusion Matrix* yang didapatkan dari *Support Vector Machine Kernel Linear* dan *dataset* pasien berdasarkan gejala-gejala COVID-19, yang terdiri dari:

1. *True Positive (TP)* merupakan total data prediksinya benar dan realitanya benar, yang berjumlah sebanyak 1171 data.
2. *True Negative (TN)* merupakan total data prediksinya tidak benar dan realitanya tidak benar, yang berjumlah sebanyak 41403 data.
3. *False Positive (FP)* merupakan total data prediksinya benar dan realitanya tidak benar yang berjumlah sebanyak 2372 data.
4. *False Negative (FN)* merupakan total data prediksinya tidak benar dan realitanya benar, yang berjumlah sebanyak 68 data.

Setelah diketahui nilai, tingkat akurasi dapat dihitung dengan rumus Confusion Matrix. Pada rumus Confusion Matrix, jumlah nilai *True Positive (TP)* dan *True Negative (TN)* dijumlahkan, dibagi dengan jumlah nilai *True Positive (TP)*, *False Positive (FP)*, *False Negative (FN)*, dan *True Negative (TN)*. Sehingga rumus diatas akan menjadi nilai seperti dibawah ini:

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN} \quad (1)$$

$$\frac{1135+41403}{1135+2372+68+41403} \quad (2)$$

$$= \frac{42538}{44978} \quad (3)$$

Setelah dilakukan perjumlahan *True Positive (TP)* dan *True Negative (TN)*, didapatkan total nilai sebanyak 42538 data, sedangkan perjumlahan *True Positive (TP)*, *False Positive (FP)*, *False Negative (FN)*, dan *True Negative (TN)*, memiliki nilai sebanyak 44978 data. Dari kedua nilai tersebut, akan dibagi dua sehingga akan menghasilkan tingkat akurasi sebagai berikut:

$$= 0.945751256 \quad (4)$$

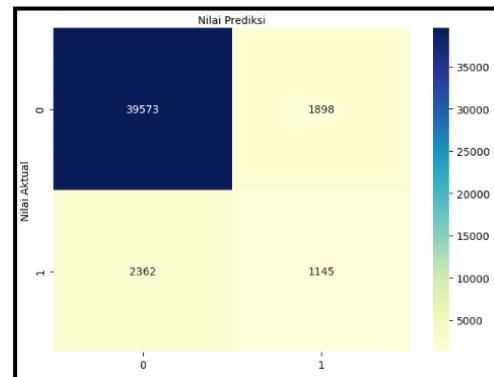
	precision	recall	f1-score	support
negative covid	0.95	1.00	0.97	41471
positive covid	0.94	0.32	0.48	3507
accuracy			0.95	44978
macro avg	0.94	0.66	0.73	44978
weighted avg	0.95	0.95	0.93	44978

Gambar 15. Hasil Tingkat Akurasi *Support Vector Machine Kernel Linear* Pada *Google Colaboratory*

Hasil dari perhitungan diatas mendapatkan nilai sebesar 0.945751256 berdasarkan perhitungan dengan menggunakan rumus Confusion Matrix. Pada gambar 15 merupakan hasil tingkat akurasi metode *Support Vector Machine Kernel Linear* pada *tools Google Colaboratory*, yang dikonversi dalam bentuk pecahan persen menjadi 0.95%, *dataset* yang digunakan dari *website Israeli Ministry of Health* pada tahun 2022.

b. Kernel Sigmoid

Tingkat akurasi yang didapatkan dari *Support Vector Machine Kernel Sigmoid* sebesar 0.95% berdasarkan hasil hitungan dari rumus *Confusion Matrix* pada gambar dibawah ini:



Gambar 16. *Confusion Matrix Support Vector Machine Kernel Sigmoid*

Pada gambar 16 merupakan diagram *Confusion Matrix* yang didapatkan dari *Support Vector Machine Kernel Sigmoid* dan *dataset* pasien berdasarkan gejala-gejala *COVID-19*, yang terdiri dari:

1. *True Positive (TP)* merupakan total data prediksinya benar dan realitanya benar, yang berjumlah sebanyak 1171 data.
2. *True Negative (TN)* merupakan total data prediksinya tidak benar dan realitanya tidak benar, yang berjumlah sebanyak 41403 data.
3. *False Positive (FP)* merupakan total data prediksinya benar dan realitanya tidak benar yang berjumlah sebanyak 2372 data.
4. *False Negative (FN)* merupakan total data prediksinya tidak benar dan realitanya benar, yang berjumlah sebanyak 68 data.

Setelah diketahui nilai, tingkat akurasi dapat dihitung dengan rumus *Confusion Matrix*. Pada rumus Confusion Matrix, total nilai *True Positive (TP)* dan *True Negative (TN)* dijumlahkan, dibagi dengan jumlah nilai *True Positive (TP)*, *False Positive (FP)*, *False Negative (FN)*, dan *True Negative (TN)*. Sehingga rumus diatas akan menjadi nilai seperti dibawah ini:

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN} \quad (1)$$

$$\frac{1145+39573}{1145+2362+1898+39573} \quad (2)$$

$$= \frac{40718}{44978} \quad (3)$$

Setelah dilakukan perjumlahan *True Positive (TP)* dan *True Negative (TN)*, didapatkan nilai sebanyak 40718 data, sedangkan perjumlahan *True Positive (TP)*, *False Positive (FP)*, *False Negative (FN)*, dan *True Negative (TN)*, memiliki nilai sebanyak 44978 data. Dari kedua

nilai tersebut, akan dibagi dua sehingga akan menghasilkan tingkat akurasi sebagai berikut:

$$= 0.905287029 \quad (4)$$

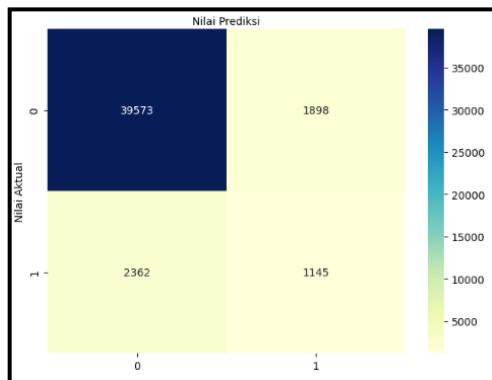
	precision	recall	f1-score	support
negative covid	0.94	0.95	0.95	41471
positive covid	0.38	0.33	0.35	3507
accuracy			0.91	44978
macro avg	0.66	0.64	0.65	44978
weighted avg	0.90	0.91	0.90	44978

Gambar 17. Hasil Tingkat Akurasi Support Vector Machine Kernel Sigmoid Pada Google Colaboratory

Hasil dari perhitungan diatas mendapatkan nilai sebesar 0.905287029 berdasarkan perhitungan dengan menggunakan rumus *Confusion Matrix*. Pada gambar 17 merupakan hasil tingkat akurasi metode *Support Vector Machine Kernel Sigmoid* pada *tools Google Colaboratory*, yang dikonversi dalam bentuk pecahan persen menjadi 0.91%, *dataset* yang digunakan dari *website Israeli Ministry of Health* pada tahun 2022.

c. Kernel RBF

Tingkat akurasi yang didapatkan dari metode *Support Vector Machine* dengan menggunakan *Kernel RBF* sebesar 0.97% berdasarkan hasil hitungan dari rumus *Confusion Matrix* pada gambar dibawah ini :



Gambar 18. Confusion Matrix Support Vector Machine Kernel RBF

Pada gambar 18 merupakan diagram *Confusion Matrix* yang didapatkan dari metode *Support Vector Machine Kernel RBF* dan *dataset* pasien berdasarkan gejala-gejala *COVID-19*, yang terdiri dari:

1. *True Positive (TP)* merupakan total data prediksinya benar dan realitanya benar, yang berjumlah sebanyak 1305 data.
2. *True Negative (TN)* merupakan total data prediksinya tidak benar dan realitanya tidak benar, yang berjumlah sebanyak 71395 data.
3. *False Positive (FP)* merupakan total data prediksinya benar dan realitanya tidak benar yang berjumlah sebanyak 2.034 data.
4. *False Negative (FN)* merupakan total data prediksinya tidak benar dan realitanya benar, yang berjumlah sebanyak 325 data.

Setelah diketahui nilai, tingkat akurasi dapat dihitung dengan rumus *Confusion Matrix*. Pada rumus *Confusion Matrix*, total nilai *True Positive (TP)* dan *True Negative (TN)* dijumlahkan, dibagi dengan jumlah nilai *True Positive (TP)*, *False Positive (FP)*, *False Negative (FN)*, dan *True Negative (TN)*. Sehingga rumus diatas akan menjadi nilai seperti dibawah ini:

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN} \quad (1)$$

$$= \frac{1276+41283}{1276+2231+188+41238} \quad (2)$$

$$= \frac{42559}{44978} \quad (3)$$

Setelah dilakukan perjumlahan *True Positive (TP)* dan *True Negative (TN)*, didapatkan nilai sebanyak 71.452 data, sedangkan perjumlahan *True Positive (TP)*, *False Positive (FP)*, *False Negative (FN)*, dan *True Negative (TN)*, memiliki nilai sebanyak 75.059 data. Dari kedua nilai tersebut, akan dibagi dua sehingga akan menghasilkan tingkat akurasi sebagai berikut:

$$= 0.959011222 = 0.95\% \quad (4)$$

	precision	recall	f1-score	support
negative covid	0.95	1.00	0.97	41471
positive covid	0.87	0.36	0.51	3507
accuracy			0.95	44978
macro avg	0.91	0.68	0.74	44978
weighted avg	0.94	0.95	0.94	44978

Gambar 19. Hasil Tingkat Akurasi Support Vector Machine Kernel RBF Pada Google Colaboratory

Hasil dari perhitungan diatas mendapatkan nilai sebesar 0.959011222 berdasarkan perhitungan dengan menggunakan rumus *Confusion Matrix*. Pada gambar 19 merupakan hasil tingkat akurasi metode *Support Vector Machine Kernel RBF* pada *tools Google Colaboratory*, yang dikonversi dalam bentuk pecahan persen menjadi 0.95%, *dataset* yang digunakan dari *website Israeli Ministry of Health* pada tahun 2022.

Setelah dilakukan pengujian tingkat akurasi dalam memprediksi gejala *COVID-19* dengan menggunakan metode *Logistic Regression* dan *Support Vector Machine*, nilai yang didapatkan dapat dilihat pada tabel dibawah ini.

Tabel 2. Hasil Tingkat Akurasi Prediksi Gejala COVID-19

Metode	Tingkat Akurasi
Neural Network	0.95%
Logistic Regression	0.95%
Support Vector Machine (Kernel Linear)	0.95%
Support Vector Machine (Kernel Sigmoid)	0.91%
Support Vector Machine (Kernel RBF)	0.95%

Berdasarkan tabel 2 bahwa metode *Neural Network* dari penelitian sebelumnya memiliki tingkat akurasi 0.95%, *Logistic Regression* memiliki tingkat akurasi sebesar 0.95%, sementara metode *Support Vector*

Machine Kernel Linear memiliki akurasi sebesar 0.95%, *Support Vector Machine Kernel Sigmoid* 0.91%, dan *Support Vector Machine Kernel RBF* memiliki akurasi sebesar 0.95%. Hasil tingkat akurasi dievaluasi dengan menggunakan *Confusion Matrix* untuk mengukur performa dan tingkat akurasi yang didapatkan. Akurasi yang tertinggi untuk sebuah metode adalah tingkat akurasi yang mendekati angka "1".

Support Vector Machine Kernel Sigmoid memiliki akurasi yang rendah dibandingkan dengan *Support Vector Machine Kernel Linear* dan *Kernel RBF*, maka *Kernel Sigmoid* belum memiliki keakuratan yang maksimal, sehingga tidak direkomendasikan untuk digunakan dalam memprediksi gejala-gejala *COVID-19* berdasarkan dataset yang digunakan dalam penelitian ini.

4. Kesimpulan

Dari hasil dan pembahasan yang telah dijelaskan, bahwa penulis menyimpulkan dari analisis tingkat akurasi prediksi gejala-gejala *COVID-19* menggunakan metode *Logistic Regression* dan *Support Vector Machine* sebagai berikut:

1. Melakukan pengujian ulang dari penelitian sebelumnya dengan menggunakan metode *Logistic Regression* dan dataset pasien berdasarkan gejala-gejala *COVID-19* dengan data sebanyak 136.294 data. Hasil yang didapatkan sama dengan penelitian sebelumnya, dengan tingkat akurasi sebesar 0.95%.
2. Hasil penelitian pada metode *Logistic Regression* mendapatkan tingkat akurasi sebesar 0.95% Artinya, tingkat akurasi yang didapatkan metode *Logistic Regression* memiliki nilai yang sama dengan metode *Neural Network* untuk memprediksi gejala-gejala *COVID-19*.
3. Hasil penelitian pada metode *Support Vector Machine* adalah sebagai berikut:
 - a. *Kernel Linear* memiliki tingkat akurasi sebesar 0.95%.
 - b. *Kernel Sigmoid* memiliki tingkat akurasi sebesar 0.91%
 - c. *Kernel RBF* memiliki tingkat akurasi sebesar 0.95%

Artinya, metode *Support Vector Machine* dalam memprediksi gejala-gejala awal *COVID-19* memiliki akurasi yang baik dengan menggunakan *Kernel Linear* dan *RBF*, karena kedua *kernel* tersebut mendapatkan akurasi yang tinggi dibandingkan *Kernel Sigmoid*. Oleh karena itu, *Kernel Sigmoid* tidak direkomendasikan untuk melakukan prediksi gejala-gejala awal *COVID-19* berdasarkan dari dataset yang digunakan dalam penelitian ini.

Daftar Rujukan

- [1] Ferdiansyah, J. Bahru, M. hajar, th Yoppy Sazaki, and rd Raja Zahilah Raja Md Radzi, "A LSTM-Method for Bitcoin Price Prediction: A Case Study Yahoo Finance Stock Market."
- [2] S. Mubarq and I. M. Insyiroh, "Jurnal Kependudukan Indonesia | Edisi Khusus Demografi dan COVID-19," 2020.
- [3] World Health Organization, "Pertanyaan Dan Jawaban Terkait Coronavirus," 2023. https://www.who.int/indonesia/news/novel_coronavirus/qa/qa-for-public (accessed Aug. 25, 2023).
- [4] R. Rahmi, D. Antoni, H. Syaputra, F. Fatoni, and T. B. Kurniawan, "Metode Klasifikasi Gejala Penyakit Coronavirus Disease 19 (COVID-19) Menggunakan Algoritma Neural Network," *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 12, no. 1, pp. 16–23, Mar. 2023, doi: 10.32736/sisfokom.v12i1.1406.
- [5] A. Bimantara and T. A. Dina, *Klasifikasi Web Berbahaya Menggunakan Metode Logistic Regression*. 2018.
- [6] H. Catur, "PERBANDINGAN METODE SUPPORT VECTOR MACHINE," 2018. Accessed: Aug. 25, 2023. [Online]. Available: <https://www.semanticscholar.org/paper/PERBANDINGAN-METODE-SUPPORT-VECTOR-MACHINE-%28SVM%29-Ningrum/d763dda63a960f3f28978349b70401256288a3b7>
- [7] K. Kelvin, J. Banjarnahor, and E. Indra, "2365-Article Text-7532-1-10-20220208," *Jurnal Sistem Informasi dan Ilmu Komputer Universitas Prima Indonesia*, 2022, Accessed: Aug. 25, 2023. [Online]. Available: <https://doi.org/10.34012/journalsisteminformasidankilmukomputerv5i2.2365>
- [8] D. Mualfah, Ramadholi, R. Gunawan, and M. D. Suratno, "4920-Article Text-16394-1-10-20230720".
- [9] A. Toha, P. Purwono, and W. Gata, "Model Prediksi Kualitas Udara dengan Support Vector Machines dengan Optimasi Hyperparameter GridSearch CV," *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 4, no. 1, pp. 12–21, May 2022, doi: 10.12928/biste.v4i1.6079.
- [10] M. Novela and T. Basaruddin, "DATASET SUARA DAN TEKS BERBAHASA INDONESIA PADA REKAMAN PODCAST DAN TALK SHOW," *Agustus*, vol. 11, no. 2, pp. 61–66.
- [11] D. Normawati and S. A. Prayogi, "Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter," 2021.