Perbandingan Algoritma *K-Nearest Neighbor* dan *Logistic Regression* pada Analisis Sentimen terhadap Vaksinasi Covid-19 pada Media Sosial *Twitter* dengan Pelabelan *Vader* dan *Textblob*

1st Fadhilah Fazrin
Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia
fadhilahfzrn@student.telkomuniversity.ac.id

2nd Oktariani Nurul Pratiwi
Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia
onurulp@telkomuniversity.ac.id

3rd Rachmadita Andreswari
Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia
andreswari@telkomuniversity.ac.id

***Abstrak*— Pada analisis ini metode yang digunakan yaitu metode klasifikasi *K-Nearest Neighbor* dan metode klasifikasi *Logistic Regression* dengan data yang diambil pada aplikasi twitter. Penelitian ini mengkaji tingkat akurasi pada sentimen masyarakat mengenai vaksinasi Covid-19 dengan label positif dan negatif. Nilai AUC pada algoritma KNN dengan pelabelan *TextBlob* yaitu sebesar 0,765 dengan dan 0,768 untuk pelabelan *VaderSentiment* keduanya termasuk kedalam kriteria *fair classification*. Sementara itu, pada algoritma *Logistic Regression* menghasilkan akurasi sebesar 84,97% dengan perbandingan rasio 90:10 untuk pelabelan *TextBlob*, sementara untuk pelabelan *VaderSentiment* dengan perbandingan rasio 90:10 menghasilkan akurasi sebesar 85,22%. Kedua algoritma divalidasi menggunakan *K-Fold Cross Validation* dengan jumlah *fold* 10. Hasil perbandingan yang diperoleh saat melakukan evaluasi dengan *confusion matrix* menunjukkan bahwa algoritma *Logistic Regression* dengan pelabelan *VaderSentiment* memiliki nilai akurasi yang paling tinggi dibandingakan dengan algoritma *K-Nearest Neighbor* dengan pelabelan *TextBlob* dan *VaderSentiment*.**

Kata kunci*—*vaksinasi covid-19, k-nearest neighbor, logistic regression, analisis sentimen

I. PENDAHULUAN

Berdasarkan data kemenkes Indonesia terdapat 208,265,720 sasaran vakksinasi dan telah tercapai 110,406,777 (53.01%) dosisi pertama serta 65,173,148 (31.29%) dosis kedua [1]. Berdasarkan data tersebut, target vaksin yang ditetapkan pemerintah belum tercapai, dimana baru terdapat 36,59% masyarakat yang telah melakukan vaksin pada bulan November 2021 baik dosis satu maupun dosis dua. Sementara itu, pemerintah meningkatkan vaksinasi 70% pada akhir tahun 2021 terutama untuk vaksin dosis kedua. Respon masyarakat terhadap vaksinasi pemerintah sangat bervariasi. Analisis sentimen dapat memberikan informasi kepada banyak pihak, dimana analisis sentimen merupakan proses pengklasifikasian teks menjadi sentimen positif dan negatif. Pendapat tersebut haruslah dipertimbangkan sebagai bahan evaluasi agar program vaksinasi yang dilaksanakan dapat berjalan dengan baik dan pemerintah dapat melakukan sosialisasi lebih lanjut untuk meningkatkan kesadaran masyrakat akan vaksinasi Covid-19.

Metode yang banyak digunakan untuk melakukan analisis sentimen adalah algoritma *K-Nearest Neighbor* dan *Logistic Regression*. Pada *machine learning* kedua algoritma tersebut merupakan teknik *supervised learning*. Pada sudut pandang parameter algoritma yang harus dioptimasi, *K-Nearest Neighbor* merupakan kategori non-parametrik yaitu tidak mengasumsikan permasalahan, sementara *Logistic Regression* merupakan kategori parametrik dimana harus dilakukan reduksi permasalahan sebagai optimasi parameter. Sedangkan dari sudut pandang lainnya, *K-Nearest Neighbor* termasuk ke dalam model non-linear sementara *Logistic Regression* termasuk ke dalam model linear

Berdasarkan perbedaan tersebut maka dilakukan perbandingan performansi dan akurasi yang dihasilkan oleh kedua algoritma tersebut untuk menentukan algoritma mana yang memiliki performansi paling optimal dan akurasi paling tinggi. Pada penelitian ini digunakan dua metode pelabelan yaitu dengan pelabelan *TextBlob* dan *Vader*. Pemilihan kedua metode pelabelan tersebut karena *TextBlob* dan *Vader* adalah dua *library python* untuk pemrosesan teks. Bedanya di *TextBlob*, polaritasnya ditentukan dengan menghitung jumlah kalimat/ulasan positif dan negatif kemudian diberikan skor polaritas menggunakan fungsi *sentiment()*. Sementara itu, *Vader* menganalisis sepotong teks untuk melihat apakah ada kata dari teks dalam kamus *Vader*, serta penentuan indeks polaritas pada *Vader* menggunakan fungsi *polarity_score()*. Perbedaan karakteristik tersebut menyebabkan hasil pelabelan yang berbeda, sehingga dilakukan perbandingan untuk menentukan teknik pelabelan mana yang memberikan hasil optimal pada dataset dalam analisis sentimen program Vaksinasi Covid-19 pada Media Sosial *Twitter*.

II. KAJIAN TEORI

A. *Sentiment Analysis*

Sentiment analysis atau *opinion mining* mengacu pada bidang yang luas dari pengolahan bahasa alami, komputasi linguistik dan *text mining* yang bertujuan menganalisa pendapat, sentimen, evaluasi, sikap, penilaian dan emosi seseorang apakah pembicara atau penulis berkenan dengan suatu topik, produk, layanan, organisasi, individu, ataupun kegiatan tertentu. Tugas dasar dalam *sentiment analysis* adalah mengelompokkan polaritas dari teks yang ada dalam dokumen, kalimat, atau fitur/tingkat aspek apakah pendapat yang dikemukakan dalam dokumen, kalimat atau fitur entitas/aspek bersifat positif atau negatif. *Sentiment analysis*

dapat diklasifikasikan ke dalam kelas sentimen bersifat positif dan negatif [2].

1. Sentimen Positif: Menurut Kamus Besar Bahasa Indonesia (KBBI) sentimen positif merupakan reaksi atau sikap yang meningkatkan nilai seseorang atau sesuatu.
2. Sentimen Negatif: Menurut Kamus Besar Bahasa Indonesia (KBBI) sentimen negatif merupakan reaksi atau sikap yang menurunkan nilai seseorang atau sesuatu, jadi kalimat bersentimen negatif akan menyebabkan penyurutan nilai pandang terhadap sesuatu, sehingga membentuk tren down. Umumnya kalimat bersentimen negatif ditandai dengan penggunaan kata negasi. Negasi merupakan sesuatu yang dikenal dalam semua bahasa dan biasanya negasi digunakan untuk mengubah polaritas dari suatu pernyataan.

B. *K-Nearest Neighbor*

Klasifikasi *K-Nearest-Neighbor* (KNN) adalah salah satu metode klasifikasi yang paling mendasar dan sederhana dan harus menjadi salah satu pilihan pertama untuk studi klasifikasi ketika ada sedikit atau tidak ada pengetahuan sebelumnya tentang distribusi data. Klasifikasi *K-Nearest-Neighbor* dikembangkan dari kebutuhan untuk melakukan analisis diskriminan ketika estimasi parametrik yang andal dari kepadatan probabilitas tidak diketahui atau sulit ditentukan.[3]

Dalam pengaturan klasifikasi, algoritma KNN pada dasarnya menghitung suara mayoritas dari K contoh yang paling mirip dengan pengamatan "*unseen*" yang diberikan. Kesamaan didefinisikan menurut metrik jarak antara dua titik data. Pilihan yang populer adalah jarak *Euclidean* [4]

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_{training}^{i} - y_{tetsing}^{i})^2} \qquad (1)$$

Keterangan:
d(x,y) = jarak
$x_{training}^{i}$ = *data training*
$y_{tetsing}^{i}$ = *data testing*
i = variabel data
n = dimensi data

Semakin dekat dan mirip maka semakin kecil jarak antara dua titik tersebut. *Euclidean Distance* dikatakan baik jika data baru memiliki jarak minimum dan memiliki kemiripan yang tinggi. Hasil perhitungan *Euclidean Distance* dapat digunakan untuk mengetahui nilai jarak antara *data training* dan *data testing*. Setelah itu dilakukan pengurutan data dari paling kecil ke besar, kemudian menggunakan nilai K untuk batas ruang jarak tetangga terdekat, dari batas jumlah data yang diambil berdasarkan nilai k, nilai yang mayoritas diambil sebagai hasil prediksi [5].

C. *Logistic Regression*

Logistic Regression adalah teknik pemodelan prediktif klasik dan masih tetap menjadi pilihan populer untuk pemodelan variabel kategori *biner*. *Logistic Regression* adalah padanan klasifikasi dari regresi linier. Ini juga merupakan salah satu algoritma pembelajaran mesin terawasi yang paling populer. Ini digunakan untuk memprediksi variabel dependen kategoris menggunakan satu set variabel independen yang diberikan. *Logistic Regression* adalah cara yang efisien dan ampuh untuk menganalisis efek dari sekelompok variabel independen dengan hasil biner dengan mengukur kontribusi unik setiap variabel independen untuk memprediksi *output* dari variabel dependen kategoris. Oleh karena itu, hasil harus kategoris atau harus memiliki nilai diskrit. Masing-masing dapat berupa ya atau tidak, 0 atau 1, benar atau salah, dan sebagainya. Tetapi alih-alih memberikan nilai yang tepat seperti 0 atau 1, ini memberikan nilai probabilistik yang terletak antara 0 dan 1. Regresi linier digunakan untuk masalah regresi, sedangkan regresi logistik digunakan untuk menyelesaikan masalah klasifikasi. Nama "Regresi" menyiratkan bahwa model linier harus sesuai dengan ruang linier. Prediksi dipetakan antara 0 dan 1 melalui fungsi logistik [4].

Proses klasifikasi menggunakan *Logistic Regression* dilakukan dengan mengekstrak fitur bernilai *real* dari input, mengalikan masing-masing dengan bobot, menjumlahkannya dan melewatkan jumlah tersebut melalui fungsi *sigmoid* untuk menghasilkan probabilitas. Nilai ambang digunakan untuk membuat keputusan. *Logistic Regression* mampu mengklasifikasikan sentimen menjadi dua kelas dengan label positif dan negatif atau kelas ganda menggunakan *Logistic Regression Multinomial* [6].

D. *TextBlob*

Lisensi *textblob* dipegang oleh Steven Loria untuk tahun 2013-2020 [7]. *Textblob* adalah pustaka *python* yang menyediakan penambangan teks, analisis teks, dan modul pemrosesan teks untuk pengembang *python*. *Textblob* menggunakan kembali *corpora* NLTK, dan jika NLTK telah diinstal sebelum *Textblob*, maka *Textblob* akan diinstal dengan sangat mudah. *Textblob* adalah analisis tingkat kalimat. Pertama, dibutuhkan dataset sebagai input kemudian membagi *review* menjadi kalimat. Cara umum untuk menentukan polaritas untuk seluruh kumpulan data adalah dengan menghitung jumlah kalimat/ulasan positif dan negatif dan memutuskan apakah tanggapannya positif dan negatif berdasarkan jumlah total ulasan positif dan negatif. Polaritas dan subjektivitas *review* yang diberikan dapat diketahui dengan menggunakan fungsi *sentimen()*. Ini mengembalikan *tuple* bernama dengan dua parameter yang disebut polaritas dan subjektivitas. Skor polaritas berkisar dari -1 hingga 1 dan rentang subjektivitas adalah dari 0 hingga 1 di mana 0 paling objektif dan 1 paling subjektif [8].

E. *Vader*

Vader adalah leksikon dan alat analisis sentimen berbasis aturan. *Vader* menggunakan kombinasi leksikon sentimen, daftar fitur leksikal yang umumnya diberi label sesuai dengan orientasi semantiknya sebagai positif atau negatif. *Vader* telah cukup berhasil ketika berurusan dengan teks media sosial, ulasan film, dan ulasan produk. Ini karena *Vader* tidak hanya menceritakan tentang skor positif dan negatif tetapi juga menceritakan tentang seberapa positif atau negatif sebuah sentimen. Pengembang *Vader* telah menggunakan Turk Mekanik Amazon untuk mendapatkan sebagian besar peringkat. Keuntungan *Vader* yaitu bekerja dengan sempurna pada teks jenis media sosial, tidak memerlukan data pelatihan apa pun tetapi dibangun dari leksikon standar emas yang digeneralisasikan, berbasis valensi, dan dikuratori oleh manusia, *Vader* mendukung emoji untuk klasifikasi

sentimen, cukup cepat untuk digunakan secara online dan tidak terlalu menderita dari *tradeoff* kecepatan-kinerja [8].

Vader menganalisis sepotong teks untuk melihat apakah ada kata dari teks yang ada dalam leksikon *Vader*. Serta dapat menemukan indeks polaritas menggunakan fungsi *polarity_scores()* yang akan mengembalikan nilai metrik negatif, netral, positif, dan majemuk untuk kalimat tertentu. Skor majemuk adalah metrik yang menghitung jumlah semua peringkat leksikon yang telah dinormalisasi antara -1 dan +1 di mana -1 menunjukkan negatif paling ekstrem dan +1 menunjukkan positif paling ekstrem. Hal ini berguna untuk menetapkan ambang standar untuk mengklasifikasikan kalimat sebagai positif, netral atau negatif. Nilai ambang tipikal diberikan di bawah ini [8]

a. Sentimen Positif: skor majemuk >= 0,05
b. Sentimen Netral: skor majemuk > -0,05 dan <0,05
c. Sentimen Negatif: skor majemuk <= -0,05

Vader dikembangkan oleh Gilbert pada tahun 2014 [9].

F. *K-Fold Cross Validation*

Menurut Rohani, Abbas., et al. (dalam Jiang, Ping., 2017) *K-Fold Cross Validation* adalah salah satu dari jenis pengujian *cross validation* yang berfungsi untuk menilai kinerja proses sebuah metode algoritma dengan membagi sampel data secara acak dan mengelompokkan data tersebut sebanyak nilai K *k-fold*. Kemudian salah satu kelompok *k-fold* tersebut akan dijadikan sebagai data uji sedangkan sisa kelompok yang lain akan dijadikan sebagai data latih [10].

Dalam *K-Fold Cross Validation*, dataset X dibagi secara acak menjadi K bagian yang berukuran sama. Untuk menghasilkan setiap pasangan, harus mengeluarkan salah satu bagian K sebagai set validasi, dan menggabungkan bagian K-1 yang tersisa untuk membentuk set pelatihan. Melakukan K kali, setiap kali meninggalkan salah satu bagian K lainnya. Ada dua masalah dengan ini: Pertama, untuk menjaga set pelatihan tetap besar, harus mengizinkan set validasi yang kecil. Kedua, set pelatihan saling tumpang tindih, yaitu, dua set pelatihan berbagi bagian K-2. K biasanya 10 atau 30. Saat K meningkat, persentase *instance* pelatihan meningkat dan akan mendapatkan lebih banyak estimator *rebost*, tetapi set validasi menjadi lebih kecil. Selanjutnya, ada biaya pelatihan pengklasifikasi K kali, yang meningkat dengan meningkatnya K. Saat N meningkat, K bisa lebih kecil; jika N kecil, K harus besar untuk memungkinkan set pelatihan yang cukup besar. Satu kasus ekstrim *K-Fold Cross Validtaion* adalah *leave-one-out* di mana set data yang diberikan dari N *instance*, hanya satu *instance* yang ditinggalkan sebagai set validasi (*instance*) dan pelatihan menggunakan instance N-1. *Leave-one-out* tidak mengizinkan stratifikasi [11].

G. *Confusion Matrix*

Confusion matrix adalah tabel yang menyatakan klasifikasi jumlah data uji yang benar dan jumlah data uji yang salah. Contoh *confusion matrix* untuk klasifikasi biner sebagai berikut [12].

TABEL 1
CONTOH CONFUSION MATRIX

		Actual Values: Positive (1)	Actual Values: Negative (0)
Predictive Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Dengan keterangan sebagai berikut:

a. *True Positive*: memprediksi positif dan itu benar
b. *True Negative*: memprediksi negatif dan itu benar
c. *False Positive*: (Kesalahan Tipe 1), memprediksi positif dan itu salah
d. *False Negative*: (Kesalahan Tipe 2), memprediksi negatif dan itu salah

III. METODE

Pada penelitian ini, sistematika penyelesain yang digunakan yaitu *Knowledge Discovery in Database* (KDD). Terdapat beberapa tahapan yaitu, identifikasi masalah, *data selection*, *data preprocessing*, *data mining* dan evaluasi. Berikut tahapan metode penelitian:

a. Inisiasi
Pada tahap awal ini, hal pertama yang dilakukan adalah mengidentifikasi dan merumuskan masalah mengenai opini masyarakat terhadap vaksinasi Covid-19. Berdasarkan permasalahan yang diperoleh, ditemukan solusi melalui studi literatur dan pemilihan algoritma yang akan digunakan untuk mengklasifikasikan sentimen masyarakat.

b. *Data Selection*
Tahap selanjutnya adalah memilih data, terutama mengidentifikasi kata kunci yang berkaitan dengan vaksinasi Covid-19 dan melakukan *crawling* pada media sosial *twitter*, untuk mendapatkan dataset yang akan digunakan pada tahapan selanjutnya.

c. *Data Preprocessing*
Selanjutnya adalah tahap *preprocessing*, yang meliputi penghapusan data duplikat, pengecekatan inkonsisten data dan memperbaiki kesalahan pada data..

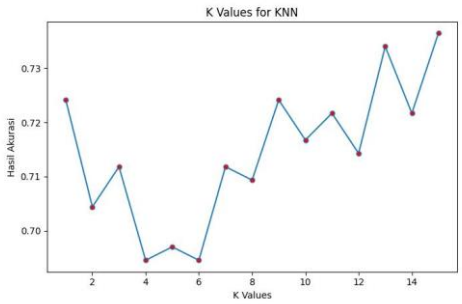
d. *Data Mining*
Pada tahap *data mining*, pelabelan dilakukan dengan menggunakan library python yaitu *Vader* dan *TextBlob*. Kemudian dilakukan pembobotan TF-IDF, splitting data dan implementasi model algortima *K-Nearest Neighbor* dan *Logistic Regression*.

e. *Evaluation*
Pada tahap evaluasi, hasil klasifikasi analisis sentimen akan di cek nilai *accuracy*, *recall*, *precision* dan *F1-Score* untuk mengetahui performansi algortima *K-Nearest Neighbor* dan algoritma *Logistic Regression*. Evaluasi ini menggunakan *Confusion Matrix* dan sistem akan menampilkan hasil evaluasi algoritma dalam bentuk *pie chart*, *bar chart* dan kurva ROC AUC. Oleh karena itu, dengan melakukan evaluasi, dimungkinkan untuk menarik kesimpulan terhadap analisis sentimen serta evaluasinya.

IV. HASIL DAN PEMBAHASAN

A. Klasifikasi K-Nearest Neighbor

Implementasi pertama yaitu menggunakan algoritma *K-Nearest Neighbor* menggunakan library yang tersedia pada bahasa pemrograman python. Sebelum melakukan implementasi terlebih dahulu dilakukan pencarian nilai K terbaik dengan range 1-15. Dimana nilai K terbaik yang didapatkan akan masuk ke dalam model KNN yang menunjukan kedekatan ketetanggan. Berikut adalah grafik hasil perhitungan nilai K terbaik:



GAMBAR 1
(NILAI K UNTUK KNN)

Berdasarkan grafik di atas, diketahui bahwa nilai K terbaik yaitu 15. Setelah itu juga diketahui nilai maksimum akurasi pada K=15 adalah 0.7364 atau 74%. Nilai K tersebut akan digunakan pada implementasi algoritma KNN

1. Hasil Akurasi

Berdasarkan nilai K yang diperoleh sebelumnya yaitu 15. Dilakukan implementasi algoritma KNN pada dataset dengan pelabelan *TextBlob* dan pelabelan *VaderSentiment* dengan rasio 70:30, 80:20, dan 90:10, dan pengujian yang dilakukan memberikan hasil perbandingan akurasi sebagai berikut:

TABEL 2
(HASIL AKURASI KNN)

Ratio	**Labeling**	
	TextBlob	**Vader**
70:30	0.6937	0.6609
80:20	0.7056	0.6687
90:10	0.7364	0.7019

Pada tabel di atas, dapat dilihat bahwa hasil akurasi terbaik dari algoritma KNN dengan nilai K=15 diperoleh dari rasio 90:10 dengan metode pelabelan *TextBlob* yang menghasilkan nilai akurasi sebesar 73,64% . Sementara untuk pelabelan *VaderSentiment* diperoleh nilai akurasi sebesar 70,19% untuk ratio 90:10.

2. K-Fold Validation

K-Fold Cross Validation digunakan untuk memvalidasi hasil uji klasifikasi pada dataset. Dalam pengujian ini, jumlah *fold* yang digunakan adalah 10. Hasil dari setiap *fold* akan dirata-ratakan sebagai akurasi pengujian. Berikut perbandingan hasil skor *K-Fold Cross Validation*:

TABEL 3
(K-FOLD PADA KNN)

Fold	**Pelabelan**	
	TextBlob	**Vader**
1	0.714	0.600
2	0.714	0.697
3	0.748	0.674
4	0.721	0.660
5	0.679	0.677
6	0.687	0.640
7	0.662	0.684
8	0.706	0.721
9	0.716	0.650
10	0.711	0.701
Rata-rata	**0.706**	**0.670**

Berdasarkan hasil akurasi 10 *fold* dalam *K-Fold Cross Validation* diperoleh nilai rata-rata sebesar 71% untuk pelabelan *TextBlob* dan 67% untuk pelabelan *VaderSentiment* dengan standar deviasi 0.02. Hasil akurasi ini menghasilkan skor yang lebih rendah dibandingkan dengan pengujian menggunakan splitting data.

3. Confusion Matrix

Berdasarkan hasil pengujian KNN pada dataset yang menggunakan pelabelan *TextBlob* dengan akurasi 74% pada data splitting 90:10, dilakukan pengukuran performa model menggunakan *confusion matrix*, dengan hasil sebagai berikut:

TABEL 4 (CM KNN - TEXTBLOB)

	Predicted Negative	**Predicted Positive**
Actual Negative	271 TN	4 FP
Actual Positive	103 FN	28 TP

Dari hasil *confusion matrix* tersebut, dapat dilakukan perhitungan *accuracy, precision, recall* dan *F1-Score* dan menghasilkan classification report sebagai berikut:

TABEL 5
(CR KNN - TEXTBLOB)

	Precision	**Recall**	**F1-Score**
Positif	88%	21%	34%
Negatif	72%	99%	84%
Accuracy	74%		

Sementara pada dataset yang menggunakan pelabelan *VaderSentiment* dengan akurasi 70% pada data splitting 90:10, dilakukan pengukuran performa model menggunakan *confusion matrix*, dengan hasil sebagai berikut:

TABEL 6 -
(CM KNN - VADER)

	Predicted Negative	**Predicted Positive**
Actual Negative	240 TN	11 FP
Actual Positive	110 FN	45 TP

Dari hasil *confusion matrix* tersebut, dapat dilakukan perhitungan *accuracy, precision, recall* dan *F1-Score* dan menghasilkan *classification report* sebagai berikut:

TABEL 7
(CR KNN - VADER)

	Precision	**Recall**	**F1-Score**

Positif	80%	29%	43%
Negatif	69%	96%	80%
Accuracy	70%		

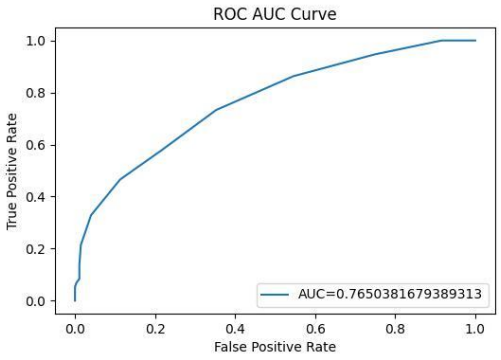
4. Kurva ROC dan Nilai AUC

Setelah mengukur kinerja model dengan menggunakan *confusion matrix*, pengukuran juga dilakukan dengan menampilkan informasi kinerja algoritma klasifikasi menggunakan kurva *Receiver Operating Characteristic* (ROC) dan menghitung skor *Area Under the Curve* (AUC). ROC (*Receiver Operating Characteristic*) adalah cara untuk menggambarkan, mengatur dan mengklasifikasikan beberapa kategori yang ditentukan dalam model statistik berdasarkan kinerja. Sementara AUC (*Area Under the Curve*) merupakan area di bawah kurva ROC, yang memberikan gambaran tentang keseluruhan pengukuran atas kesesuain dari model yang digunakan. Dengan panduan tingkat akurasi model pada AUC:

TABEL 8
(NILAI AKURASI ROC)

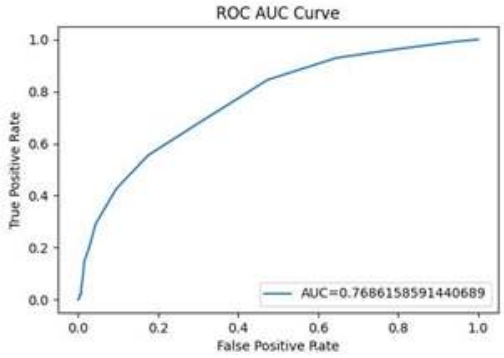
Nilai AUC	**Klasifikasi Performance**
0.90 - 1.00	Very Good
0.80 - 0.90	Good
0.70 - 0.80	Fair
0.60 - 0.70	Low
0.50 - 0.60	Fail

Grafik dari kurva ROC dan hasil skor AUC pada algoritma KNN dengan pelabelan TextBlob adalah sebagai berikut:



GAMBAR 2
(ROC KNN - TEXTBLOB)

Kurva ROC yang dihasilkan dari algoritma KNN menggunakan pelabelan *TextBlob* menampilkan grafik yang dapat digunakan untuk menghitung skor dari AUC dengan hasil skor 0.7650. Nilai yang dihasilkan dari AUC termasuk ke dalam kriteria *fair classification*. Sementara grafik dari kurva ROC dan hasil skor AUC pada algoritma KNN dengan pelabelan *Vader* adalah sebagai berikut:



GAMBAR 3
(ROC KNN- VADER)

Kurva ROC yang dihasilkan dari algoritma KNN menggunakan pelabelan *VaderSentiment* menampilkan grafik yang dapat digunakan untuk menghitung skor dari AUC dengan hasil skor 0.7686. Nilai yang dihasilkan dari AUC termasuk ke dalam kriteria *fair classification*.

Berdasarkan kurva ROC, pemodelan algoritma KNN menggunakan teknik pelabelan *VaderSentiment* menghasilkan nilai AUC yang lebih tinggi dibandingkan dengan teknik pelabelan *TextBlob*.

B. Klasifikasi Logistic Regression

Implementasi kedua yaitu menggunakan algoritma *Logistic Regression* terhadap dataset yang telah dilakukan pembagian *data training* dan *data testing*, data tersebut dibagi berdasarkan jumlah rasio penggunaan data. Implementasi *Logistic Regression* menggunakan *library python*

1. Hasil Akurasi

Pengujian dilakukan dengan perbandingan tiga rasio terhadap dataset yang telah dilakukan pelabelan dengan *VaderSentiment* dan *TextBlob*. Berikut nilai akurasi pada algoritma *Logistic Regression*:

TABEL 9
(HASIL AKURASI LOGISTIC REGRESSION)

Ratio	**Labelling**	
	TextBlob	**Vader**
70:30	0.7873	0.7996
80:20	0.8115	0.8165
90:10	0.8497	0.8522

Pada tabel di atas, dapat dilihat bahwa hasil akurasi terbaik dari algoritma *Logistic Regression* diperoleh dari rasio 90:10 dengan metode palabelan *VaderSentiment* yang menghasilkan nilai akurasi sebesar 85%. Sementara untuk pelabelan *TextBlob* diperoleh nilai akurasi sebesar 85% dengan akurasi 90:10

2. K-Fold Validation

Dalam pengujian ini, jumlah *fold* yang digunakan adalah 10. Hasil dari setiap *fold* akan dirata-ratakan sebagai akurasi pengujian. Berikut hasil skor *K-Fold Cross Validation* serta grafik performa *K-Fold Cross Validation*:

TABEL 10
(K-FOLD PADA LOGISTIC REGRESSION)

Fold	Pelabelan	
	TextBlob	Vader
1	0.800	0.763
2	0.790	0.815
3	0.822	0.805
4	0.807	0.793
5	0.788	0.807
6	0.778	0.842
7	0.780	0.820
8	0.790	0.832
9	0.812	0.810
10	0.783	0.810
Rata-Rata	**0.795**	**0.810**

Berdasarkan hasil akurasi 10 *fold* dalam *K-Fold Cross Validation* diperoleh nilai rata-rata sebesar 80% untuk pelabelan *TextBlob* dan 81% untuk pelabelan *VaderSentiment* dengan standar deviasi 0.02. Hasil akurasi ini menghasilkan skor yang lebih rendah dibandingkan dengan pengujian menggunakan *splitting data*.

3. Confusion Matrix

Berdasarkan hasil pengujian *Logistic Regression* pada dataset yang menggunakan pelabelan *TextBlob* dengan akurasi 85% pada *data splitting* 90:10, dilakukan pengukuran performa model menggunakan *confusion matrix*, dengan hasil sebagai berikut:

TABEL 11
(CM LOGISTIC REGRESSION - TEXTBLOB)

	Predicted Negative	Predicted Positive
Actual Negative	270 TN	5 FP
Actual Positive	56 FN	75 TP

Dari hasil *confusion matrix* tersebut, dapat dilakukan perhitungan *accuracy, precision, recall* dan *F1-Score* dan menghasilkan classification report sebagai berikut:

TABEL 12
(CR LOGISTIC REGRESSION - TEXTBLOB)

	Precision	Recall	F1-Score
Positif	94%	57%	71%
Negatif	83%	98%	90%
Accuracy	85%		

Sementara pada dataset yang menggunakan pelabelan *VaderSentiment* dengan akurasi 85% pada *data splitting* 90:10, dilakukan pengukuran performa model menggunakan *confusion matrix*, dengan hasil sebagai berikut:

TABEL 13
(CM LOGISTIC REGRESSION - VADER)

	Predicted Negative	Predicted Positive
Actual Negative	235 TN	16 FP
Actual Positive	44 FN	111 TP

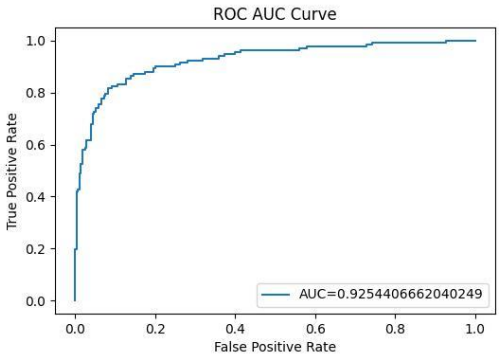
Dari hasil *confusion matrix* tersebut, dapat dilakukan perhitungan *accuracy, precision, recall* dan *F1-Score* dan menghasilkan *classification report* sebagai berikut:

TABEL 14
(CR LOGISTIC REGRESSION - VADER)

	Precision	Recall	F1-Score
Positif	87%	72%	79%
Negatif	84%	94%	89%
Accuracy	85%		

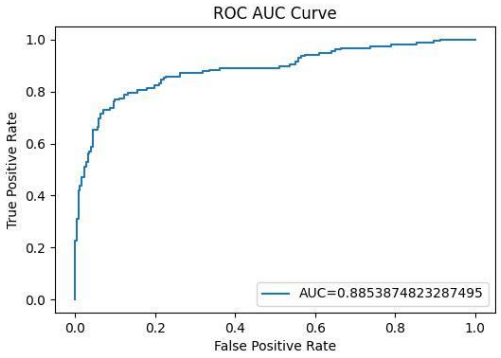
4. Kurva ROC dan Nilai AUC

Setelah mengukur kinerja dengan *confusion matrix*, pengukuran dapat juga dilakukan dengan menampilkan informasi kinerja algoritma klasifikasi dalam bentuk grafik menggunakan kurva ROC dan perhitungan skor AUC. Berikut merupakan grafik dari kurva ROC dan hasil skor AUC pada algoritma *Logistic Regression*:



GAMBAR 4
(ROC LOGISTIC REGRESSION - TEXTBLOB)

Kurva ROC yang dihasilkan dari algoritma *Logistic Regression* menggunakan pelabelan *TextBlob* menampilkan grafik yang dapat digunakan untuk menghitung skor dari AUC dengan hasil skor 0.9254. Nilai yang dihasilkan dari AUC termasuk ke dalam kriteria *very good classification*. Sementara grafik dari kurva ROC dan hasil skor AUC pada algoritma *Logistic Regression* dengan pelabelan *VaderSentiment* adalah sebagai berikut:



GAMBAR 5
(ROC LOGISTIC REGRESSION - VADER)

Kurva ROC yang dihasilkan dari algoritma *Logistic Regression* menggunakan pelabelan *VaderSentiment* menampilkan grafik yang dapat digunakan untuk menghitung

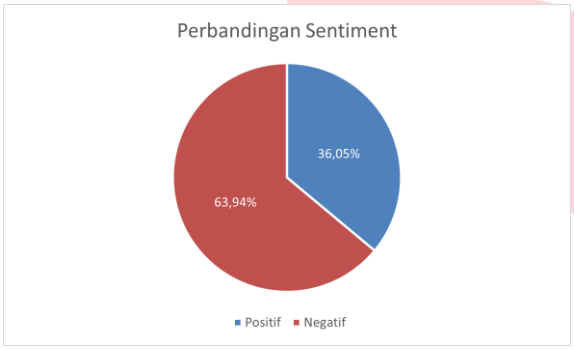
skor dari AUC dengan hasil skor 0.8853. Nilai yang dihasilkan dari AUC termasuk ke dalam kriteria *good classification*.

Berdasarkan kurva ROC, pemodelan algoritma *Logistic Regression* menggunakan teknik pelabelan *VaderSentiment* menghasilkan nilai AUC yang lebih tinggi dibandingkan dengan teknik pelabelan *TextBlob*.

C. Analisis Sentimen

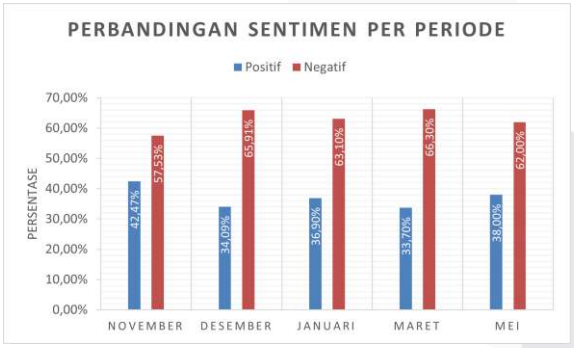
1. TextBlob

Pelabelan menggunakan library *TextBlob* pada *python*, mendapatkan 2596 *tweet* dengan sentiment negatif dan 1464 *tweet* dengan sentimen positif dari total data 4060 *tweet*. Visualisasi perbandingan sentimen positif dan negatif dengan pelabelan *TextBlob* sebagai berikut:



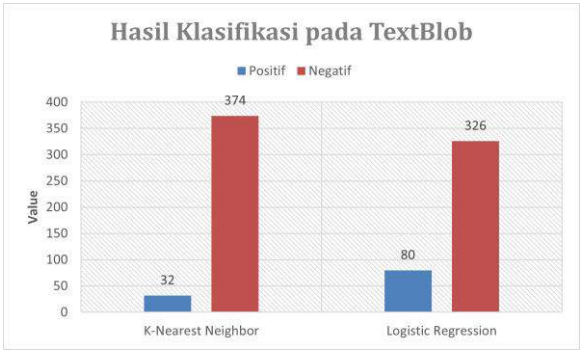
GAMBAR 6
(PERBANDINGAN SENTIMEN TEXTBLOB)

Berdasarkan grafik di atas, terdapat 63,94% sentimen negatif dan 36,05% sentimen positif dari total 4.060 data, yang menunjukkan bahwa banyak orang memiliki pendapat negatif tentang vaksinasi Covid-19. Berikut adalah visualisasi sentimen yang dikelompokkan berdasarkan periode waktu (bulan) pada pelabelan *TextBlob*.



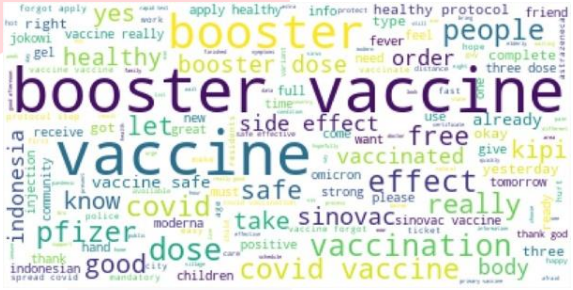
GAMBAR 7
(SENTIMEN TEXTBLOB PER PERIODE)

Berdasarkan Gambar 7, diketahui bahwa sentimen negatif terhadap vaksin Covid-19 setiap bulannya paling tinggi dibandingkan dengan sentimen positif. Sentimen negatif tertinggi terjadi selama periode Maret sebesar 66,30%. Dari hasil implementasi algoritma *K-Nearest Neighbor* dan algoritma *Logistic Regression* pada pelabelan *TextBlob* didapatkan gambaran visual hasil klasifikasi dengan label positif dan negatif dari masing-masing algoritma, sebagai berikut:



GAMBAR 8
(HASIL KLASIFIKASI PADA TEXTBLOB)

Berdasarkan Gambar 8, pada kedua algoritma klasifikasi negatif lebih banyak daripada klasifikasi positif. Dimana banyak masyarakat yang memiliki pendapat negatif atau kontradiktif terhadap program vaksinasi Covid-19. Hasil klasifikasi yang diperoleh berdasarkan pada *confusion matrix*. Dengan *wordcloud* dibuat sebagai berikut:



GAMBAR 9
(WORDCLOUD POSITIF PADA TEXTBLOB)

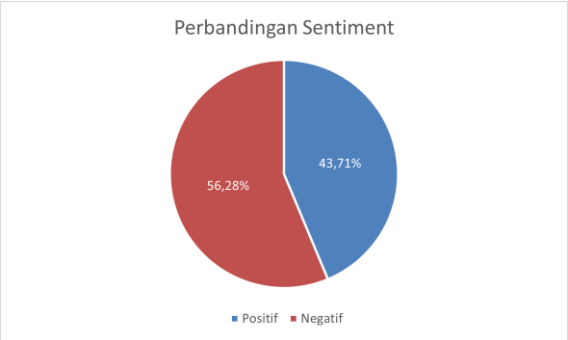
Berdasarkan Gambar 9, kata yang sering muncul pada sentimen positif dalam dataset yang menggunakan pelabelan *TextBlob* adalah *"vaccine" "booster" "healthy" "healthy protocol" "side effect" "three dose" "good"*.

GAMBAR 10
(WORDCLOUD NEGATIF PADA TEXTBLOB)

Sementara itu, berdasarkan Gambar 10, kata yang sering muncul pada sentimen negatif dalam dataset yang menggunakan pelabelan *TextBlob* adalah *"vaccine" "booster vaccine" "sick" "injection" "side effect" "booster dose"*

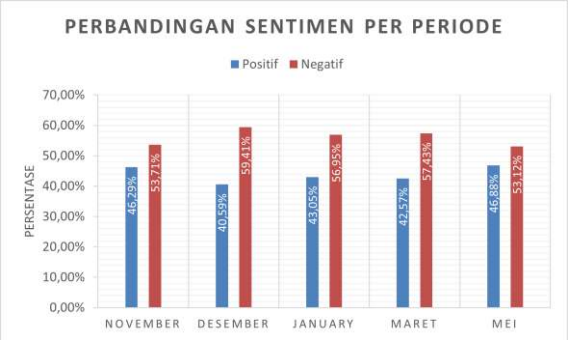
2. VaderSentiment

Pelabelan menggunakan *library VaderSentiment* pada *python*, mendapatkan 2285 *tweet* dengan sentimen negatif dan 1775 *tweet* dengan sentimen positif dari total data 4060 *tweet*. Visualisasi perbandingan sentimen positif dan negatif dengan pelabelan *VaderSentiment* sebagai berikut:



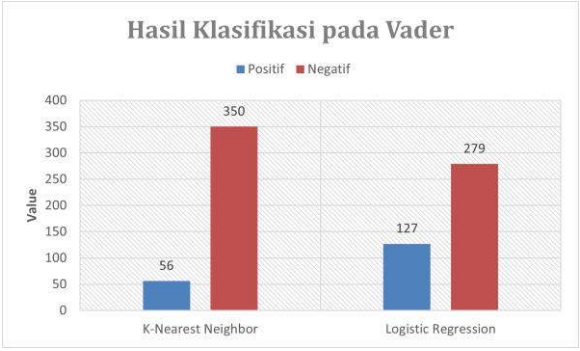
GAMBAR 11
(PERBANDINGAN SENTIMEN VADER)

Berdasarkan grafik di atas, terdapat 56,28% sentimen negatif dan 43,71% sentimen positif dari total 4.060 data, sama hal nya dengan pelabelan *TextBlob* pada pelabelan *VaderSentiment* juga menunjukkan bahwa banyak orang memiliki pendapat negatif tentang vaksinasi Covid-19. Berikut adalah visualisasi sentimen yang dikelompokkan berdasarkan periode waktu (bulan) pada pelabelan *VaderSentiment*.



GAMBAR 12
(SENTIMEN VADER PER PERIODE)

Berdasarkan Gambar 12, tidak jauh berbeda dengan pelabelan *TextBlob* sebelumnya dimana sentimen negatif terhadap vaksin Covid-19 setiap bulannya paling tinggi dibandingkan dengan sentimen positif. Sentimen negatif tertinggi terjadi selama periode Desember sebesar 59,41%. Dari hasil implementasi algoritma *K-Nearest Neighbor* dan algoritma *Logistic Regression* pada pelabelan *VaderSentiment* didapatkan gambaran visual hasil klasifikasi dengan label positif dan negatif dari masing-masing algoritma, sebagai berikut:

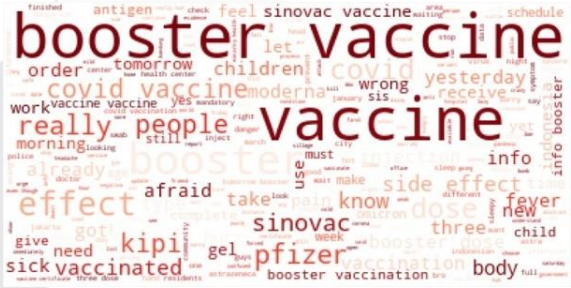


GAMBAR 13
(HASIL KLASIFIKASI PADA VADER)

Berdasarkan Gambar 13, pada kedua algoritma klasifikasi negatif lebih banyak daripada klasifikasi positif, sama seperti pada pelabelan *TextBlob*. Dimana banyak masyarakat yang memiliki pendapat negatif atau kontradiktif terhadap program vaksinasi Covid-19. Hasil klasifikasi yang diperoleh berdasarkan pada *confusion matrix*. Dengan *wordcloud* dibuat sebagai berikut:

GAMBAR 14
(WORDCLOUD POSITIF PADA VADER)

Berdasarkan Gambar 14, kata yang sering muncul pada sentimen positif dalam dataset yang menggunakan pelabelan *Vader* adalah *"booster vaccine" "vaccine" "thank god" "effect" "side effect" "booster dose" "healthy" "vaccine safe".*

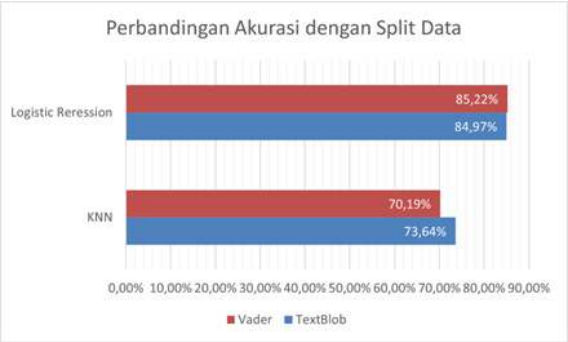


GAMBAR 15
(WORDCLOUD NEGATIF PADA VADER)

Berdasarkan Gambar 15, kata yang sering muncul pada sentimen negatif dalam dataset yang menggunakan pelabelan *Vader* adalah *"booster vaccine" "vaccine" "kipi" "sick" "side effect" "afraid" "effect".*

3. Perbandingan Akurasi

Dari hasil pengujian hasil akurasi dan evaluasi model pada algoritma *K-Nearest Neighbor* dan *Logistic Regression* menggunakan pelabelan *TextBlob* dan *VaderSentiment*, kinerja dari masing-masing algoritma dibandingkan dan dapatkan hasil sebagai berikut:



GAMBAR 16
(PERBANDINGAN DATA SPLITTING)

Berdasarkan gambar di atas, akurasi paling tinggi berada pada algoritma *Logistic Regression* dengan pelabelan *VaderSentiment* sebesar 85,22% dan memiliki selisih sebesar 0,25% dengan pelabelan *TextBlob*.

V. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan dengan menggunakan algoritma *K-Nearest Neighbor* dan algoritma *Logistic Regression* untuk klasifikasi sentimen terkait vaksinasi Covid-19, dapat disimpulkan bahwa.

Implementasi algoritma *K-Nearest Neighbor* menggunakan nilai K=15. Nilai K ini merupakan jumlah tetangga terdekat yang digunakan dalam proses kalsifikasi. Sementara itu, implementasi algoritma *Logistsic Regression* menggunakan parameter default dari *library python*. Kedua algoritma divalidasi menggunakan *K-Fold Cross Validation* dengan jumlah *fold* 10.

Algoritma *Logistic Regression* menghasilkan akurasi tertinggi dengan pelabelan *VaderSentiment* dan menjadi algoritma dengan performa terbaik dengan palebalan *TextBlob*. Dimana algoritma *Logistic Regression* dengan pelabelan *VaderSentiment* dan rasio perbandingan 90:10 memiliki akurasi tertinggi sebesar 85,22%. Sedangkan algoritma *Logistic Regression* dengan pelabelan *TextBlob* memiliki performanasi kinerja terbaik dengan nilai AUC sebesar 0,9254 termasuk ke dalam kriteria *very good classification.*

Hasil perbandingan sentimen positif dan sentimen negatif pada kedua algoritma adalah sentimen negatif lebih besar dari sentimen postif. Terdapat 64% sentimen negatif pada hasil klasifikasi menggunakan *TextBlob* dan terdapat 56% sentimen negatif pada hasil klasifikasi menggunakan *VaderSentiment*. Hal itu dikarenakan banyaknya masyarakat yang tidak setuju dengan program vaksinasi Covid-19.

REFERENSI

[1] Kementrian Kesehatan, "Vaksinasi Covid-19 Nasional," *https://vaksin.kemkes.go.id/#/vaccines*, Nov. 2021.

[2] L. Ardiani, H. Sujaini, and T. Tursina, "Implementasi Sentiment Analysis Tanggapan Masyarakat Terhadap Pembangunan di Kota Pontianak," *Jurnal Sistem dan Teknologi Informasi (Justin)*, vol. 8, no. 2, p. 183, Apr. 2020, doi: 10.26418/justin.v8i2.36776.

[3] Leif E. Peterson, "K-nearest neighbor," *http://scholarpedia.org/*, Feb. 21, 2009.

[4] Pramod Gupta and Naresh K Sehgal, *Introduction to Machine Learning in the Cloud with Python: Concepts and Practices*. Switzherland: Springer International Publishing, 2021.

[5] A. Yudhana and dan Agus Jaka Sri Hartanta, "ALGORITMA K-NN DENGAN EUCLIDEAN DISTANCE UNTUK PREDIKSI HASIL PENGGERGAJIAN KAYU SENGON," *TRANSMISI*, vol. 22, no. 4, doi: 10.14710/transmisi.22.4.107-141.

[6] Imamah and F. H. Rachman, "Twitter sentiment analysis of Covid-19 using term weighting TF-IDF and logistic regresion," in *Proceeding - 6th Information Technology International Seminar, ITIS 2020*, Oct. 2020, pp. 238–242. doi: 10.1109/ITIS50118.2020.9320958.

[7] "textblob".

[8] V. Bonta, N. Kumaresh, and N. Janardhan, "A Comprehensive Study on Lexicon Based Approaches for Sentiment Analysis," 2019. [Online]. Available: www.rottentomatoes.

[9] C. J. Hutto and E. Gilbert, "VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text," 2014. [Online]. Available: http://sentic.net/

[10] A. Hutapea and M. Tanzil Furqon, "Penerapan Algoritme Modified K-Nearest Neighbour Pada Pengklasifikasian Penyakit Kejiwaan Skizofrenia," 2018. [Online]. Available: http://j-ptiik.ub.ac.id

[11] E. Alpaydin, *Introduction to Machine Learning*. London: MIT Press, 2004.

[12] D. Normawati and S. A. Prayogi, "Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter," 2021.