ANALISIS SENTIMEN VAKSINASI COVID-19 DI INDONESIA DENGAN METODE SUPPORT VECTOR MACHINE (SVM)

Lista Kurniawati[1], Elkania Samanta Nagani[2], dan Salsabil Felia Armansyah[3]

Departemen Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam,

Universitas Indonesia, Depok, Indonesia

[1]lista.kurniawati@sci.ui.ac.id

[2]elkania.samanta@sci.ui.ac.id

[3]salsabil.felia@sci.ui.ac.id

Keywords :
Analisis sentimen
Vaksinasi COVID-19
Twitter
SVM
Akurasi

ABSTRAK

Coronavirus Disease 2019 (COVID-19) telah ditetapkan pemerintah sebagai bencana non-alam yang menjadi ancaman dalam kehidupan masyarakat. Tentunya berbagai usaha pencegahan pun diupayakan oleh pemerintah. Salah satu pencegahan yang dilakukan adalah vaksinasi COVID-19. Pemerintah, melalui Kemenkes RI telah mengeluarkan kebijakan terkait pelaksanaan vaksinasi COVID-19 di Indonesia. Kejadian ini menimbulkan pro-kontra dalam masyarakat. Masyarakat mengekspresikan opini-opini mereka melalui sosial media, salah satunya *Twitter*. Penelitian ini membahas mengenai analisis sentimen vaksinasi COVID-19 dengan metode *Support Vector Machine* (SVM). *Tweet* diberi label secara manual dan dilakukan tahap *pre-processing* dan ekstraksi fitur. Kemudian data dibagi menjadi data *training* dan data *testing* dengan rasio *training*:*testing* = 80:20. Lalu, dilakukan pembangunan model SVM dan dilakukan simulasi numerik untuk mencari *hyperparameter* terbaik dari model SVM. Diperoleh kombinasi *hyperparameter* terbaik model SVM untuk masalah ini adalah kernel = *rbf*, C = 6, γ = 2, dan d = 0,6 dengan performa model SVM menghasilkan akurasi, presisi, dan *recall* berturut-turut sebesar 94,7106%, 100%, 94,07%.

1. PENDAHULUAN

Pandemi COVID-19 telah berada di Indonesia lebih dari 1 tahun. Banyak warga Indonesia yang terjangkit dan meninggal akibat COVID-19 setiap harinya dan cenderung bertambah [10]. Pemerintah Indonesia di tahun 2021, telah memulai pelaksanaan vaksinasi COVID-19 untuk memutuskan mata rantai penularan penyakit COVID-19. Namun kenyataannya, banyak hoaks terkait vaksinasi COVID-19 yang beredar di masyarakat Indonesia. Kemkominfo RI mendeteksi banyak hoaks COVID-19 yang beredar di masyarakat [11]. *Hoaks* yang beredar ini bertujuan untuk membujuk masyarakat agar enggan untuk melakukan vaksinasi COVID-19.

Berdasarkan data Hootsuite, *Twitter* menempati urutan ke-5 sebagai platform media sosial yang paling aktif di Indonesia yakni sebanyak 56% dari jumlah populasi Indonesia [13]. Hal ini yang menjadikan *Twitter* sebagai media sosial untuk menyampaikan opini yang dapat digunakan dalam menganalisis sentimen publik terhadap masalah yang terjadi di masyarakat.

Analisis sentimen merupakan salah satu teknik untuk mengekstrak informasi berupa sikap seseorang terhadap suatu isu atau kejadian dengan mengkasifikasikan sebuah teks bersifat positif, negatif, atau netral. Salah satu metode yang dapat digunakan untuk menyelesaikan masalah klasifikasi adalah SVM (Support Vector Machine).

Data yang diambil dari *Twitter* akan dianalisis dengan metode SVM (*Support Vector Machine*). Novantirani, dkk, dalam analisis sentimen pada *Twitter* mengenai penggunaan transportasi umum darat dalam kota dengan metode *Support Vector Machine* telah melakukan penelitian dengan 1138 data dan mencapai nilai akurasi sebesar 78,12% [1]. Rahmawati, dkk, dalam analisis sentimen publik pada media sosial *Twitter* terhadap pelaksanaan pilkada serentak menggunakan algoritma *Support Vector Machine* telah melakukan penelitian dengan 3000 data dan mencapai nilai akurasi sebesar 98% [7]. Listari, dkk, dalam analisis sentimen *Twitter* terhadap bom bunuh diri di Surabaya 13 Mei 2018 menggunakan pendekatan *Support Vector Machine* telah melakukan penelitian dengan 2042 data dan mencapai nilai akurasi sebesar 100% [9]. Beberapa penelitian yang telah dilakukan oleh peneliti tersebut menunjukkan bahwa metode SVM memiliki tingkat akurasi yang cukup bagus dalam masalah klasifikasi. Dengan dilakukannya Analisis Sentimen Vaksinasi COVID-19 di Indonesia dengan Metode SVM diharapkan dapat membantu untuk mengetahui sentimen masyarakat

Indonesia secara umum terhadap pengadaan vaksinasi COVID-19.

2. TINJAUAN PUSTAKA

2.1 *Twitter*

Twitter merupakan situs web yang menawarkan jaringan sosial berupa *microblog* yang memungkinkan penggunanya mengirim dan membaca pesan blog. Namun pesan yang disampaikan terbatas hanya 140 karakter yang berada pada halaman profil pengguna. Pesan dalam *Twitter* dikenal dengan sebutan *tweet* [1].

Pada dasarnya, *tweet* terdiri dari teks biasa, URL, nama pengguna (*username*), dan tagar. *Hashtag* adalah konvensi untuk mengawali kata atau frasa yang tidak diberi spasi dalam *tweet* dengan simbol "#" untuk menyoroti ide utama atau inti dari *tweet* itu. Semua *tweet* yang memiliki *hashtag* yang sama secara otomatis dikelompokkan bersama oleh *Twitter*. Sehingga, dapat mempermudah kategorisasi dan penelusuran topik khusus [12].

2.2 Analisis Sentimen

Analisis sentimen atau *opinion mining* mulai banyak dilakukan penelitian pada tahun 2013. Analisis sentimen adalah riset komputasional dari opini, sentimen dan emosi yang diekspresikan secara tekstual [2]. Analisis sentimen dilakukan berdasarkan data tekstual yang berguna untuk mengetahui opini publik terkait isu yang sedang hangat dibicarakan [8].

2.3 Vaksinasi COVID-19

Vaksin merupakan zat yang dimasukkan ke dalam tubuh yang berfungsi untuk menghasilkan sistem kekebalan tubuh manusia pada suatu penyakit. Vaksinasi merupakan suatu kegiatan menyuntikan vaksin ke dalam tubuh, yang berguna untuk menstimulasi sistem imun tubuh sehingga dapat memproduksi imunitas terhadap suatu penyakit [3].

COVID-19 merupakan virus jenis baru yang berasal dari golongan *coronavirus* yakni SARS-CoV-2 (virus corona) [4]. Dalam banyak kasus, COVID-19 menimbulkan gejala ringan seperti batuk kering, kelelahan dan demam. Gejala ringan lain yang diderita adalah hidung tersumbat, sakit dan nyeri, pilek, sakit tenggorokan atau diare.

Tetapi terdapat beberapa orang yang terinfeksi COVID-19 tidak menunjukkan gejala apa pun. Mayoritas orang yang sembuh dari COVID-19 tanpa memerlukan perawatan khusus. Sekitar 1 dari setiap 6 orang yang tertular COVID-19 menjadi sakit parah dan kesulitan bernapas [5].

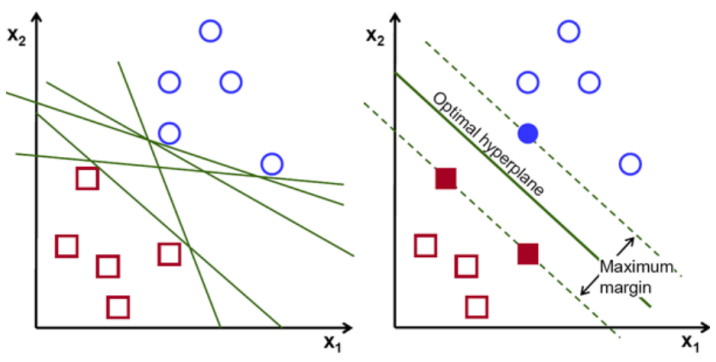
Maka dari itu, upaya intervensi untuk memutus mata rantai penularan penyakit COVID-19 dilakukan dengan Vaksinasi. Vaksinasi COVID-19 bertujuan untuk mengurangi transmisi/penularan, menurunkan angka kesakitan dan kematian akibat COVID-19, dan mencapai kekebalan kelompok di masyarakat (*herd immunity*) serta menjaga masyarakat agar tetap produktif secara sosial dan ekonomi [6].

2.4 *Support Vector Machine* (SVM)

2.4.1 Garis Besar SVM dan Fungsi *Loss* SVM

Support Vector Machine (SVM) adalah suatu teknik untuk melakukan prediksi, baik dalam kasus klasifikasi maupun regresi. Prinsip dasar dalam SVM yakni linear *classifier* yaitu kasus klasifikasi yang secara linier dapat dipisahkan, namun SVM telah dikembangkan agar dapat bekerja pada problem non-linier dengan memasukkan konsep kernel pada ruang kerja berdimensi tinggi. Pada ruang berdimensi tinggi, akan dicari *hyperplane* (*hyperplane*) yang dapat memaksimalkan jarak (margin) antara kelas data [14].

Konsep SVM dapat dijelaskan secara sederhana sebagai usaha mencari hyperplane terbaik yang berfungsi sebagai pemisah dua buah class pada input space yaitu data bersentimen positif (berlabel +1) dengan data bersentimen negatif (berlabel -1) [1]. SVM melibatkan data *training* dan data *testing* dimana masing-masing terdiri dari beberapa *input* dari suatu *dataset*. Masing-masing *input* dalam data *training* berisi satu nilai target dari klasifikasi dan beberapa atribut/parameter. Sehingga, SVM dapat memproduksi suatu model yang dapat memprediksi nilai target dari data *testing* yang hanya diberikan nilai atributnya / parameternya [9].



Gambar 1. SVM mencari hyperplane terbaik.

Support Vector Machine (SVM) menggunakan model linear sebagai *decision boundary* dengan bentuk umum sebagai berikut:

$$f(x) = w^T \Phi(x) + b \ldots (1)$$

dimana: w^T :Parameter bobot

$\Phi(x)$: Fungsi yang memetakan vector input x ke dimensi yang lebih tinggi

b : Bias

$f(x)$ bernilai positif jika x merupakan sentimen positif, dan akan bernilai negatif jika x merupakan sentimen negatif. Nilai $y = 1$, jika sentimen merupakan sentimen positif dan $y = -1$, jika sentimen tersebut merupakan sentimen negatif. Maka, $y(f(x))$ akan bernilai positif jika klasifikasi tepat menurut fungsi prediksi, dan akan bernilai negatif jika klasifikasi tidak tepat menurut fungsi prediksi [15].

Pada klasifikasi biner, fungsi *loss* yang digunakan oleh SVM didefinisikan sebagai berikut:

$$|1 - y(f(x))|_+ = \begin{cases} 0\,, \; untuk\; 1 - y(f(x)) < 0 \\ 1 - y(f(x)), \; untuk\; lainnya \end{cases} \ldots (2)$$

Jika $|1 - y(f(x))|_+ = 0$, maka data diklasifikasikan secara tepat dan pada sisi *decision boundary* yang benar. Jika $0 < |1 - y(f(x))|_+ < 1$, maka data berada di dalam margin dan pada sisi *decision boundary* yang benar. Jika $|1 - y(f(x))|_+ > 1$, maka data berada pada sisi *decision boundary* yang salah [16].

Dengan demikian, fungsi *loss* ini merupakan fungsi loss dengan *soft margin*, sebab memungkinkan beberapa data berada pada sisi *hyperplane* yang salah (tidak bisa diklasifikasikan secara benar) atau memberikan kelunakan untuk beberapa data yang salah dalam pengklasifikasian [14].

Kernel yang umumnya dipakai dalam SVM adalah [15]:

1. Kernel Linear

$$K(x_i, x_j) = \langle x_i, x_j \rangle \ldots (3)$$

2. Kernel Polinomial

$$K(x_i, x_j) = (\gamma \langle x_i, x_j \rangle + r)^d \quad , \gamma > 0 \ldots (4)$$

3. Kernel RBF

$$K(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2} \dots (5)$$

2.4.2 Model Matematika SVM

Misalkan $x_i,\ i = 1,\ 2,\ \dots,\ n$ adalah vector data training pada dua kelas dan vector $y_i = \{-1,\ 1\}^n$. Maka bentuk primal dari masalah optimisasi untuk *soft margin* adalah:

$$\min_{\vec{w},\, b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i \dots (6)$$

$$st.\quad y_i(w^T \Phi(x_i) + b) \geq 1 - \xi_i,\quad i = 1,\ 2,\ \dots,\ n$$

$$\xi_i \geq 0$$

dimana:

ξ_i	:	variabel *slack* data training ke-i
C	:	parameter regularisasi yang mengendalikan *trade off*
Φ	:	fungsi yang memetakan vector ke runag dimensi yang lebih tinggi
x_i	:	vektor data training
y_i	:	vektor label kelas dari x_i yang berdimensi n

Ketika $C \to \infty$, maka *hyperplane* yang optimal akan memisahkan data secara sempurna. Semakin besar nilai C, fungsi klasifikator bersifat hard margin. Semakin kecil nilai C, maka margin error untuk tiap titik datanya akan cenderung membesar. Batasan $y_i(w^T \Phi(x_i) + b) \geq 1 - \xi_i$ menunjukkan bahwa fungsi prediksi diharapkan melakukan kesalahan yang sama atau lebih kecil daripada variable *slack*.

SVM sulit diselesaikan melalui primalnya karena komputasi $w^T \Phi(x)$ sangat sulit. Oleh karena itu, penyelesaian masalah SVM akan diselesaikan dari dual masalahnya. Fungsi objektif pada masalah dual diperoleh dari pengali lagrange dan batasan masalah dual pada persamaan (6), diperoleh dari KKT *condition*. Sehingga diperoleh rumus masalah dual adalah:

$$\max_{a} \sum_{i=1}^{n} a_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{i=1}^{m} a_i a_j y_i y_j K(x_i, x_j) \dots (7)$$

$$st.\ 0 \leq a_i \leq C,\quad i = 1,\ 2,\ \dots,\ n$$

$$\sum_{i=1}^{n} a_i y_i = 0$$

dimana: a : pengali lagrange

$K(x_{i,}x_j)$: *kernel trick* dimana $K(x_{i,}x_j) = \Phi(x_i)\Phi(x_j)$

y_i : vektor berdimensi n dimana $y_i = \{-1,\ 1\}^n$

$x_{i,}$: vector data *training*

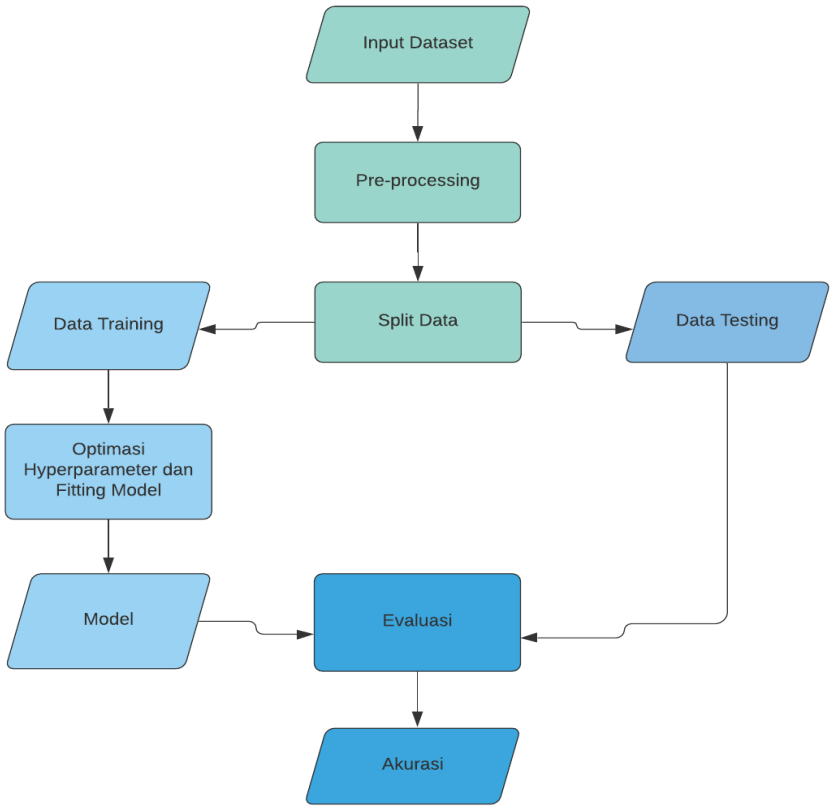
Fungsi keputusan dari masalah dual ini adalah:

$$f(x) = \sum_{i=1,\ x_i \in SV}^{n} a_i y_i K(x, x_i) + b \ldots (8)$$

dimana SV merupakan subset data *training* yang terpilih sebagai *support vector* [16].

3. METODE

3.1 Alur Pembangunan Model



Gambar 2. *Flowchart* pembangunan model

3.2 *Dataset*

3.2.1 Pengambilan *Dataset*

Dataset diambil dari *tweet*s *Twitter* dari kalangan masyarakat di Indonesia dengan melakukan *crawling text*. *Crawling text* dilakukan dengan menggunakan API *Twitter*. Namun karena keterbatasan waktu, akan digunakan *dataset tweet* mengenai vaksinasi COVID-19 yang ada di Indonesia. *Dataset* berisi 8397 *tweet* berbahasa Indonesia dari pengguna *Twitter* mengenai vaksinasi COVID-19 di Indonesia yang berasal dari *github* ShinyQ yang dapat diakses melalui tautan https://github.com/ShinyQ/Analisis-Sentimen-Kebijakan-Vaksinasi-COVID-19-Pemerintah_Naive-Bayes-Classifier/blob/main/*Dataset*_Covid_*Twitter*_Raw.csv. Kemudian akan dihilangkan *tweet* yang memiliki isi yang sama (duplikat) sedemikian sehingga diperoleh 1277 *tweet*.

3.2.2 Pelabelan Data

Dilakukan pelabelan secara manual pada setiap *tweet* yang ada pada *dataset*. *Tweet* akan dibagi menjadi dua kelas sentimen, yaitu sentimen positif dan sentimen negatif, dengan nilai "1" untuk *tweet* yang memiliki sentimen positif dan "-1" untuk *tweet* yang memiliki sentimen negatif. *Dataset* terdiri dari 1200 sentimen positif dan 77 sentimen positif.

3.3 Pre-processing

Dalam *preprocessing dataset* akan dibentuk atau dipersiapkan sesuai dengan kebutuhan klasifikasi dan memudahkan pemrosesan pada sistem.

3.3.1 Pembersihan data

Berikut ini merupakan langkah-langkah dalam proses pembersihan data []:

- Mengubah semua kalimat menjadi huruf kecil pada *tweet*.
- Menghilangkan tautan web pada *tweet*. Contohnya "https://emas.ui.ac.id" akan dihilangkan.

- Menghilangkan username yang ada pada *tweet*. Contoh "@nama" akan dihilangkan.
- Menghilangkan tagar yang ada pada kalimat bertagar pada *tweet*. Contoh "#kata" akan diubah menjadi"kata".
- Menghilangkan tanda baca (contohnya ., ;, :, ", dll) dan angka (contohnya 1, 2, 3, 4, dll) pada *tweet*.
- Memisahkan kata yang disambung oleh tanda baca '-' menjadi dua buah kata terpisah pada *tweet*. Contohnya "buku-buku" akan diubah menjadi "buku" dan "buku".
- Menghapus kata yang hanya terdiri dari satu huruf pada *tweet*.
- Menghapus huruf berulang lebih dari dua yang bersebelahan menjadi hanya dua pada *tweet*. Contohnya "cintaaaaa" akan diubah menjadi "cintaa".
- Mengubah kata-kata yang tidak baku menjadi kata baku pada *tweet*.
- Menghilangkan kata "rt" pada *tweet*.

Contoh pembersihan data *tweet*:

Sebelum	Vaksin Gratis COVID-19, Penolakan Masyarakat Tetap Perlu Diantisipasi https://t.co/GKRSBOK6wX via @HARNAS.ID #BeritaTerkini #beritanasional #vaksinuntukkita #penolakanvaksin #vaksingratis #ingatpesanibu #pakaimasker https://t.co/Z8bMxn9zjP
Sesudah	vaksin gratis covid penolakan masyarakat diantisipasi via beritaterkini beritanasional vaksinuntukkita penolakanvaksin vaksingratis ingatpesanibu pakaimasker

Tabel 1. Contoh *tweet* sebelum dan sesudah dibersihkan

3.3.2 Ekstraksi Fitur

Ekstraksi fitur memiliki tujuan untuk merepresentasikan data secara menyeluruh. Pada penelitian ini digunakan metode *Term Frequency-Inverse Document Frequency* (TF-IDF). Nantinya, kata akan diberikan nilai bobot berdasarkan frekuensi munculnya dalam data, sehingga dapat direpresentasikan dalam bentuk vektor [1].

3.4 Pemisahan data

Pemisahan *dataset* PGK dilakukan dengan memisahkan *dataset* menjadi data training dan data testing. Setiyono dkk, melakukan penelitian terkait klasifikasi SMS spam menggunakan algoritma SVM dimana perbandingan rasio data yang digunakan adalah *training:testing* = 80%:20% dengan tingkat akurasi sebesar 96,72% [17]. Puspitasari dkk, melakukan penelitian terkait klasifikasi penyakit gigi dan mulut menggunakan metode SVM dimana perbandingan rasio data dengan tingkat akurasi terbaik terdapat pada rasio data training:testing 80%:20% dengan nilai akurasi sebesar 93,328% [18].

Berdasarkan kedua penelitian yang pernah dilakukan oleh peneliti sebelumnya, didapatkan rasio data training:testing berupa 80%:20% memberikan tingkat akurasi yang cukup baik. Oleh karena itu, pada penelitian ini *dataset* akan dipisah menjadi 80% data training dan 20% data testing.

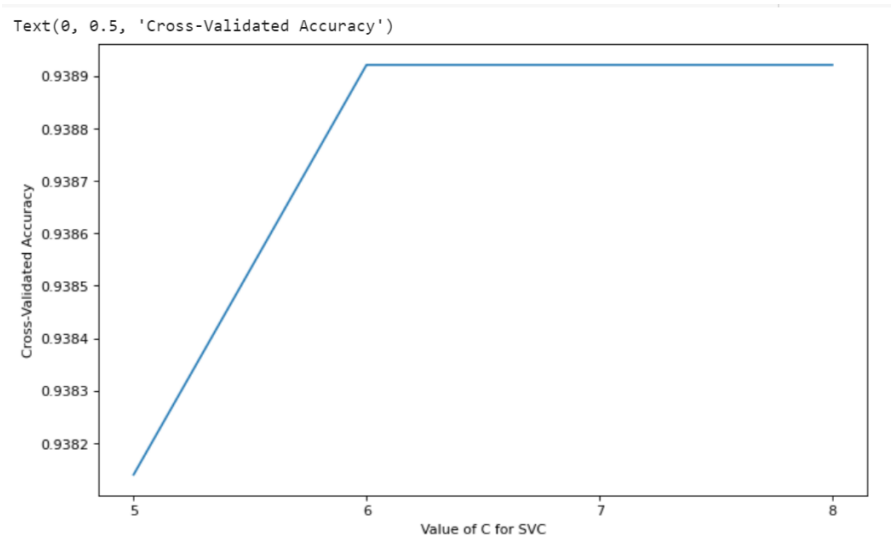
3.5 Penentuan *Hyperparameter* untuk Optimisasi

Selanjutnya, data *training* akan dilatih dengan model SVM. Pelatihan model dilakukan menggunakan perinth *SVC()* yang merupakan singkatan dari *Support Vector Classifier*. Beberapa parameter yang terdapat pada model akan dioptimisasi agar dapat menghasilkan model dengan performa terbaik. Optimisasi dilakukan dengan *Grid Search Cross Calidation* (*GridSearchCV*). Metode *Grid Search* adalah metode dimana akan dicoba semua kombinasi yang mungkin dari calon-calon *hyperparameter*. *Cross validation* yang dikenal juga *k-fold CV* adalah salah satu proses validasi model untuk mengestimasi performa dari sebuah model. Dalam *k-fold cross validation*, data *training* dipartisi/dipotong secara *random* menjadi sejumlah *k*-subhimpunan (*fold*), yaitu $D_1, D_2, \ldots, D_k$, dimana $D_1, D_2, \ldots, D_k$berukuran sama [1]. Dalam penelitian ini *cross validation* dilakukan sebanyak 10-*fold*.

Hyperparameter yang akan di-*tuning* adalah sebagai berikut:

***Hyperparameter* pada model**	
C	Parameter regularisasi yang mengendalikan *trade off*. C merupakan salah satu parameter yang akan dioptimisasi.
γ	Koefisien kernel untuk kernel *rbf*. γ merupakan salah satu parameter yang akan dioptimisasi.
d	Derajat pada fungsi kernel polinomial. d merupakan salah satu parameter yang akan dioptimisasi.
Kernel	Fungsi kernel merupakan fungsi yang memetakan data x ke dimensi yang lebih tinggi. Kernel merupakan salah satu parameter yang akan dioptimisasi.

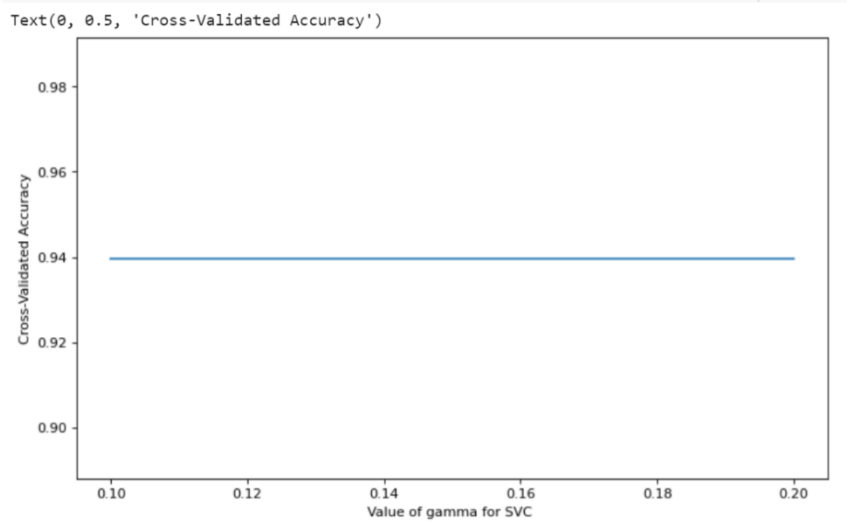
Dalam memilih nilai C yang baik, akan diperiksa dengan prinsip *trial-error* untuk menemukan nilai C yang menghasilkan model SVM dengan akurasi terbaik. Dengan memeriksa nilai C pada interval 5-10 dengan *step size* 0,1 menggunakan kernel linear, diperoleh hasil pada Gambar 3.



Gambar 3. Pengaruh nilai C dalam interval 5-9 terhadap tingkat akurasi model

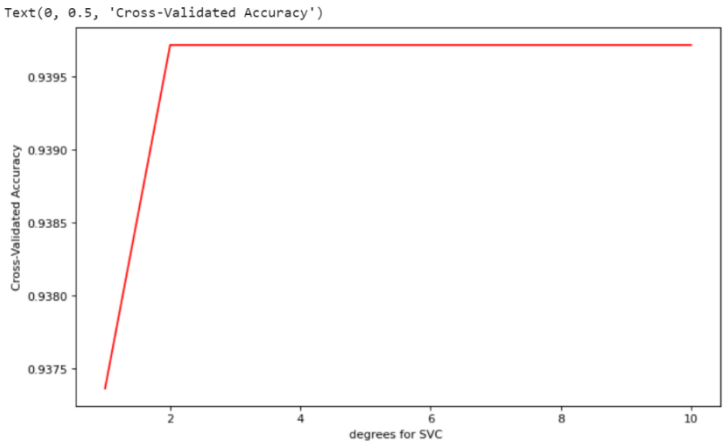
Pada Gambar 3, terlihat bahwa nilai C yang berada pada interval 6-9 menghasilkan tingkat akurasi yang tertinggi. Kemudian, dengan memeriksa nilai γ pada

interval 0,1-5 dengan *step size* 0,5 menggunakan kernel *rbf*, diperoleh hasil pada Gambar 4.



Gambar 4. Pengaruh nilai γ dalam interval 0,1-5 terhadap tingkat akurasi model

Pada Gambar 4, terlihat bahwa nilai γ yang berada pada interval 0,1-5 menghasilkan tingkat akurasi yang tertinggi (karena bersifat konstan). Kemudian, dengan memeriksa nilai d pada interval 1-10 dengan *step size* 1 menggunakan kernel polinomial, diperoleh hasil pada Gambar 5.



Gambar 5. Pengaruh nilai γ dalam interval 1-10 terhadap tingkat akurasi model

Pada Gambar 5, terlihat bahwa nilai d yang berada pada interval 2-10 menghasilkan tingkat akurasi yang tertinggi. Sehingga calon *hyperparameter* yang akan dioptimisasi adalah:

Calon *Hyperparameter* pada model		**Kernel yang digunakan**
C	6 - 9 dengan *step size* 0,5	Linear
γ	0,1 - 5 dengan *step size* 0,1	rbf
d	2, 3, 4, 5, 6, 7, 8, 9, 10	Polinomial

Selanjutnya, dicari kombinasi *hyperparameter* terbaik dari calon-calon *hyperparameter* dengan menggunakan *GridSearchCV* dengan *cross validation* sebanyak 10-*fold* dan nilai akurasi terbaik sebagai penentu model SVM terbaik. Diperoleh kombinasi nilai *hyperparameter* terbaik yang akan digunakan adalah C = 6, d = 2, γ= 0,6, dan kernel = *rbf*. Karena kernel yang digunakan adalah kernel *rbf*, maka *hyperparameter* d tidak akan memberikan pengaruh pada tahap pelatihan model yang sudah teroptimisasi.

3.6 Evaluasi Performa Model

Selanjutnya, dilakukan evaluasi performa untuk menguji hasil dari klasifikasi yang telah dilakukan dengan metode SVM. Evaluasi performa dilakukan dengan mengukur nilai performa dari sistem/model yang telah dibuat. Evaluasi performa yang digunakan adalah akurasi, presisi, *recall* yang perhitungannya diperoleh melalui *confusion matrix* [1].

		True Class	
		Positive	**Negative**
Predicted Class	**Positive**	**True Positive Count (TP)**	**False Positive Count (FP)**
	Negative	**False Negative Count (FN)**	**True Negative Count (TN)**

Tabel 2. Tabel *Confusion Matrix*

Dengan rumus akurasi, presisi, *recall* sebagai berikut:

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \dots (9)$$

$$Presisi = \frac{TP}{TP + FP} \dots (10)$$

$$Recall = \frac{TP}{TP + FP} \dots (11)$$

4. HASIL DAN PEMBAHASAN

4.1 Analisis Performa SVM

Berikut ini adalah visualisasi kata yang sering muncul pada *dataset* dengan *word cloud*. Terlihat bahwa kata yang sering muncul adalah vaksin, vaksinuntukkita, covid, vaksincovid, dll. Semakin besar ukuran katanya, artinya kata tersebut sering muncul di dalam *dataset*.

Gambar 6. *Word Cloud* Vaksinasi COVID-19

Data yang telah melewati *pre-processing* kemudian akan dibagi menjadi 2 yaitu data *training* dan data *testing* dengan 80% data digunakan untuk data *training* untuk membangun model SVM dan 20% data digunakan untuk data *testing*. Performa dari model SVM dapat divalidasi dengan memeriksa akurasi ketika diuji dengan data testing, yaitu data yang tidak pernah menjadi *input* pada saat pembangunan model SVM. Pencarian kombinasi *hyperparameter* terbaik dilakukan dengan menggunakan *GridSearchCV* dengan *cross validation* sebanyak 10-*fold* dan nilai validasi akurasi sebagai penentu model SVM terbaik. Diperoleh model SVM dengan *hyperparameter* yang telah dioptimisasi memiliki akurasi secara keseluruhan sekitar 94,140625%. Untuk melihat lebih detail akurasi dari model SVM yang diperoleh, akan dibentuk *confusion matrix* dari data *testing*.

Predicted Class	**Negatif**	**Positif**
True Class		
Negatif	3	15
Positif	0	238

Tabel 3. *Confusion Matrix*

Pada tabel *confusion matrix* diperoleh 4 nilai yaitu *true negative* (TN) yang berjumlah 3 buah, *false positive* (FP) yang berjumlah 0 buah, *false negative* (FN) yang berjumlah 15 buah, dan *true positive* (TP) yang berjumlah 238 buah.

TN yang berjumlah 3 adalah banyaknya data yang diprediksi merupakan sentimen positif dan sesungguhnya merupakan sentimen positif. FP yang berjumlah 0 adalah banyaknya data yang diprediksi merupakan sentimen positif tetapi sesungguhnya merupakan sentimen negatif. FN yang berjumlah 15 adalah banyaknya data yang diprediksi merupakan sentimen negatif tetapi sesungguhnya merupakan sentimen positif. TN yang berjumlah 238 adalah banyaknya data yang diprediksi merupakan sentimen negatif dan sesungguhnya merupakan sentimen negatif.

Akurasi, presisi, *recall* dapat ditentukan melalui persamaan (9), (10), (11), sedemikian sehingga diperoleh:

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} = \frac{238 + 3}{238 + 3 + 15 + 0} = 0{,}9414$$

$$Presisi = \frac{TP}{TP + FP} = \frac{238}{238 + 0} = 1$$

$$Recall = \frac{TP}{TP + FN} = \frac{238}{238 + 15} = 0{,}9407$$

Karena nilai akurasi, presisi, *recall* tidak berbeda jauh, maka model SVM yang diperoleh memiliki performa yang seimbang dalam melakukan klasifikasi.

4.2 Analisis Overfit Model

Suatu model akan dikatakan tidak *overfit* jika akurasi yang diperoleh data *training* tinggi dan akurasi pada data *validation* cenderung tinggi juga.

Data yang digunakan	**Akurasi**
Data *training*	1
Data *validation*	0,94710642

Tabel 4. Akurasi data *training* dan data *validation*

Berdasarkan perhitungan yang dilakukan, didapatkan tingkat akurasi data *training* sebesar 1 dan tingkat akurasi data *validation* sebesar 0,94710642. Diperoleh perbedaan skor sekitar 5,289358%. Karena perbedaan skor akurasi yang cukup kecil inilah yang menandakan penggunaan model pada data *validation* tidak menunjukkan performa yang buruk. Sedemikian sehingga diperoleh bahwa model tidak *overfit*.

5. PENUTUP

5.1 Kesimpulan

1. Pada penelitian ini telah dilakukan analisis sentimen vaksinasi COVID-19 di Indonesia dengan SVM.
2. Berdasarkan hasil simulasi yang telah dilakukan diperoleh kombinasi *hyperparameter* yaitu kernel = rbf, C = 6, γ=0,6, dan d = 2. Performa model SVM menghasilkan akurasi, presisi, dan *recall* berturut-turut sebesar 94,7106%, 100%, 94,07%.

5.2 Saran

Penelitian lebih lanjut yang dapat dilakukan mengenai masalah analisis sentimen vaksinasi COVID-19 di Indonesia adalah:

1. Menggunakan *dataset* terkini yang dapat dilakukan dengan *scraping data* dengan API *Twitter* dan menambah jumlah *tweet* yang digunakan dalam *dataset*.
2. Menggunakan model klasifikasi lain seperti ANN (*Artificial Neural Network*), *Naive-Bayes*, dll.

DAFTAR PUSTAKA

[1] Novantirani, A., Sabariah, K., & Effendy, V. (2015). Analisis Sentimen pada *Twitter* untuk Mengenai Penggunaan Transportasi Umum Darat Dalam Kota dengan Metode Support Vector Machine. *e-Proceeding of Engineering* : Vol.2, No.1 April 2015, 1177.

<https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/2444>

[2] Yunitasari, Y., Musdholifah, A., & Sari, A. K. (2019). Sarcasm Detection For Sentiment Analysis in Indonesian *Tweets*. *IJCCS (Indonesian J. Comput. Cybern. Syst.)*, vol. 13, no. 1, p. 53, 2019, doi: 10.22146/ijccs.41136. <https://doi.org/10.22146/ijccs.41136>

[3] Satuan Tugas Penanganan COVID-19. (2020, Oktober 19). Apa itu: Vaksin, Vaksinasi, Imunisasi dan Imunitas? *Satuan Tugas Penanganan COVID-19*. Diakses pada 19 Maret 2021, dari <https://covid19.go.id/edukasi/masyarakat-umum/apa-itu-vaksin-vaksinasi-imunisasi-dan-imunitas>

[4] Pane, Merry Dame Cristy. (2021, Maret 12). COVID-19. *ALODOKTER*. Diakses pada 19 Maret 2021, dari <https://www.alodokter.com/covid-19>

[5] WHO. FAQ: What is COVID-19? *WHO*. Diakses pada 19 Maret 2021, dari <https://www.who.int/emergencies/diseases/novel-coronavirus-2019/coronavirus-disease-answers?query=What+is+COVID19%3F&referrerPageUrl=https%3A%2F%2Fwww.who.int%2Femergencies%2Fdiseases%2Fnovel-coronavirus-2019%2Fcoronavirus-disease-answers>

[6] Keputusan Direktur Jenderal Pencegahan Dan Pengendalian Penyakit Nomor HK.02.02/4/1/2021 Tentang Petunjuk Teknis Pelaksanaan Vaksinasi Dalam Rangka Penanggulangan Pandemi Coronavirus Disease 2019 (COVID-19).

[7] Rahmawati, A., Marjuni, A., & Zeniarja, J. (2017). Analisis Sentimen Publik pada Media Sosial *Twitter* Terhadap Pelaksanaan Pilkada Serentak Menggunakan Algoritma Support Vector Machine. *CCIT Journal*, *10*(2), 197-206. <https://doi.org/https://doi.org/10.33050/ccit.v10i2.539>

[8] Aditama, M. I., Pratama, R. I., Wiwaha, K. H. U., & Rakhmawati, N. A. (2020). Analisis Klasifikasi Sentimen Pengguna Media Sosial *Twitter* Terhadap Pengadaan Vaksin COVID-19. *JIEET: Volume 04 Nomor 02, 2020 (Journal Information Engineering and Educational Technology).* <https://journal.unesa.ac.id/index.php/jieet/article/view/11018/pdf>

[9] Listari, Ihsan, M., Paradistia E. R., & Widodo E. (2019). Analisis Sentimen *Twitter* terhadap Bom Bunuh Diri di Surabaya 13 Mei 2018 menggunakan Pendekatan Support Vector Machine. *PRISMA, Prosiding Seminar Nasional Matematika 2, 416-426.* <https://journal.unnes.ac.id/sju/index.php/prisma/>

[10] WHO. WHO Coronavirus (COVID-19) Dashboard - Data Table. *WHO*. Diakses pada 20 Maret 2021, dari <https://covid19.who.int/table>

[11] Kemkominfo. Laporan Isu *Hoaks*. *Kemkominfo*. Diakses pada 20 Maret 2021, dari <https://www.kominfo.go.id/content/all/laporan_isu_hoaks>

[12] Doshi, Z., Nadkarni, S., Ajmera, K., Shah, N. (2017). TweerAnalyzer: *Twitter* Trend Detection and Visualization. International Conference on Computing, Communication, Control and Automation (ICCUBEA), pp. 1-6, doi: 10.1109/ICCUBEA.2017.8463951. <https://remote-lib.ui.ac.id:2147/document/8463951/>

[13] Hootsuite (We are Social): Indonesian Digital Report 2020 – Andi Dwi Riyanto, Dosen, Praktisi, Konsultan, Pembicara: E-bisnis/Digital Marketing/Promotion/Internet marketing, SEO, Technopreneur, Fasilitator Google Gapura Digital yogyakarta. (2020). Diakses 18 Juni 2021 <https://andi.link/hootsuite-we-are-social-indonesian-digital-report-2020/>

[14] Octaviani, P. A., Wilandari, Y., Ispriyanti. D. (2014). Penerapan Metode Klasifikasi *Support Vector Machine* (SVM) pada Data Akreditasi Sekolah Dasar (SD) di Kabupaten Magelang. JURNAL GAUSSIAN, Volume 3, Nomor 4, Tahun 2014, Halaman 811 - 820. <http://ejournal-s1.undip.ac.id/index.php/gaussian>

[15] Nugroho, A. S., Witarto, A. B., & Handoko, D. (2003). Support Vector Machine: Teori dan Aplikasinya dalam Bioinformatika.

[16] Murphy, K. P. (2012). Machine Learning. London: The MIT Press.

[17] Setiyono, A., & Pardede, H. F. (2019). Klasifikasi SMS Spam Menggunakan Support Vector Machine. *Jurnal Pilar Nusa Mandiri*, *15*(2), 275–280. Diakses pada 18 Juni 2021 https://doi.org/10.33480/pilar.v15i2.693

[18] Puspitasari, Ana & Eka Ratnawati, Dian & Wahyu Widodo, Agus. 2018. Klasifikasi Penyakit Gigi Dan Mulut Menggunakan Metode Support Vector Machine. Diakses pada 18 Juni 2021 dari https://www.researchgate.net/publication/324038271_Klasifikasi_Penyakit_Gigi_Dan_Mulut_Menggunakan_Metode_Support_Vector_Machine

Lampiran Source Code Program

Import *library* yang dibutuhkan

```
import numpy as np
import pandas as pd
import re
from sklearn.model_selection import train_test_split, cross_val_score, GridSearchCV
import tensorflow as tf
import keras
from keras.preprocessing import text
from sklearn.svm import SVC
from sklearn import metrics
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
import matplotlib.pyplot as plt
from matplotlib.pyplot import figure
```

Upload *dataset*

```
#upload data
data = pd.read_csv('https://raw.githubusercontent.com/listakurniawati/COVID-19-With-SVM/main/covid19_data.csv')
```

Melakukan drop kolom yang tidak dibutuhkan

```
#drop kolom yang tidak dibutuhkan
data.drop(columns = ['acctdesc', 'location',
                     'following', 'totaltweets',
                     'usercreatedts', 'tweetcreatedts',
                     'retweetcount', 'followers'])
```

Menampilkan total *tweet*s yang akan digunakan

```
#total tweets yang akan digunakan
print("Total tweets: %d" %data.shape[0])
```

Upload *dataset* yang sudah diberi label

```
dt = pd.read_csv('https://raw.githubusercontent.com/listakurniawati/COVID-19-With-SVM/main/FIXtweetcovid19.csv?token=ARCQD7CAWBILPO5FVX3TIWLAXMS52')
dt
```

Menghapus data yang terduplikat

```
df = dt.drop_duplicates(subset=['text'], keep='last', inplace=False)
df.text = df.text.astype(str)
df
```

Me*reset* index dari *dataset*

```
df_ = df.reset_index()
```

```
dfnew = df_.drop(['index'], axis=1)
dfnew
```

Menampilkan banyaknya label yang bernilai 1 dan -1

```
dfnew['label'].value_counts()
```

Membuat diagram batang dari *dataset* sentimen

```
labels = ['Positive','Negative']
Category1 = [1200, 77]
plt.bar(labels, Category1, tick_label=labels, width=0.5, color=['coral', 'c'])
plt.xlabel('Kelas Sentimen')
plt.ylabel('Data')
plt.title('Diagram Bar Data Analisis Sentimen')
```

Mengubah label {-1,1} menjadi {0,1}

```
dfnew['label'] = dfnew['label'].replace(-1,0)
```

Melakukan tahapan pembersihan data

```
#PEMBERSIHAN(CLEANING DATA)
stopwords = pd.read_csv("https://raw.githubusercontent.com/listakurniawati/COVID-19-With-SVM/main/stopwords_id.csv?token=ARCQD7EZ55J4TUTAWYYLYOTAX3FTW")
stopwords = np.append(stopwords, "rt")

def clean_text(tweet):
    # Convert to lower case
    tweet = tweet.lower()
    # Clean www.* or https?://*
    tweet = re.sub('((www\.[^\s]+)|(https?://[^\s]+))','',tweet)
    # Clean @username
    tweet = re.sub('@[^\s]+','',tweet)
    #Remove punctuation
    tweet = re.sub(r'[^\w\s]',' ', tweet)
    #Replace #word with word
    tweet = re.sub(r'#([^\s]+)', r'\1', tweet)
    #Clean number
    tweet = re.sub(r'[\d-]', '', tweet)
    #Remove additional white spaces
    tweet = re.sub('[\s]+', ' ', tweet)
    #trim
    tweet = tweet.strip('\'"')
    # Clean per Words
    words = tweet.split()
    tokens=[]
    for ww in words:
        #split repeated word
        for w in re.split(r'[-/\s]\s*', ww):
            #replace two or more with two occurrences
            pattern = re.compile(r"(.)\1{1,}", re.DOTALL)
            w = pattern.sub(r"\1\1", w)
            #strip punctuation
            w = w.strip('\'"?,.')
            #check if the word cosists of two or more alphabets
            val = re.search(r"^[a-zA-Z][a-zA-Z][a-zA-Z]*$", w)
            #add tokens
            if(w in stopwords or val is None):
                continue
            else:
                tokens.append(w.lower())

    tweet = " ".join(tokens)
    return tweet

```

```
dfnew['text'] = dfnew['text'].map(lambda x: clean_text(x))
dfnew = dfnew[dfnew['text'].apply(lambda x: len(x.split()) >=1)]
dfnew
```

Melakukan ekstraksi fitur

```
from sklearn.feature_extraction.text import TfidfVectorizer
tf = TfidfVectorizer()
text_tf = tf.fit_transform(dfnew['text'])
```

Melakukan pemisahan data dengan rasio 80%:20%

```
x_train, x_test, y_train, y_test = train_test_split(text_tf, dfnew['label'], test_size=0.2, random_state=42)
```

Melakukan analisis *hyperparameter* C dengan kernel linear

```
C_range=list(np.arange(1,10,0.5))
acc_score=[]
for c in C_range:
    svc = SVC(kernel='linear', C=c)
    scores = cross_val_score(svc, text_tf, dfnew['label'], cv=10, scoring='accuracy')
    acc_score.append(scores.mean())

C_values=list(np.arange(1,10,0.5))
figure(num=None, figsize=(10, 6), dpi=80, facecolor='w', edgecolor='k')
plt.plot(C_values,acc_score)
plt.xticks(np.arange(1,10,1))
plt.xlabel('Value of C for SVC')
plt.ylabel('Cross-Validated Accuracy')
```

```
C_range=list(np.arange(5,9,1))
acc_score=[]
for c in C_range:
    svc = SVC(kernel='linear', C=c)
    scores = cross_val_score(svc, text_tf, dfnew['label'], cv=10, scoring='accuracy')
    acc_score.append(scores.mean())

C_values=list(np.arange(5,9,1))
figure(num=None, figsize=(10, 6), dpi=80, facecolor='w', edgecolor='k')
plt.plot(C_values,acc_score)
plt.xticks(np.arange(5,9,1))
plt.xlabel('Value of C for SVC')
plt.ylabel('Cross-Validated Accuracy')
```

Melakukan analisis *hyperparameter* γ dengan kernel *rbf*

```
gamma_range=list(np.arange(0.1,5,0.1))
acc_score=[]
for g in gamma_range:
    svc = SVC(kernel='rbf', gamma=g)
    scores = cross_val_score(svc, text_tf, dfnew['label'], cv=10, scoring='accuracy')
    acc_score.append(scores.mean())

figure(num=None, figsize=(10, 6), dpi=80, facecolor='w', edgecolor='k')
plt.plot(gamma_range,acc_score)
plt.xlabel('Value of gamma for SVC ')
plt.xticks(np.arange(0.1,5,0.5))
plt.ylabel('Cross-Validated Accuracy')
```

```
gamma_range=list(np.arange(0.1,0.3,0.1))
acc_score=[]
for g in gamma_range:
    svc = SVC(kernel='rbf', gamma=g)
    scores = cross_val_score(svc, text_tf, dfnew['label'], cv=10, scoring='accuracy')
    acc_score.append(scores.mean())

figure(num=None, figsize=(10, 6), dpi=80, facecolor='w', edgecolor='k')
plt.plot(gamma_range,acc_score)
plt.xlabel('Value of gamma for SVC ')
plt.ylabel('Cross-Validated Accuracy')
```

Melakukan analisis *hyperparameter d* dengan kernel polinomial

```
degree=[1,2,3,4,5,6,7,8,9,10]
acc_score=[]
for d in degree:
    svc = SVC(kernel='poly', degree=d)
    scores = cross_val_score(svc, text_tf, dfnew['label'], cv=10, scoring='accuracy')
    acc_score.append(scores.mean())

figure(num=None, figsize=(10, 6), dpi=80, facecolor='w', edgecolor='k')
plt.plot(degree,acc_score,color='r')
plt.xlabel('degrees for SVC ')
plt.ylabel('Cross-Validated Accuracy')
```

Melakukan optimisasi *hyperparameter*

```
svm_model= SVC()
tuned_parameters = {'C': np.arange(6,9,1),
                    'kernel': ['linear','rbf','poly'],
                    'gamma': np.arange(0.1,5,0.5),
                    'degree': [2,3,4,5]}
model_svm = GridSearchCV(svm_model, tuned_parameters, cv = 10, scoring = 'accuracy', return_train_score = True)
```

```
model_svm.fit(x_train, y_train)
print("Akurasi terbaik: %f menggunakan %s" % (model_svm.best_score_, model_svm.best_params_))
means = model_svm.cv_results_['mean_test_score']
stds = model_svm.cv_results_['std_test_score']
params = model_svm.cv_results_['params']
for mean, stdev, param in zip(means, stds, params):
    print("Akurasi: %f (%f) dengan: %r" % (mean, stdev, param))
```

Menampilkan akurasi data validasi model SVM

```
print(model_svm.best_score_)
```

Menampilkan kombinasi *hyperparameter* terbaik model SVM

```
print(model_svm.best_params_)
```

Melakukan prediksi pada data *testing*

```
y_pred= model_svm.predict(x_test)
print('Akurasi:',metrics.accuracy_score(y_pred,y_test))
print('Presisi:',metrics.precision_score(y_pred,y_test))
print('Recall:',metrics.recall_score(y_pred,y_test))
```

Menampilkan *confusion matrix* dari model SVM

```
cm_svm = confusion_matrix(y_test, y_pred)
cm_svm

cm_display = ConfusionMatrixDisplay(cm_svm, display_labels= ['negative','positive']).plot()
```

Melakukan analisis *overfit*/tidak *overfit* pada model

```
svm_model_val= SVC()
tuned_parameters_val = {'C': [5.0], 'degree': [2], 'gamma': [0.30000000000000004], 'kernel': ['rbf']}
model_svm_val = GridSearchCV(svm_model_val, tuned_parameters_val, cv=10,scoring='accuracy',return_train_score=True)
```

```
model_svm_val.fit(x_train,y_train)
model_svm_val.cv_results_
```

Menampilkan wordcloud dari *dataset*

```
from wordcloud import  WordCloud, STOPWORDS, ImageColorGenerator

text_ = " ".join(review for review in test)

# Generate the image
wordcloud = WordCloud(stopwords=stopwords, background_color="white", max_words=100, colormap='gist_rainbow_r').generate(text_)

# visualize the image
fig=plt.figure(figsize=(15, 12))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title('Wordcloud Vaksinasi COVID-19')
plt.show()
```