

# Prediksi Tingkat Risiko Penyebaran COVID-19 pada Pulau Sumatera menggunakan Metode KNN dan SVM

Fadhilah Nur Amaliah<sup>#1</sup>, Ghifary Catur Achmad Dhany<sup>#2</sup>, Mohammad Luthfan Faohan<sup>#3</sup>, Naila Ameera Larasati<sup>\*4</sup>, Nova Nurul Putri<sup>#5</sup>

<sup>#</sup>Universitas Pendidikan Indonesia, Universitas Gunadarma

[mluthfanfaohan@upi.edu](mailto:mluthfanfaohan@upi.edu), [novanurul@upi.edu](mailto:novanurul@upi.edu), [nailaameera@student.gunadarma.ac.id](mailto:nailaameera@student.gunadarma.ac.id),  
[ghifarycaturachmaddhany@student.gunadarma.ac.id](mailto:ghifarycaturachmaddhany@student.gunadarma.ac.id), [fadhilahamaliah@student.gunadarma.ac.id](mailto:fadhilahamaliah@student.gunadarma.ac.id).

Jl. Veteran No. 08, Purwakarta, Jawa Barat Indonesia, Jl. Margonda Raya No.100, Pondok Cina, Beji, Kota Depok, Jawa Barat 16424

kampus\_purwakarta@upi.edu

sektor@gunadarma.ac.id

**Abstrak**— COVID-19 merupakan virus yang menyebabkan infeksi saluran pernafasan. Penularan COVID-19 yang sangat cepat ini memiliki dampak yang luas bagi seluruh masyarakat dari negara di seluruh dunia. Bidang Data science sangat berperan dalam membantu pemantauan COVID-19 di Indonesia. Pada saat ini, forecasting yang merupakan salah satu penerapan data science berperan penting untuk memprediksi tingkat risiko penyebaran COVID-19. Sebagai salah satu negara yang terkena dampak bahayanya COVID-19, Indonesia melakukan penutupan diseluruh wilayah yang memiliki jumlah kasus positif yang cukup besar, salah satunya adalah wilayah Pulau Sumatera. Dalam hal ini, dibutuhkannya informasi prediksi yang akurat mengenai penyebaran COVID-19 di wilayah Sumatera. Penelitian ini bertujuan untuk melihat efektivitas penerapan model KNN dan SVM dalam memprediksi tingkat penyebaran COVID-19 di Pulau Sumatera. Pengukuran pengujian kedua model tersebut dapat dilihat melalui confusion matrix. Dimana nilai evaluasi matrix tertinggi yang berupa akurasi, precision, recall dan f1-score masing-masing menghasilkan 87%, 89%, 87% dan 86% didapat dari penggunaan algoritma KNN dengan n neighbor 5.

**Kata Kunci**— COVID-19, KNN, SVM, Kernel, Pulau Sumatera, Prediksi.

## I. PENDAHULUAN

Pada Desember tahun 2019, SARS-CoV-2 muncul pertama kali di wilayah Wuhan, Tiongkok. SARS-CoV-2 atau yang biasa dikenal dengan COVID-19 merupakan virus yang menyebabkan infeksi saluran pernafasan.

Virus ini memiliki macam gejala, mulai dari gejala yang mirip dengan flu biasa hingga penyakit yang lebih parah seperti Middle East Respiratory Syndrome (MERS). COVID-19 dapat menyebar melalui percikan-percikan cairan tubuh yang keluar saat seseorang yang terinfeksi batuk, bersin atau berbicara.

Penularan COVID-19 yang sangat cepat ini memiliki dampak yang luas bagi seluruh masyarakat dari negara di seluruh dunia. Di Indonesia, kasus COVID-19 muncul pertama kali pada awal Maret 2020. Sejak saat itu penyebaran COVID-19 meluas, hingga jumlah angka positif mencapai 4,7 juta jiwa dengan korban meninggal mencapai 132 ribu (Kemenkes RI, 2021)[1].

Saat ini banyak peneliti medis di seluruh dunia telah menemukan obat dan vaksin yang tepat untuk virus COVID-19. Pemerintah Indonesia juga cukup baik dalam menangani pencegahan virus ini, dengan berbagai cara mulai dari diberlakukannya protokol kesehatan, tes deteksi COVID-19 menggunakan PCR dan pemberian vaksin gratis. Bidang Data science juga sangat berperan dalam membantu pemantauan COVID-19 di Indonesia. Data Science merupakan salah satu bidang studi yang mampu membuat prediksi atau *forecasting*. Metode ini digunakan untuk memprediksi masa depan berdasarkan tren masa lalu dan masa sekarang[2]. Beberapa contoh *forecasting* diantaranya adalah peramalan cuaca, prediksi penyakit dan pasar saham. Pada saat ini, *forecasting* berperan penting untuk memprediksi tingkat risiko penyebaran COVID-19.

Sebagai salah satu negara yang terkena dampak bahayanya COVID-19, Indonesia melakukan penutupan diseluruh wilayah yang memiliki jumlah kasus positif yang cukup besar, salah satunya adalah wilayah Sumatera. Menurut data 30 April 2022 Pulau Sumatera, terdapat 811.180 orang terkonfirmasi positif, 24.621 orang meninggal, 784.869 orang sembuh total [2]. Dalam hal ini, dibutuhkannya informasi prediksi yang akurat mengenai penyebaran COVID-19 di wilayah Sumatera. Oleh karena itu, penulis melakukan prediksi tingkat penyebaran COVID-19 di wilayah Sumatera.

Dalam penelitian ini, penulis menggunakan metode algoritma KNN dan SVM untuk memprediksi tingkat risiko COVID-19 pada Pulau Sumatera. *K-Nearest Neighbors* (KNN) digunakan untuk mengklasifikasi sekumpulan data berdasarkan pembelajaran data. Salah satu ciri khusus algoritma ini adalah untuk mencari kasus dengan menghitung kedekatan kasus yang baru dengan kasus yang lama. Sedangkan *Support Vector Machine* (SVM) adalah algoritma klasifikasi yang sering digunakan dalam klasifikasi teks dan dapat menunjukkan penampilan yang baik dibanding algoritma lainnya. Dua Metode ini biasanya digunakan untuk pemecahan masalah pada bidang *Data Science*. Tujuan penelitian ini untuk melihat seberapa efektif penerapan metode KNN dan SVM untuk prediksi sebuah data penyebaran

COVID-19 pada Pulau Sumatera. Oleh karena itu, peneliti melakukan komparasi hasil evaluasi dari algoritma KNN dan SVM pada data penyebaran COVID-19 Pulau Sumatera.

## II. STUDI LITERATUR

### 2.1 *K-Nearest Neighbors*

*K-Nearest Neighbors* atau K-NN melakukan klasifikasi dengan proyeksi data pembelajaran pada ruang berdimensi banyak. Ruang ini dibagi menjadi bagian-bagian yang merepresentasikan kriteria data pembelajaran. Setiap data pembelajaran direpresentasikan menjadi titik-titik  $c$  pada ruang dimensi banyak. K-NN adalah algoritma yang menghitung kemiripan pada tiap data uji dan semua data awal untuk menghitung daftar nearest neighbor-nya [3]. Konsep dasar dari K-NN adalah mencari jarak terdekat antara data yang akan dievaluasi dengan  $K$  tetangga terdekatnya dalam data pelatihan [4]. Teknik pencarian tetangga terdekat yang umum dilakukan dengan menggunakan formula jarak euclidean. Berikut beberapa formula yang digunakan dalam algoritma KNN.

- *Euclidean Distance*

Jarak *Euclidean* adalah formula untuk mencari jarak antar titik dalam ruang dimensi [5].

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

- *Hamming Distance*

Jarak *Hamming* adalah cara mencari jarak antara 2 titik yang dihitung dengan panjang vektor biner yang dibentuk oleh dua titik tersebut dalam

$$d_{ij} = q + r$$

block kode biner.

- *Manhattan Distance*

*Manhattan Distance* atau Taxicab Geometri adalah formula untuk mencari jarak  $d$  antar 2 vektor  $p, q$  pada ruang dimensi  $n$

$$d(x, y) = \sum_{i=1}^n |x_i - y_i|$$

- *Minkowski Distance*

*Minkowski distance* adalah formula pengukuran antara 2 titik pada ruang vektor normal yang merupakan hibridisasi yang menggeneralisasi *euclidean distance* dan *manhattan distance*.

$$d(x, y) = \left( \sum_{i=1}^n |x_i - y_i|^p \right)^{1/p}$$

Tujuan utama dari algoritma ini yaitu mengklasifikasikan suatu obyek berdasarkan atribut-atribut dan training sample. Algoritma K-Nearest Neighbor (K-NN) menggunakan klasifikasi kedekatan titik sebagai nilai perkiraan dari query instance yang baru. Jarak yang digunakan adalah jarak *Euclidean Distance* [6].

## 2.2 Support Vector Machine (SVM)

*Support Vector Machine* (SVM) dikenalkan pertama kali oleh Vapnik tahun 1992 [7]. SVM adalah algoritma *machine learning* dengan pendekatan *supervised learning* yang dapat digunakan untuk klasifikasi (SVM *Classification*) dan regresi (SVM *Regression*). SVM dapat mengatasi masalah klasifikasi dan regresi dengan *linear* maupun *non-linear*. Cara kerja dari metode SVM khususnya pada masalah *non-linear* adalah dengan memasukkan konsep kernel ke dalam ruang berdimensi tinggi. Tujuannya adalah untuk mencari *hyperplane* atau pemisah yang dapat memaksimalkan jarak (*margin*) antar data kelas [8].

Dalam SVM terdapat beberapa jenis kernel yang sering digunakan, diantaranya adalah:

1. Kernel *Linear*, umum digunakan untuk dataset linear.

2. Kernel *Polynomial*, umum digunakan untuk data normal.

$$K(\vec{X}_i, \vec{X}_j) = (\vec{X}_i, \vec{X}_j + 1)^q$$

3. Kernel *Radial Basis Function* (RBF) atau *Gaussian*, umum digunakan untuk data *non-linear*.

$$K(\vec{X}_i, \vec{X}_j) = \exp\left(\frac{\|\vec{X} - \vec{X}_i\|^2}{2\sigma^2}\right)$$

4. Kernel *Sigmoid*, merupakan pengembangan dari jaringan syaraf tiruan.

## 2.3 Confusion Matrix

*Confusion Matrix* adalah tabel evaluasi informasi yang berisi perbandingan hasil prediksi klasifikasi dari data yang akan dievaluasi. Evaluasi ini dilakukan oleh sistem dengan nilai yang sebenarnya. Contoh *Confusion Matrix* dapat dilihat pada **Tabel 1**.

**Tabel 1**

	Actual Values	
	1 (Positive)	0 (Negative)

<b>Predicted Values</b>	TP (True Positive)	FP (False Positive) <b>Type I Error</b>
	FN (False Negative) <b>Type II Error</b>	TN (True Negative)

*Confusion Matrix* atau *Error Matrix*, membandingkan hasil klasifikasi yang dilakukan oleh sistem (model) dengan hasil klasifikasi sebenarnya [9]. Untuk mengukur *performance metrics* dari *confusion matrix*, dapat digunakan *Accuracy*, *Precision* dan *Recall*.

- *Accuracy* (Q): Tingkat kedekatan nilai prediksi dengan nilai aktual (sebenarnya). Dari nilai akurasi, kita dapat mengetahui berapa persen kota yang benar diprediksi masuk kategori tingkat resiko rendah maupun tinggi.

$$\text{Rumus: } Accuracy = \frac{(TP+TN)}{(TP+FP+FN+TN)}$$

- *Precision* (PR): rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif. Dari nilai Precision, kita dapat mengetahui berapa persen

kota/kabupaten yang benar masuk dalam tingkat penyebaran dengan resiko tinggi dari seluruh kota/kabupaten yang diprediksi masuk ke kategori resiko tinggi.

$$\text{Rumus: } Precision = \frac{(TP)}{(TP + FP)}$$

- *Recall* atau *Sensitivity* (SE): rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif. Dengan nilai *Recall*, kita dapat mengetahui berapa persen kota/kabupaten yang diprediksi masuk tingkat resiko tinggi dibandingkan keseluruhan kota/kabupaten yang sebenarnya masuk dalam risiko tinggi.

$$\text{Rumus: } Recall = \frac{TP}{(TP + FN)}$$

- *F1-Score* adalah parameter yang digunakan untuk menghitung matrik kinerja dari kombinasi *Precision* dan *Recall*.

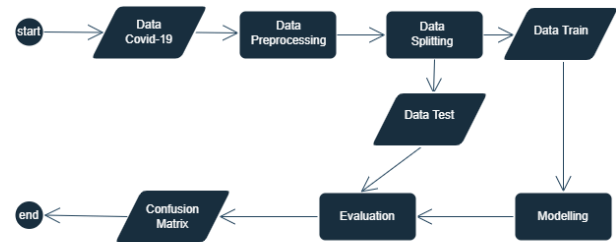
$$\text{Rumus: } F-1 \text{ Score} = \frac{(2 * Recall * Precision)}{(Recall + Precision)}$$

### III. METODE PENELITIAN

Penelitian ini adalah penelitian eksperimen yang bertujuan untuk mencari *accuracy* dan prediksi kasus COVID-19. Data yang digunakan merupakan *time series* atau deret waktu yang disajikan per hari secara berurutan. Pada **Gambar 1** menjelaskan mengenai metode penelitian yang diawali dengan pengumpulan data (data uji). Penelitian ini menggunakan algoritma KNN dan SVM untuk prediksi kasus COVID-19 di Pulau Sumatera. Untuk melakukan klasifikasi dibutuhkan data uji sebagai data pembelajaran lalu dilakukan pengujian model dengan menggunakan *testing data* (data latih). Hasil validasi pada kedua metode tersebut menggunakan *confusion matrix* dan bertujuan untuk mengetahui tingkat akurasi keseluruhan dari sistem yang dirancang. Tahapan Penelitian ini dilakukan secara bertahap dan sistematis, adapun tahapan penelitian ini akan dijabarkan pada berikut:

#### 3.1 Alur Penelitian

Tahapan Penelitian ini dilakukan secara bertahap dan sistematis, adapun tahapan penelitian ini akan dijabarkan pada berikut :



Gambar 1 Sistemika Penelitian

#### 3.1.1 Data Acquisition

Dalam penelitian ini data yang digunakan didapat melalui situs informasi dan koordinasi COVID-19 milik web <https://www.andrafarm.com/> [10]. Data yang digunakan merupakan data dengan format CSV dan dibuat manual.

No.	Negara, Provinsi, Kota, Kabupaten, Kecamatan, Kelurahan, Desa (update 13 Juni, jam 22:41 WIB)	Kasus Positif (Konfirmasi)	Meninggal	Sembuh	Masih Sakit, Positif Aktif, Dirawat
1	Sumatera Pulau	811.667	24.661	786.783	223
2	Sumatera Barat (Sumbar) Provinsi	103.840	2.350	101.460	30
3	Sumatera Selatan (Sumsel) Provinsi	80.501	3.343	77.133	25
4	Sumatera Utara (Sumut) Provinsi	155.104	3.257	151.793	54

Gambar 2 Data Keseluruhan Pulau Sumatera

Berdasarkan data yang diambil dari *website* resmi dapat dilihat pada gambar di atas yang merupakan hasil dari kasus serta konfirmasi berupa positif, meninggal, sembuh dan juga sakit. Selain itu, pada penelitian ini juga terdapat penambahan variabel dengan tingkat risiko seperti yang tertera pada gambar 3 di bawah ini.

	kota_kabupaten	kasus_positif_total	total_meninggal	masih_sakit	total_sembuh	tingkat_resiko
0	Kota Padang	25.421	424.0	1.0	24.996	Resiko Tinggi
1	Kab. Agam	4.475	115.0	104.0	4.256	Resiko Tinggi
2	Kab. Tanah Datar	2.818	87.0	64.0	2.667	Resiko Rendah
3	Kota Bukittinggi	2.566	52.0	26.0	2.488	Resiko Rendah
4	Kab. Padang Pariaman	2.148	93.0	78.0	1.977	Resiko Tinggi
5	Kab. Pesisir Selatan	2.127	68.0	53.0	2.006	Resiko Rendah
6	Kab. Solok	2.052	70.0	52.0	1.930	Resiko Rendah
7	Kab. Limapuluh Kota	1.768	49.0	33.0	1.686	Resiko Rendah
8	Kota Payakumbuh	1.672	28.0	13.0	1.631	Resiko Rendah
9	Kota Padang Panjang	1.666	31.0	11.0	1.624	Resiko Rendah

**Gambar 3 Dataset Sumatera.csv**

### 3.1.2 Data Exploration

- Data Exploration merupakan pendekatan yang mirip dengan analisis data awal, di mana seorang analis data menggunakan eksplorasi visual untuk memahami apa yang ada dalam dataset dan karakteristik data, daripada melalui sistem manajemen data tradisional [11].
- Data Selection merupakan proses pemilihan data dari sekumpulan data operasional yang ada sebelum masuk ke tahap mining data maupun informasi. Pada tahap ini akan dilakukan langkah-langkah sebagai berikut.

#### A. Pivot Data

Data yang sudah dipivot sebanyak 150 data berdasarkan kota dan kabupaten serta mengakumulasikan angka kasus dari masing-masing kabupaten dan kota terhadap penyebaran COVID-19 di Sumatera ini nantinya yang akan digunakan dalam proses data mining menggunakan algoritma *K-Nearest Neighbor*.

#### B. Seleksi Atribut

Penggunaan atribut atau label pada prediksi ini hanya menggunakan 5 atribut yaitu atribut total kasus suspek, total kasus meninggal, masih sakit dan total sembuh sebagai atribut yang dijadikan indikator prediksi, dan satu atribut identitas yaitu nama kabupaten/kota. Sedangkan untuk atribut lain yang mengandung angka-angka akumulasi penyebaran kasus COVID-19 ditiadakan karena tidak bersinggungan langsung dengan proses selanjutnya.

### 3.1.3 Modeling

Tahap selanjutnya adalah membuat model dengan algoritma KNN (*K-Nearest Neighbor*). KNN merupakan salah satu metode yang menerapkan algoritma *supervised* dimana hasil dari sampel uji yang baru diklasifikasikan berdasarkan mayoritas dari kategori pada KNN. Kemudian dilanjutkan dengan membuat model SVM. SVM merupakan algoritma yang bekerja menggunakan pemetaan nonlinear untuk mengubah data pelatihan asli ke dimensi yang lebih tinggi.

### 3.1.4 Evaluation

Pada Tahapan evaluasi ini KNN dan SVM digunakan untuk membentuk suatu prediksi kelompok suatu masalah dengan akurasi yang tepat.

### 3.2 Dataset

Dataset yang digunakan dalam penelitian ini berupa data COVID-19 dari Pulau Sumatera. Dataset berjumlah 150 baris data dan 6 kolom yang masing-masing terdiri dari kolom kota dan kabupaten, dimana data berisikan daftar nama kota dan kabupaten yang berada di Pulau Sumatera. Diikuti dengan kolom kasus positif total, total meninggal dan total sembuh yang ketiganya terhitung mulai dari awal Maret 2020 hingga 30 April 2022. Kolom masih sakit, dimana data berisikan jumlah pasien yang masih menderita COVID-19 hingga saat ini. Kolom terakhir merupakan tingkat resiko yang merupakan kolom yang mengkategorikan tinggi atau rendahnya setiap kota dan kabupaten di Pulau Sumatera. Dataset COVID-19 di Pulau Sumatera dapat dilihat pada **Tabel 2**.

Keterangan kolom:

*M\_total*: Total Meninggal

**Tabel 2 Dataset COVID-19 Pulau Sumatera**

kota/kab	Post_total	M_total	sakitt	sembuh	Resiko
Kota Padang	25.421	424	1	24.996	Resiko Tinggi
Kab. Agam	4.475	115	104	4.256	Resiko Tinggi
Kab. Tanah Datar	2.818	87	64	2.667	Resiko Rendah
Kota Bukittinggi	2.566	52	26	2.488	Resiko Rendah
Kab. Padang Pariaman	2.148	93	78	1.977	Resiko Tinggi
Kab. Pesisir Selatan	2.127	68	53	2.006	Resiko Rendah
Kab. Solok	2.052	70	52	1.930	Resiko Rendah
Kab. Limapuluh Kota	1.768	49	33	1.686	Resiko Rendah
Kota Payakumbuh	1.672	28	13	1.631	Resiko Rendah
Kota Padang Panjang	1.666	31	11	1.624	Resiko Rendah
Kab. Sijunjung	1.509	43	10	1.456	Resiko Rendah
Kab. Dharmasraya	1.373	26	14	1.333	Resiko Rendah
Kota Solok	1.371	29	17	1.325	Resiko Rendah
Kab. Pasaman Barat	1.247	76	65	1.106	Resiko Tinggi
Kab. Solok Selatan	1.081	18	8	1.055	Resiko Rendah
Kota Pariaman	1.043	20	12	1.011	Resiko Rendah
Kota Sawahlunto	890	18	9	863	Resiko Rendah
Kab. Kepulauan Mentawai	834	1	0	833	Resiko Rendah
Kab. Pasaman	662	37	31	594	Resiko Tinggi



Dari *Tabel 2*, dapat dilihat bahwa Kota Padang menghasilkan angka 25.421 untuk kasus positif total, 24.996 total sembuh dan 424 total meninggal. Hingga pada Kabupaten Pasaman menghasilkan 662 untuk kasus positif total, 594 total sembuh, dan 31 total meninggal.

Dataset yang digunakan berjumlah 150 data dengan 44 berlabel resiko tinggi atau 1 dan 106 data berlabel resiko rendah atau 0. Selanjutnya, data dibagi menjadi 2 bagian yaitu data latih (*training*) dan data uji (*testing*) dengan perbandingan komposisi pembagian data yaitu 80% untuk data latih dan 20% untuk data uji. Data latih terdiri dari 120 *rows* sedangkan untuk data uji terdiri dari 30 *rows*.

### 3.3 Pembangunan Model dan Validasi

Dalam penelitian ini penulis membangun model menggunakan metode KNN dan SVM. Pada tahapan Pembangunan ini, klasifikasi KNN dan SVM menggunakan data latih serta data uji dengan fitur yang telah terpilih. Setelah dilakukan pembangunan kedua model tersebut, selanjutnya adalah melakukan validasi model untuk mengetahui performansi model, meliputi Akurasi, *Recall*, *Precision* dan *F1-Score*. Terdapat 8 model yang dibangun dengan jumlah *n-neighbor* serta kernel yang berbeda. Model yang dibangun pada penelitian ini dapat dituliskan pada *Tabel 3* dan *Tabel 4*

**Tabel 3 Model Prediksi KNN berdasarkan jumlah *n-neighbor***

Algoritma	JUMLAH K
KNN	K=3
KNN	K=4
KNN	K=5
KNN	K=6

**Tabel 4 Model Prediksi SVM berdasarkan Jenis Kernel**

Algoritma	Kernel
SVM	Linear
SVM	Sigmoid
SVM	Polynomial
SVM	RBF

Tahap selanjutnya adalah membandingkan hasil yang diperoleh dari implementasi dengan kriteria dan standar yang telah ada untuk mendapatkan angka keberhasilan dari implementasi. Pada tahap selanjutnya dilakukan validasi model yang telah dibangun menggunakan metode *confusion matrix*. *Confusion matrix* adalah tabel yang digunakan untuk mendapatkan nilai kinerja *classifier* melalui nilai parameter matrix performansi.

Dalam penelitian ini, *confusion matrix* digunakan membandingkan hasil prediksi yang dilakukan oleh sistem (model) dengan kriteria dan standar yang telah ada. Adapun performansi yang digunakan adalah Akurasi, *Precision*, *Recall* dan *F1-Score*. Bentuk *confusion matrix* yang menggambarkan kelas prediksi dan kelas aktual dapat dilihat pada Tabel 4.

## IV. HASIL & PEMBAHASAN

### 4.1 Pembahasan Algoritma KNN

Setelah melakukan proses *split* dan normalisasi data, langkah selanjutnya adalah melakukan pengujian dari hasil pemodelan menggunakan algoritma K-NN dan SVM. Penulis akan melakukan pengujian dengan jumlah *n* neighbors yang berbeda untuk algoritma K-NN dan pengujian terhadap 4 jenis kernel dari algoritma SVM. Maka dari itu hasil dan pembahasan dalam penelitian ini terbagi menjadi ke beberapa skema, yaitu:

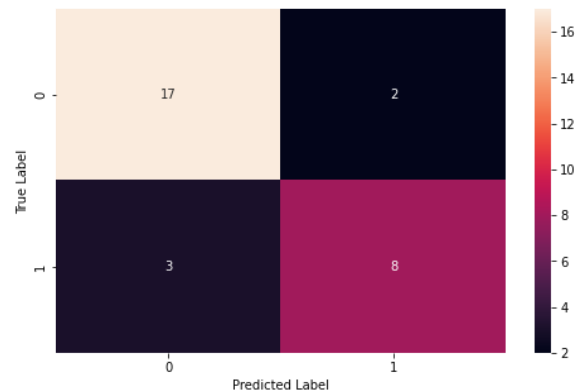
#### 4.1.1 Hasil Pengujian Model KNN dengan *n* neighbor 3

Pada skenario pertama digunakan algoritma K-NN dengan jumlah *n* neighbor 3. Dari pemodelan ini, melalui performansi matrix didapatkan hasil akurasi sebesar 0.833 atau 83%. Hal ini menunjukkan prediksi yang telah

dilakukan memiliki tingkat keakuratan yang baik. Kemudian, diikuti dengan nilai *Precision* sebesar 83%, nilai *recall* sebesar 83% dan *f1\_score* sebesar 83%. Hasil dari *confusion matrix* model KNN dengan jumlah *n* neighbor 3 dapat dilihat di **Gambar 4**.

	precision	recall	f1-score	support
0	0.85	0.89	0.87	19
1	0.80	0.73	0.76	11
accuracy			0.83	30
macro avg	0.82	0.81	0.82	30
weighted avg	0.83	0.83	0.83	30

**Gambar 4** Nilai Performansi Matriks



**Gambar 5** Visualisasi confusion matrix

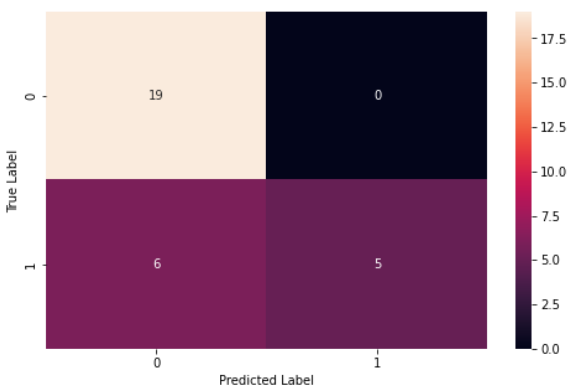
Berdasarkan visualisasi confusion matrix dari **Gambar 5**, dari 20 kota tingkat resiko rendah atau 0, model memprediksi terdapat 3 kota yang diprediksi tingkat resiko tinggi (FN). Sedangkan dari 10 kota tingkat resiko tinggi atau 1, model memprediksi terdapat 2 kota yang tingkat resiko rendah.

#### 4.1.2 Hasil Pengujian Model KNN dengan n neighbor 4

Pada skenario kedua digunakan algoritma KNN dengan jumlah n Neighbor 4. Tahap selanjutnya adalah melakukan validasi model performansi matrix dan didapatkan hasil akurasi sebesar 0.8 atau 80%. Kemudian, diikuti dengan nilai *Precision* sebesar 85%, nilai *recall* sebesar 80% dan *f1-score* sebesar 78%. Hasil dari confusion matrix model KNN dengan jumlah n neighbor 4 dapat dilihat pada **Gambar 6**.

	precision	recall	f1-score	support
0	0.76	1.00	0.86	19
1	1.00	0.45	0.62	11
accuracy			0.80	30
macro avg	0.88	0.73	0.74	30
weighted avg	0.85	0.80	0.78	30

**Gambar 6 Nilai Performansi Matriks**



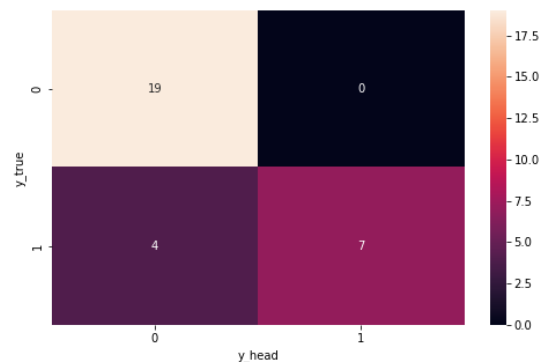
**Gambar 7 Visualisasi Confusion Matrix**

Berdasarkan visualisasi confusion matrix dari **Gambar 7**, dari 25 kota tingkat resiko rendah, model memprediksi terdapat 6 kota yang diprediksi masuk ke tingkat resiko tinggi (FN).

Sedangkan dari 5 kota tingkat resiko tinggi, model memprediksi terdapat 0 kota yang tingkat resiko rendah dimana prediksi tersebut sesuai dengan *true value*-nya yaitu 5.

#### 4.1.3 Hasil Pengujian Model KNN dengan n neighbor 5

Hasil pengujian model KNN dengan n Neighbor berjumlah 5 dapat dilihat di **Gambar 8**. Dengan ini dari 25 kota tingkat resiko rendah dan tinggi diprediksi ada 4 kota yang tinggi (FN), sedangkan dari 7 Kota tingkat resiko tinggi, model prediksi terpadat 0 kota serta prediksinya benar.



**Gambar 8 Visualisasi Confusion Matrix**

	precision	recall	f1-score	support
0	0.83	1.00	0.90	19
1	1.00	0.64	0.78	11
accuracy			0.87	30
macro avg	0.91	0.82	0.84	30
weighted avg	0.89	0.87	0.86	30

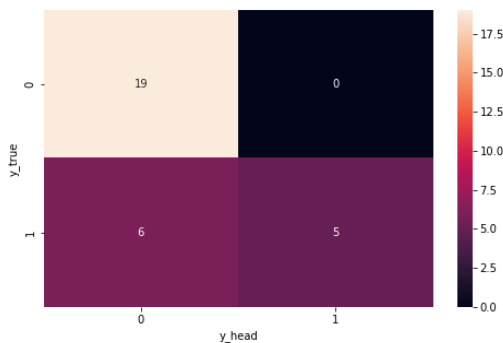
**Gambar 9 Nilai Performansi Matriks**

Berdasarkan hasil dari **Gambar 9** menunjukkan bahwa nilai akurasi mencapai 87%. Diikuti

dengan nilai *precision* yang cukup tinggi yaitu 89%, nilai *recall* 87% dan *f1-score* bernilai 86%.

#### 4.1.4 Hasil Pengujian Model KNN dengan n neighbor 6

Pada skenario keempat dengan menggunakan algoritma K-NN dengan jumlah n Neighbor 6. Jika dilihat hasil dari confusion matrix pada **Gambar 10**, dari 25 kota dengan tingkat resiko rendah atau 0, model algoritma memprediksi 6 kota yang masuk ke kategori tingkat resiko tinggi (FN). Sedangkan dari 5 kota dengan tingkat resiko tinggi, model algoritma memprediksi terdapat 0 kota yang tingkat resiko rendah atau prediksinya benar sempurna.



**Gambar 10 Visualisasi Confusion Matrix**

	precision	recall	f1-score	support
0	0.76	1.00	0.86	19
1	1.00	0.45	0.62	11
accuracy			0.80	30
macro avg	0.88	0.73	0.74	30
weighted avg	0.85	0.80	0.78	30

**Gambar 11 Nilai Performansi Matriks**

Berdasarkan nilai ketepatan prediksi **Gambar 11** maka dapat diketahui bahwa akurasi menghasilkan nilai 80%, diikuti dengan nilai *precision* 85%, *recall* bernilai 80% dan *f1-score* sebesar 78%.

## 4.2 Pembahasan Algoritma SVM

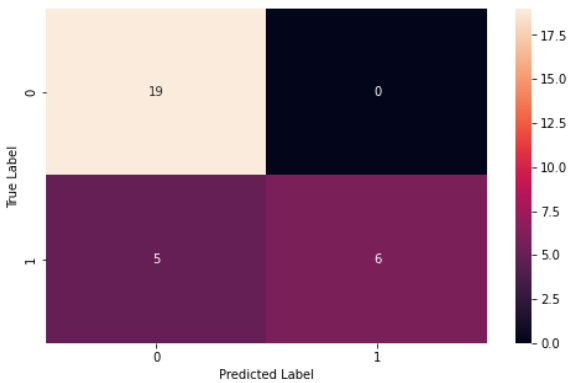
### 4.2.1 Hasil Pengujian Model SVM dengan Jenis Kernel Linear

Skema selanjutnya merupakan pengujian model prediksi menggunakan algoritma SVM. Pada tahap ini jenis kernel yang digunakan merupakan kernel linear. Melalui performa matrik didapatkan hasil akurasi sebesar 0.83 atau 83%. Kemudian, diikuti dengan nilai *Precision* sebesar 87%, nilai *recall* sebesar 83% dan *f1\_score* sebesar 82%. Hasil dari confusion matrix model SVM dengan jenis kernel linear dapat dilihat di **Gambar 12**.

```
print(classification_report(Y_test, y_pred_svm))
```

	precision	recall	f1-score	support
0	0.79	1.00	0.88	19
1	1.00	0.55	0.71	11
accuracy			0.83	30
macro avg	0.90	0.77	0.79	30
weighted avg	0.87	0.83	0.82	30

**Gambar 12 Nilai Performansi Matriks**



**Gambar 13 Visualisasi Confusion Matrix**

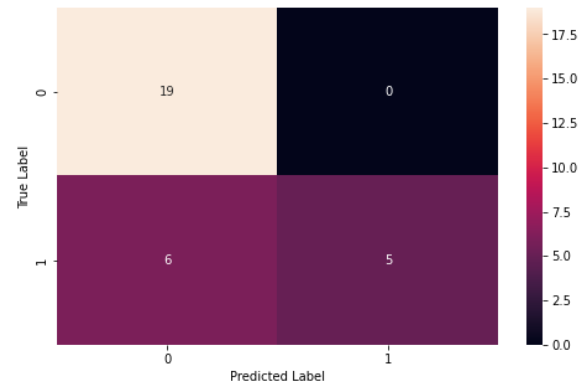
Berdasarkan **Gambar 13** Confusion matriks di atas, menunjukkan bahwa dari kota 24 kota dengan tingkat resiko rendah, model svm memprediksi terdapat 5 kota yang diprediksi masuk ke tingkat resiko tinggi (FN). Sedangkan dari 6 kota dengan tingkat resiko tinggi, model memprediksi dengan sempurna bahwa ke 6 kota tersebut termasuk ke tingkat resiko tinggi.

#### 4.2.2 Hasil Pengujian Model SVM dengan Jenis Kernel Sigmoid

Pada tahap ini jenis kernel yang digunakan merupakan kernel Sigmoid. Melalui performa matrik didapatkan hasil akurasi sebesar 0.80 atau 80%. Kemudian, diikuti dengan nilai *Precision* sebesar 85%, nilai *recall* sebesar 80% dan *f1\_score* sebesar 78%. Hasil dari confusion matrix model SVM dengan jenis kernel Sigmoid dapat dilihat di **Gambar 14**

	precision	recall	f1-score	support
0	0.76	1.00	0.86	19
1	1.00	0.45	0.62	11
accuracy			0.80	30
macro avg	0.88	0.73	0.74	30
weighted avg	0.85	0.80	0.78	30

**Gambar 14 Nilai Performansi Matriks**



**Gambar 15 Visualisasi Confusion Matrix**

Berdasarkan **Gambar 15** confusion matriks di atas, menunjukkan bahwa dari 25 kota dengan tingkat resiko rendah, model svm memprediksi terdapat 6 kota yang diprediksi masuk ke tingkat resiko tinggi (FN). Sedangkan dari 5 kota dengan tingkat resiko tinggi, model memprediksi dengan sempurna bahwa ke 5 kota tersebut termasuk ke tingkat resiko tinggi.

#### 4.2.3 Hasil Pengujian Model SVM dengan Jenis Kernel Polynomial

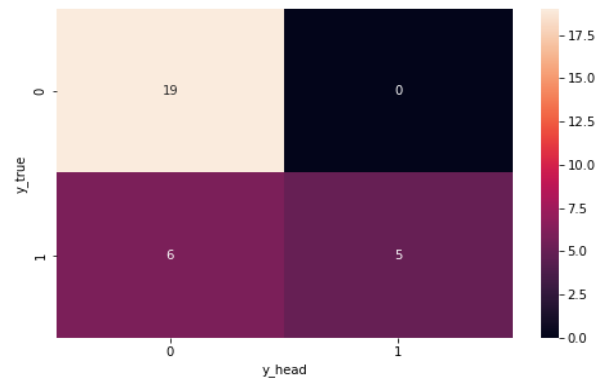
Kernel Polynomial memiliki fungsi kernel digunakan dengan data yang terpisah secara linear. Hasil *confusion matrix* dalam tahap ini menunjukkan bahwa akurasi bernilai sebesar 0.77 atau 77%. Kemudian, diikuti dengan nilai

*Precision* sebesar 83%, nilai *recall* sebesar 77% dan *f1\_score* sebesar 73%. Hasil dari *confusion matrix* model SVM dengan jenis kernel Sigmoid dapat dilihat di **Gambar 16**

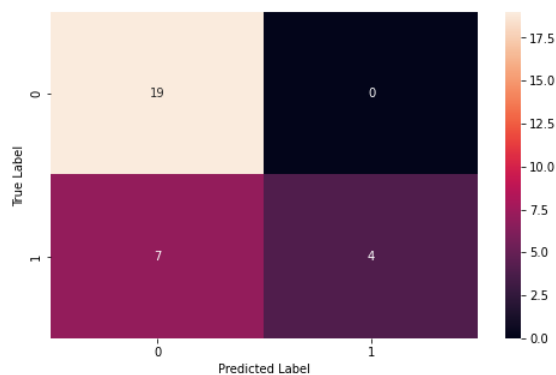
	precision	recall	f1-score	support
0	0.73	1.00	0.84	19
1	1.00	0.36	0.53	11
accuracy			0.77	30
macro avg	0.87	0.68	0.69	30
weighted avg	0.83	0.77	0.73	30

**Gambar 16 Nilai Performansi Matriks**

tinggi (FN). Sedangkan dari 5 kota tingkat resiko tinggi, model algoritma ini memprediksi dengan sempurna yang dimana prediksinya sesuai dengan nilai aslinya.



**Gambar 18 Visualisasi Confusion Matrix**



**Gambar 17 Visualisasi Confusion Matrix**

#### 4.2.4 Hasil Pengujian Model SVM dengan Jenis Kernel RBF

Untuk pemodelan algoritma SVM dengan jenis kernel RBF, jika dilihat hasil dari *confusion matrix* pada **Gambar 18** dari 25 kota dengan tingkat resiko rendah, model algoritma ini menghasilkan bahwa terdapat 6 kota yang diprediksi masuk ke kategori tingkat risiko

	precision	recall	f1-score	support
0	0.76	1.00	0.86	19
1	1.00	0.45	0.62	11
accuracy			0.80	30
macro avg	0.88	0.73	0.74	30
weighted avg	0.85	0.80	0.78	30

**Gambar 19 Nilai Performansi Matriks**

Skema pengujian model prediksi menggunakan algoritma SVM jenis RBF. Melalui performa matrik didapatkan hasil akurasi sebesar 0.80 atau 80%. Kemudian, diikuti dengan nilai *Precision* sebesar 85%, nilai *recall* sebesar 80% dan *f1\_score* sebesar 78%.

Algo	Parameter	Nilai	Akurasi	Press	Rec	F1

SVM	Kernel	Linear	83%	87%	83%	82%
		Sig	80%	85%	80%	78%
		Poly	77%	83%	77%	73%
		RBF	80%	85%	80%	78%
KNN	n	3	83%	83%	83%	83%
		4	80%	85%	80%	78%
		5	87%	89%	87%	86%
		6	80%	85%	80%	78%

## V. Kesimpulan

Berdasarkan hasil penelitian yang dilakukan, kesimpulan yang didapat yaitu sebagai berikut:

- Hasil dari pengujian model prediksi memperlihatkan bahwa tingkat akurasi yang dimiliki oleh KNN model dengan jumlah n neighbor 5 berada pada ambang akurasi yang terbaik yaitu sekitar 87%.
- Hasil dari pengujian model prediksi dengan menggunakan SVM model, menunjukkan bahwa pengujian dengan jenis kernel linear memiliki hasil akurasi terbaik yaitu sekitar 83%.

- Dari kedua model yang diuji, dapat disimpulkan bahwa penggunaan model KNN lebih efektif dalam memprediksi tingkat penyebaran COVID-19 di Pulau Sumatera.

## Daftar Pustaka

- [1] A. Aditia, "e-ISSN 2715-6885; p-ISSN 2714-9757 <http://jurnal.globalhealthsciencegroup.com/index.php/JPPP>," *Jurnal Penelitian Perawat Profesia*, vol. 3, no. 4, hlm. 8, 2021.
- [2] algoritma. (2022). "DATA MINING: PROSES PENTING DALAM DATA SCIENCE"[Online]. Tersedia: <https://algoritma.blog/data-science/data-mining/>
- [3] T. S. Sabrila, V. R. Sari, dan A. E. Minarno, "Analisis Sentimen Pada Tweet Tentang Penanganan Covid-19 Menggunakan Word Embedding Pada Algoritma Support Vector Machine Dan K-Nearest Neighbor," *FIJ*, vol. 6, no. 2, hlm. 69, Jul 2021, doi: 10.21111/fij.v6i2.5536.
- [4] M. Lestandy dan L. Syafa'ah, "PREDIKSI KASUS AKTIF COVID-19 MENGGUNAKAN METODE K-NEAREST NEIGHBORS," *Seminar Nasional Teknologi dan Rekayasa (SENTRA) 2020*, hlm. 45–48, 2020.
- [5] D. M. U. Atmaja, A. R. Hakim, D. Haryadi, dan N. Suwaryo, "PENERAPAN ALGORITMA K-NEAREST NEIGHBOR UNTUK PREDIKSI PENYEBARAN COVID-19 JAWA BARAT," vol. 1, hlm. 9, 2021.
- [6] A. Tangu Mara, E. Sedyono, dan H. Purnomo, "Penerapan Algoritma K-Nearest Neighbors Pada Analisis Sentimen Metode Pembelajaran Dalam Jaringan (DARING) Di Universitas Kristen Wira Wacana Sumba," *JOINTER - J. of Informatics Engineering*, vol. 2, no. 01, hlm. 24–31, Jun 2021, doi: 10.53682/jointer.v2i01.30.

- [7] I. M. Parapat dan M. T. Furqon, “Penerapan Metode Support Vector Machine (SVM) Pada Klasifikasi Penyimpangan Tumbuh Kembang Anak,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 10, hlm. 3163–3169, 2018.
- [8] I. A. Ropikoh, R. Abdulhakim, U. Enri, dan N. Sulistiyowati, “Penerapan Algoritma Support Vector Machine (SVM) untuk Klasifikasi Berita Hoax Covid-19,” *JAIC*, vol. 5, no. 1, hlm. 64–73, Jul 2021, doi: 10.30871/jaic.v5i1.3167.
- [9] Nugroho. K, “Confusion Matrix untuk Evaluasi Model pada Supervised Learning,” 2022. <https://ksnugroho.medium.com/confusion-matrix-untuk-evaluasi-model-pada-supervised-machine-learning-bc4b1ae9ae3f>.
- [10] Farm Andra, 2022, “Data seputar kasus perkembangan COVID-19 (virus Corona) di Pulau Sumatera per kota / kabupaten pada 30 April 2022,” 2022. [https://www.andrafarm.com/\\_andra.php?\\_i=daftar-co19-](https://www.andrafarm.com/_andra.php?_i=daftar-co19-)
- [11] X. Yayasan Multi Media, “Pahami Lebih Dalam untuk Siap Hadapi Industri Data,” 2020. <https://www.dqlab.id/data-analisis-machine-learning-untuk-proses-pengolahan-data>