

**ANALISIS SENTIMEN VAKSINASI COVID-19
PADA KOMENTAR *YOUTUBE* DENGAN MENGGUNAKAN
ALGORITMA *NAIVE BAYES CLASSIFIER* (NBC) DAN *SUPPORT
VECTOR MACHINE* (SVM)**

SKRIPSI

Diajukan Sebagai Syarat Melaksanakan Kewajiban Studi Strata Satu (S1)
Program Studi Sistem Informasi



Oleh:

MIFTA NAMIRA

11160930000018

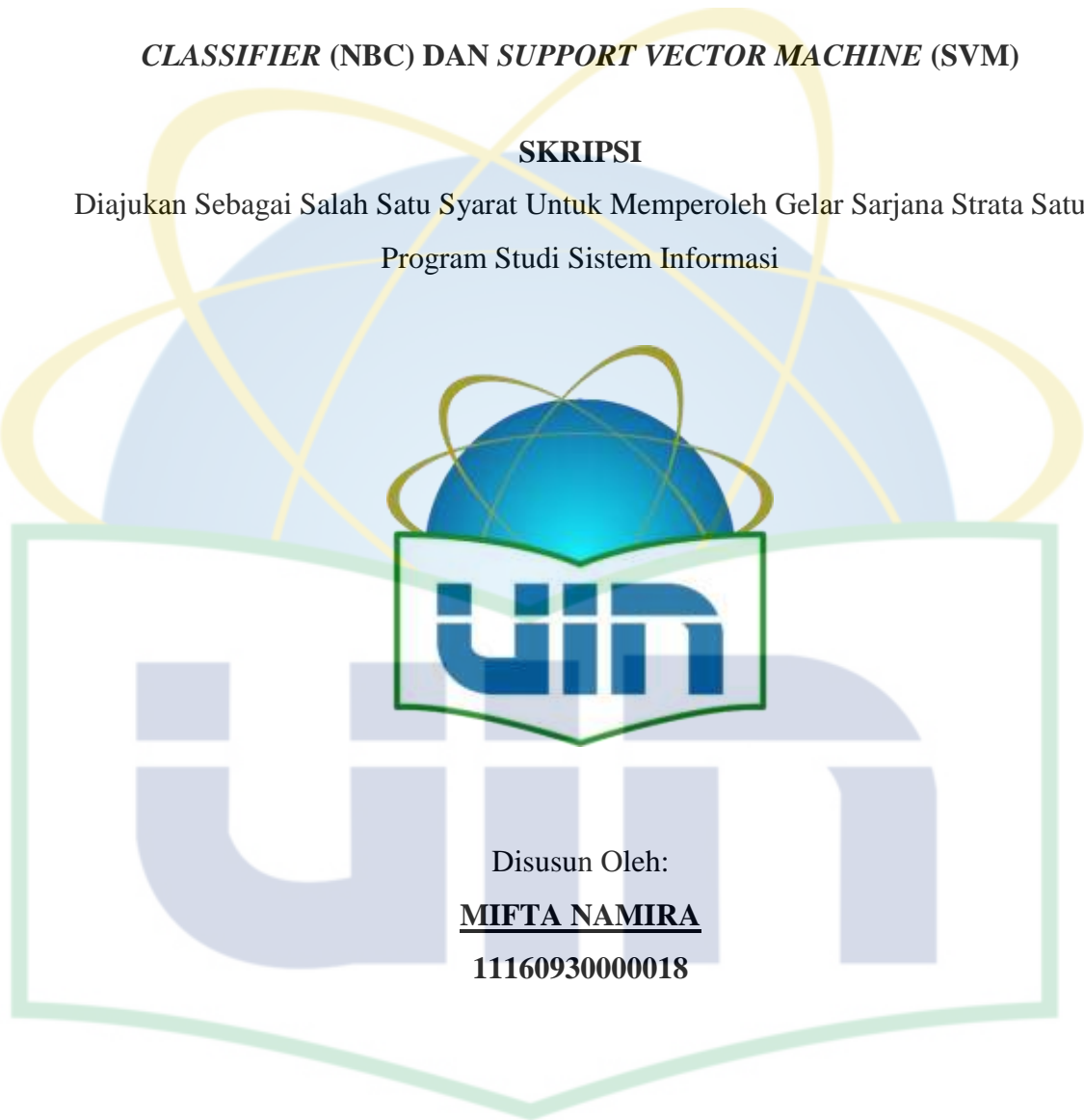
**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SYARIF HIDAYATULLAH
JAKARTA
2023 M / 1444 H**

HALAMAN JUDUL

**ANALISIS SENTIMEN VAKSINASI COVID-19 PADA KOMENTAR
YOUTUBE DENGAN MENGGUNAKAN ALGORITMA *NAIVE BAYES*
CLASSIFIER (NBC) DAN *SUPPORT VECTOR MACHINE* (SVM)**

SKRIPSI

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Strata Satu
Program Studi Sistem Informasi



Disusun Oleh:

MIFTA NAMIRA

11160930000018

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SYARIF HIDAYATULLAH
JAKARTA
2023 M / 1444 H**

LEMBAR PERNYATAAN

DENGAN INI SAYA MENYATAKAN BAHWA SKRIPSI INI BENAR-BENAR HASIL KARYA SENDIRI DAN BELUM PERNAH DIAJUKAN SEBAGAI SKRIPSI ATAU KARYA ILMIAH PADA PERGURUAN TINGGI ATAU LEMBAGA MANAPUN

Jakarta, 29 Mei 2023



Mifta Namira

11160930000018



ABSTRAK

Mifta Namira – 11160930000018, Analisis Sentimen Vaksinasi Covid-19 Pada Komentar *YouTube* dengan Menggunakan Algoritma *Naive Bayes Classifier* (NBC) dan *Support Vector Machine* (SVM) di bawah bimbingan **Dr. Qurrotul Aini, M.T. dan Suci Ratnawati, MTI**.

Media sosial kini seolah merupakan suatu hal yang wajib dimiliki oleh seluruh masyarakat. Salah satu media sosial yang paling banyak digunakan oleh masyarakat Indonesia adalah media sosial *YouTube*, yang mencakup lebih dari 93,8% dari total pengguna media sosial di Indonesia. Tujuan penelitian ini adalah menganalisis sentimen masyarakat terhadap vaksinasi covid-19 menggunakan algoritma *Naive Bayes Classifier* dan *Support Vector Machine* serta mengukur dan membandingkan kinerja kedua algoritma tersebut. Metode penelitian ini menggunakan metode SEMMA (*Sample, Explore, Modify, Model, Assess*). Pengumpulan dataset berupa komentar di-*crawling* dari *YouTube* pada tahap *sample*. Selanjutnya dilakukan eksplorasi atribut dataset pada tahap *explore*. Tahap *modify* merupakan *preprocessing* agar dataset lebih terstruktur. Setelah itu tahap model dengan menerapkan *lexicon based* untuk pemberian kelas sentimen pada dataset. Dataset yang telah memiliki label, selanjutnya diklasifikasikan menggunakan *Naive Bayes Classifier* dan *Support Vector Machine*. Pada tahap akhir adalah *assess* dari metode yang diterapkan menggunakan *confusion matrix* dan *10-fold Cross Validation*. Hasil penelitian ini adalah sebanyak 1327 data masuk ke dalam sentimen positif dan 835 data masuk dalam sentimen negatif. Akurasi metode *Naive Bayes* dengan rasio 80:20 sebesar 92%, *precision* sebesar 93% dan *recall* sebesar 95%. Untuk metode *Support Vector Machine* hasil parameter terbaik pada kernel *linear* dengan $C=1$ dan rasio 70:30 menghasilkan akurasi sebesar 94%, *precision* sebesar 93% dan *recall* sebesar 97%. Penggunaan jumlah rasio data *testing* yang lebih besar, berpengaruh pada tingginya nilai akurasi algoritma *Support Vector Machine* dibandingkan dengan algoritma *Naive Bayes Classifier* yang menghasilkan nilai akurasi tinggi dengan penggunaan rasio data *testing* sedikit.

Kata Kunci: *Lexicon Based, Naive Bayes Classifier, Support Vector Machine, Grid Search Validation, Confusion Matrix, YouTube, Vaksinasi Covid-19.*

V BAB + xx Halaman + 92 Halaman + 22 Gambar + 27 Tabel + Daftar Pustaka

Pustaka Acuan (50, 2004-2022)

KATA PENGANTAR

Assalamu'alaikum Warahmatullaahi Wabarakaatuh

Puji syukur peneliti ucapkan kepada Allah SWT karena atas nikmat dan karunia-Nya peneliti dapat menyelesaikan skripsi ini sebagai syarat untuk mencapai gelar Sarjana Komputer Program Studi Sistem Informasi Fakultas Sains dan Teknologi Universitas Islam Negeri Syarif Hidayatullah Jakarta. Shalawat serta salam kepada Nabi Muhammad SAW beserta keluarga, sahabat serta para pengikutnya hingga akhir zaman.

Selama penyusunan skripsi ini, penulis banyak mendapatkan bantuan, saran, dorongan, dan bimbingan dari berbagai pihak yang merupakan salah satu bentuk pengalaman yang tidak dapat diukur dengan materi. Pada kesempatan ini, penulis ingin mengucapkan terima kasih kepada pihak-pihak yang telah mendukung. Sebagai bentuk apresiasi yang sebesar-besarnya, dengan segala hormat dan kerendahan hati, perkenankanlah penulis mengucapkan terima kasih kepada:

1. Bapak Husni Teja Sukmana, Ph.D. selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Syarif Hidayatullah.
2. Ibu Dr. Qurotul Aini, MT selaku Ketua Program Studi Sistem Informasi dan Bapak Ir. Eri Rustamaji, MBA selaku Sekretaris Program Studi Sistem Informasi Fakultas Sains dan Teknologi Universitas Islam Negeri Syarif Hidayatullah Jakarta.

3. Ibu Dr. Qurrotul Aini, MT. selaku dosen pembimbing I dan Ibu Suci Ratnawati, MTI. selaku dosen pembimbing II yang secara kooperatif telah meluangkan waktu dan memberikan bimbingan, bantuan, semangat, dan motivasi kepada peneliti untuk dapat menyelesaikan skripsi ini.
4. Seluruh Dosen dan Staf Program Studi Sistem Informasi yang telah memberikan banyak tambahan ilmu serta bantuannya selama perkuliahan.
5. Kedua orang tua, Bapak, Ibu, Kakak serta Adik yang selalu memberikan dukungan moral, doa, semangat yang terus mengalir kepada penulis hingga skripsi ini dapat terselesaikan.
6. Zahra Ulinuha selaku sahabat peneliti, yang selalu berbagi suka dan duka bersama, serta selalu membantu peneliti dalam menyelesaikan skripsi ini.
7. Evi, Nia dan Pandu selaku rekan seperjuangan dalam menyelesaikan skripsi ini yang selalu memberikan semangat, saran dan bantuannya.
8. Hana, Firna, Viranda, Noni, Novia, Nurul, Devika dan Agust selaku sahabat peneliti yang selalu memberikan dukungan, semangat, motivasi dan menghibur peneliti dalam menyelesaikan skripsi.
9. Seluruh teman peneliti prodi Sistem Informasi Angkatan 2016, khususnya kelas A yang saling memberikan semangat, dukungan dan saran dalam menyelesaikan skripsi.

Peneliti menyadari bahwa masih terdapat banyak kekurangan yang disebabkan keterbatasan pengetahuan yang penulis miliki sehingga masih jauh dari sempurna. Oleh karena itu penulis memohon maaf atas segala kekurangan tersebut dan tidak menutup diri terhadap segala bentuk saran dan kritik yang dapat membuat penulis menjadi lebih baik dari sebelumnya. Penulis berharap dengan skripsi yang telah penulis kerjakan ini dapat bermanfaat bagi semua pihak. Aamiin.

Jakarta, Mei 2023

Mifta Namira

11160930000018



DAFTAR ISI

HALAMAN JUDUL	i
ABSTRAK	v
KATA PENGANTAR.....	vi
DAFTAR ISI.....	ix
DAFTAR TABEL	xii
DAFTAR GAMBAR.....	xiii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Identifikasi Masalah	9
1.3 Rumusan Masalah.....	9
1.4 Batasan Masalah	10
1.5 Tujuan Penelitian	10
1.5 Manfaat Penelitian.....	11
1.7 Metodologi Penelitian.....	11
1.8 Sistematika Penulisan	12
BAB 2 TINJAUAN PUSTAKA.....	14
2.1 Analisis Sentimen	14
2.2 Naive Bayes Classifier.....	15
2.2.1 Karakteristik <i>Naive Bayes</i>	16
2.3 <i>Support Vector Machine</i>	16
2.3 <i>Text Mining</i>	20
2.4 <i>Machine Learning</i>	20
2.5 Metode SEMMA.....	22
2.6 <i>Text Preprocessing</i>	24
2.6.1 Pengertian <i>Text Preprocessing</i>	24
2.6.2 Tahap <i>Preprocessing Data</i>	25
2.7 <i>Lexicon Based</i>	28
2.8 Kualitas Hasil Klasifikasi	28
2.9 <i>Confusion Matrix</i>	29
2.10 <i>K-Fold Cross Validation</i>	30
2.11 <i>YouTube</i>	31

2.11	Bahasa Pemrograman <i>Python</i>	32
2.12	<i>Google Colaboratory</i>	33
2.13	<i>Library</i>	33
2.13.1	<i>Selenium</i>	33
2.13.2	<i>Pandas</i>	33
2.13.3	<i>Numpy</i>	34
2.13.4	Sastrawi.....	34
2.14	Penelitian Sejenis.....	35
2.15	Perumusan Hipotesis Komparatif.....	44
2.16	Ranah Penelitian.....	45
2.17	Menyampaikan Informasi dengan Benar.....	46
BAB 3	METODE PENELITIAN.....	49
3.1	Metodologi Pengumpulan Data.....	49
3.2	Material.....	50
3.3	Metode SEMMA.....	50
3.4	Tahapan Metode SEMMA.....	56
3.5	Perangkat Penelitian.....	57
3.6	Alur Penelitian.....	58
BAB 4	HASIL DAN PEMBAHASAN.....	59
4.1	<i>Sample</i>	59
4.1.1	Tinjauan Pustaka.....	59
4.1.2	<i>Crawling Data</i>	59
4.2	<i>Explore</i>	62
4.3	<i>Modify</i>	63
4.3.1	<i>Case Folding</i>	63
4.3.2	<i>Cleaning</i>	64
4.3.3	<i>Tokenize</i>	65
4.3.4	<i>Normalize</i>	66
4.3.5	<i>Stopword Removal</i>	68
4.3.6	<i>Stemming</i>	69
4.4	<i>Model</i>	71
4.4.1	<i>Lexicon Based</i>	71
4.4.2	<i>Naive Bayes</i>	75

4.4.3	<i>Support Vector Machine</i>	76
4.5	<i>Assess</i>	77
4.5.1	<i>Naive Bayes</i>	77
4.5.2	<i>Support Vector Machine</i>	80
4.6	Interpretasi Hasil.....	85
4.6.1	Analisa dengan Peneliti Terdahulu	87
BAB 5	90
PENUTUP	90
5.1	Kesimpulan.....	90
5.2	Saran	91
DAFTAR PUSTAKA	93



DAFTAR TABEL

Tabel 2.1 Confusion Matrix	29
Tabel 2.2 Penelitian Sejenis Algoritma Naive Bayes Classifier dan Support Vector Machine	36
Tabel 3.1 <i>Software</i> dan <i>Hardware</i>	57
Tabel 4.1 Contoh URL masing-masing konten video	60
Tabel 4.2 Contoh Hasil Pengambilan Data	61
Tabel 4.3 Hasil Filter kolom pada dataset	62
Tabel 4.4 Hasil <i>Case Folding</i>	64
Tabel 4.5 Tabel hasil <i>cleaning</i>	65
Tabel 4.6 Komentar Sebelum Dan Sesudah Dilakukan Proses <i>Tokenize</i>	66
Tabel 4.7 Komentar Sebelum Dan Sesudah Dilakukan Proses Normalisasi	68
Tabel 4.8 Komentar Sebelum Dan Sesudah Dilakukan Proses <i>Stopword Removal</i>	69
Tabel 4.9 Komentar Sebelum dan Sesudah Dilakukan Proses <i>Stemming</i>	71
Tabel 4.10 Hasil Sentimen Komentar dengan Metode <i>Lexicon</i>	73
Tabel 4.11 Perbandingan Data Train dan Data Test Metode <i>Naive Bayes</i>	76
Tabel 4.12 Perbandingan Data Train dan Data Test Metode <i>Naive Bayes</i>	76
Tabel 4.13 Kernel dan Parameter yang diujikan	77
Tabel 4.14 Hasil Accuracy, Precision dan Recall	79
Tabel 4.15 Hasil <i>Confusion Matrix</i>	79
Tabel 4.16 Hasil <i>10-Folds Cross Validation</i>	80
Tabel 4.17 Hasil <i>Grid Search Cross Validation</i> Tiap Kernel	82
Tabel 4.18 Hasil <i>Grid Search Cross Validation</i> pada Kernel Linear	82
Tabel 4.19 <i>Accuracy, Precision, Recall</i> Kernel Linear	83
Tabel 4.20 Hasil <i>Confusion Matrix</i>	83
Tabel 4.21 Hasil <i>Cross Validation</i>	85
Tabel 4.22 Komentar Positif dan Negatif	85
Tabel 4.23 Kinerja <i>Naive Bayes</i> dan <i>Support Vector Machine</i>	87
Tabel 4.24 Hasil Perbandingan Kinerja	88

DAFTAR GAMBAR

Gambar 1.1 Data Presentase Media Sosial di Indonesia (we are social, 2021).....	3
Gambar 2.1 SVM yang Memisahkan Dua Data dengan Hyperplane.....	17
Gambar 2.2 Dua <i>Hyperplane</i> di Satu Data	18
Gambar 2.3 Dua <i>Hyperplane</i> di Satu Data	19
Gambar 2.4 Proses SEMMA (Alizah et al., 2020)	22
Gambar 2.5 <i>K-Folds Cross Validation</i> (Sanjay, 2018)	31
Gambar 2.6 Ranah Penelitian	46
Gambar 3.1 Alur Pengambilan Data di YouTube (Nuri, 2022)	51
Gambar 3.2 Alur Tahap <i>Modify</i> (Nuri, 2022).....	52
Gambar 3.3 <i>Flowchart Lexicon Based</i> (Nuri, 2022)	53
Gambar 3.4 <i>Flowchart Naive Bayes</i> (Nuri, 2022).....	54
Gambar 3.5 <i>Flowchart Support Vector Machine</i> (Nuri, 2022)	55
Gambar 3.6 Tahapan Metode SEMMA (Afifi, 2022)	57
Gambar 3.7 Alur Penelitian	58
Gambar 4.1 <i>Playlist</i> Konten Video Berdasarkan Bulan Publikasi	60
Gambar 4.2 Data Komentar Berdasarkan Waktu	63
Gambar 4.3 Contoh Normalisasi Kata.....	67
Gambar 4.4 Kamus <i>Lexicon</i>	72
Gambar 4.5 Frekuensi Kata Kelas Positif pada Komentar	74
Gambar 4.6 <i>Wordcloud</i> Kelas Positif pada Komentar	74
Gambar 4.7 Frekuensi Kata Kelas Negatif pada Komentar	75
Gambar 4.8 <i>Worldcloud</i> Kelas Negatif pada Komentar	75

BAB 1

PENDAHULUAN

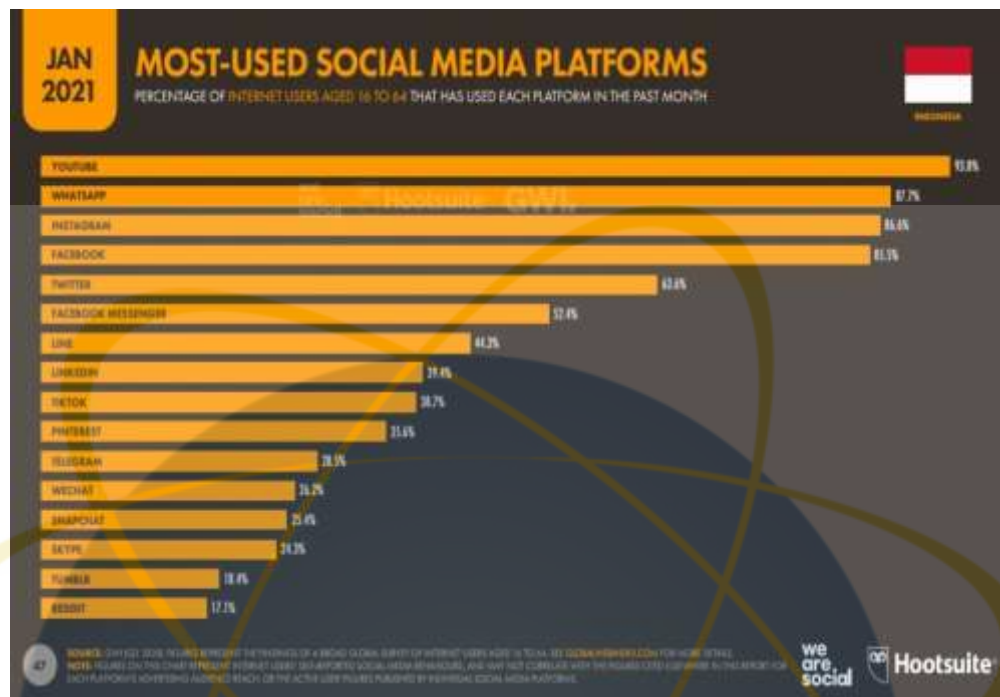
1.1 Latar Belakang

Wabah penyakit baru yang disebabkan oleh virus corona (2019-nCoV) atau yang biasa disebut dengan Covid-19 ditetapkan secara resmi sebagai pandemi global oleh *World Health Organization* (WHO) pada tanggal 11 Maret 2020. Meskipun pusat penyebaran virus tersebut pada akhir tahun 2019 berada di kota Wuhan, China, virus tersebut juga telah tersebar menjangkit ke seluruh masyarakat dunia. Pada umumnya virus corona menyebabkan gejala ringan atau sedang, seperti demam dan batuk, dan kebanyakan bisa sembuh dalam beberapa minggu. Tetapi bagi sebagian orang yang berisiko tinggi (kelompok lanjut usia dan orang dengan masalah kesehatan menahun), virus corona dapat menyebabkan masalah kesehatan yang serius. Efek lanjutan dari Covid-19 ini berpotensi membawa tantangan besar bagi sistem kesehatan dunia dan memiliki konsekuensi yang luas pada ekonomi global jika penyebaran virus tidak dikendalikan secara efektif.

Melihat pesatnya penyebaran Covid-19 dan bahaya yang akan muncul jika tidak segera ditangani, salah satu cara yang sangat mungkin untuk mencegah penyebaran virus ini adalah dengan mengembangkan vaksin. Vaksin tidak hanya melindungi mereka yang divaksinasi tetapi juga masyarakat luas dengan mengurangi penyebaran penyakit dalam populasi. Selain itu, karena virus menyebar dengan sangat cepat maka diperlukan vaksin yang dapat diterapkan dalam waktu singkat untuk meminimalisir dampaknya.

Menyikapi hal tersebut, Pemerintah Indonesia turut aktif dalam rencana kegiatan vaksinasi yang akan diberikan kepada masyarakatnya. Presiden Joko Widodo pada tanggal 5 Oktober 2020 tahun lalu, telah meresmikan Peraturan Presiden (Perpres) Republik Indonesia Nomor 99 Tahun 2020 Tentang Pengadaan Vaksin dan Pelaksanaan Vaksinasi dalam Rangka Penanggulangan Pandemi Covid-19 untuk mengatur kewenangan pemerintah, kementerian/lembaga dan para pejabatnya dalam rencana kegiatan vaksinasi. Kegiatan pengadaan vaksin tersebut juga harus mempertimbangkan berbagai masukan, salah satunya dengan melihat respon dan opini dari masyarakat terhadap vaksinasi tersebut. Sedangkan, program Vaksinasi Covid-19 di Indonesia mulai dilakukan oleh pemerintah pada 13 Januari 2021 di Istana Negara. Orang yang pertama kali disuntik vaksin buatan *sinovac* adalah presiden Joko Widodo (Kementerian Kesehatan, 2021).

Salah satu media yang banyak digunakan oleh masyarakat untuk memberikan pendapatnya terhadap sesuatu adalah media sosial. Media sosial kini seolah merupakan suatu hal yang wajib dimiliki oleh seluruh masyarakat. Berdasarkan data dari “*Hootsuite (We are Social): Indonesian Digital Report 2021*” di *We Are Social* Indonesia, pada tahun 2021 jumlah pengguna media sosial di Indonesia yaitu 170 juta (61,8% dari jumlah populasi di Indonesia). Salah satu media sosial yang paling banyak digunakan oleh masyarakat Indonesia adalah media sosial *YouTube*, yang mencakup lebih dari 93,8% dari total pengguna media sosial di Indonesia pada Gambar 1.1.



Gambar 1.1 Data Presentase Media Sosial di Indonesia (*we are social*, 2021)

YouTube merupakan salah satu saluran media yang menjadi tujuan orang saat mencari informasi tentang Covid-19. Bahkan video yang mencakup Covid-19 tersedia dalam konteks nasional tertentu kemungkinan besar memiliki jangkauan global (Basch *et al.*, 2020). Kelebihan dari *platform YouTube* dibandingkan dengan media sosial lainnya, terletak pada komunikasi audio dan visual yang diberikan, membuatnya mudah diakses oleh individu dari semua kalangan. *YouTube* juga sebagai alat pendidikan yang kuat yang dapat dimobilisasi oleh para profesional kesehatan dalam menyebarkan informasi dan mempengaruhi perilaku publik (Li *et al.*, 2020). Dalam konteks krisis kesehatan masyarakat yang besar seperti wabah Covid-19, penting untuk memahami jenis dan kualitas konten yang digunakan pengguna dalam mencari informasi terkait Covid-19. Alternatif konten di *YouTube* sangat beragam, salah satunya berita terkini (Putri, 2021). Ada

beragam kanal berita yang memberikan informasi kepada publik terkait perkembangan Vaksinasi Covid-19 di Indonesia, di antaranya ada *channel* Kompas TV dan tvOneNews. Kedua *channel* tersebut memiliki jumlah *subscriber* yang banyak yaitu 14,9 juta pada Kompas TV dan 10,7 juta dimiliki oleh tvOneNews.

Pemanfaatan data yang bersumber dari media sosial merupakan suatu terobosan baru yang dapat dijadikan sebagai alternatif sumber data pengganti survei tradisional. Pengumpulan data melalui media sosial dinilai dapat memberikan efisiensi dalam segala hal apabila dibandingkan dengan harus melakukan survei tradisional (Aziz, 2021). Efisiensi tersebut mencakup biaya yang harus dikeluarkan untuk pemerolehan data yang minimal, dapat memperoleh data secara *real time*, dan menghasilkan data yang mempunyai informasi yang lebih detail untuk menggambarkan opini masyarakat yang sebenarnya.

Perlu dilakukan pengkajian terhadap opini masyarakat menggunakan pemrosesan teks karena bentuk data tanggapan belum terstruktur, masih terdapat tanda baca, terdapat bahasa yang tidak baku, bahkan menggunakan *emoticon* dalam opininya. Analisis Sentimen atau *Opinion Mining* digunakan untuk teknik untuk menganalisis opini, sentimen, penilaian dan emosi terhadap suatu entitas seperti produk, jasa, kejadian atau atribut lainnya. Ada beberapa jenis Analisis Sentimen, yaitu *Intent Sentiment Analysis* yang bertujuan untuk mengidentifikasi dan menggali motivasi di balik pesan pengguna, apakah itu termasuk keluhan, saran atau pendapat. Tipe berikutnya adalah *Aspect-Best Sentiment Analysis* yang berfokus pada elemen-elemen yang lebih spesifik dari produk dan layanan.

Kemudian tipe berikutnya *Fine-Grained Sentiment Analysis* yang merupakan salah satu jenis yang paling umum (Annisa, 2022). Fokusnya ada pada tingkat polaritas pendapat. Pemikiran dasar dari teknik analisis sentimen ini adalah untuk mengelompokkan teks, kalimat, atau dokumen tersebut termasuk kedalam sentimen atau opini yang positif, negatif, atau netral (Sanjaya & Lhaksana, 2020), sehingga hasil klasifikasi yang didapatkan dapat membantu untuk mengetahui tanggapan atau kekhawatiran masyarakat terhadap vaksin Covid-19.

Analisis data dari media sosial dapat membantu bisnis, pemerintah, organisasi keamanan dan lingkungan untuk mengetahui masalah, saran dan kritik pada masyarakat. Analisis sentimen merupakan bidang penelitian yang muncul dari *Natural Language Processing* (NLP) untuk mengekstrak pendapat, pemikiran yang dilihat oleh orang (Sigmawaty & Adriani, 2019). Analisis sentimen atau *opinion mining* bisa dianggap sebagai kombinasi antara *Text Mining* dan *Natural Language Processing*. Salah satu metode dari *text mining* yang bisa digunakan untuk menyelesaikan masalah *opinion mining* adalah *Naive Bayes Classifier* (NBC). NBC bisa digunakan untuk mengklasifikasikan opini ke dalam opini positif dan negatif. NBC bisa berfungsi dengan baik sebagai metode klasifikasi teks yang dilakukan oleh Kamruzzaman & Rahman (2004) dan Kibriya *et al.*, (2004). Dari hasil pengujian disebutkan bahwa teks bisa diklasifikasikan dengan akurasi yang tinggi.

Metode lainnya yang digunakan untuk analisis sentimen atau *opinion mining* yaitu *Support Vector Machine* (SVM) yang dikembangkan oleh Vapnik pada tahun 1992, mampu memberikan kinerja lebih tinggi untuk akurasi

klasifikasi. SVM telah banyak dilakukan untuk klasifikasi teks dan data. Pada penelitian Srivastava dan Bhambhu (2005), dilakukan komparasi metode SVM dengan percobaan beberapa kernel dan metode lainnya. Dari penelitian itu menunjukkan bahwa SVM mendapatkan akurasi terbaik dengan fungsi kernel dan parameter tertentu dibanding metode lainnya.

Penelitian ini menggunakan algoritma *Naive Bayes Classifier* (NBC) dan *Support Vector Machine* (SVM) untuk melakukan analisis sentimen melalui komentar pada video berita yang berkaitan dengan vaksinasi Covid-19 yang diposting pada kanal *YouTube* berita seperti TVOneNews dan KOMPASTV. Berita mengenai vaksinasi covid-19 tersedia dalam jumlah yang banyak pada beberapa media sosial. Dengan ketersediaan berita mengenai vaksinasi covid-19 yang banyak tersebut, dapat memberikan sampel data yang banyak untuk penelitian ini.

Pada penelitian Putra (2021) mengenai analisis sentimen pada media sosial *YouTube* mengenai serikat pekerja Pertamina, dengan menggunakan metode *Naive Bayes Classifier* (NBC), menghasilkan tiga *class* sentimen yaitu: positif, negatif, dan netral dari data awal yang telah melalui *preprocessing* data. Penelitian yang dibuat, berhasil digunakan dalam menganalisis komentar orang pada media sosial *YouTube* dengan menggunakan 300 data *testing* dan 1000 data *training*. Pada hasil penelitian, mendapatkan tingkat akurasi sebanyak 98,00% pada ketiga sentimen negatif, positif dan netral. Pada kelas presisi, prediksi positif memiliki nilai kelas presisi sebesar 80,00% dan prediksi negatif memiliki nilai sebesar 66,67%.

Pada penelitian Yulita (2021) mengenai analisis sentimen masyarakat tentang vaksinasi Covid19 pada *Twitter* menggunakan metode *Naive Bayes Classifier* (NBC). Penggunaan teorema *Bayes* pada algoritma *Naive Bayes* ini dengan menggabungkan *prior probability* dan *conditional probability* dalam suatu rumus yang dapat digunakan untuk menghitung probabilitas dari setiap kemungkinan klasifikasi. Dalam menentukan kualitas proses dalam menentukan nilai akurasi diuji dengan parameter akurasi. Variabel seperti TP (*True Positif*), TN (*True Negative*), FP (*False Positif*), dan FN (*False Negative*) berasal dari *confusion matrix*. Hasil dari penelitian ini, rata-rata memberikan respon positif dengan persentase sebesar 60,3% dan jumlah data sebanyak 2278 data. Sedangkan respon negatif lebih kecil dibandingkan respon netralnya, dengan persentase sebesar 5,4% (203 data), dan respon netral adalah 34,4% (1299 data). Nilai akurasi yang dihasilkan sebesar 0,93 (93%).

Pada penelitian Zalyhaty (2021), metode yang digunakan dalam Analisis Sentimen yaitu dengan *Support Vector Machine* (SVM). Algoritma SVM merupakan salah satu algoritma *machine learning* yang dikenal cukup baik dalam melakukan klasifikasi berdasarkan pembobotan yang diproses dalam algoritma tersebut. Pada hasil penelitian ini, tahapan yang dilakukan mulai dari pengumpulan data dari media berita online, pelabelan data secara manual, kemudian tahapan *preprocessing*, pembobotan TF-IDF, pembuatan model klasifikasi, pengujian model klasifikasi, validasi sampai evaluasi. Dari total data 283 data tanggapan masyarakat mengenai vaksin Covid-19 dengan perbandingan 90:10, data *training* sejumlah 254 dan data *testing* sejumlah 29 menghasilkan

presentase 66,8% untuk sentimen positif dan 33,2% untuk sentimen negatif, rata-rata *cross validation score* senilai 72,44% skor akurasi senilai 82,76% presisi senilai 78,26% dan *recall* 100%.

Pada penelitian Srivastava & Bhambhu (2010), dilakukan pengklasifikasian data kesehatan menggunakan SVM, menyatakan kinerja terbaik dari SVM adalah pada penggunaan data testing dengan rasio tertinggi yang mendapatkan akurasi terbaik. Kemudian pada penelitian Tuhuteru & Iriani (2018), dilakukan analisis sentimen dengan membandingkan metode SVM dan NBC. Data *testing* yang digunakan dalam penelitian ini lebih banyak dibandingkan data *training*. Hasil akurasi pada penelitian ini menyatakan bahwa SVM lebih tinggi nilai akurasinya dibanding NBC karena menggunakan data *testing* yang besar. Pada penelitian Anjasmos et al. (2020), menghasilkan informasi tentang perbandingan dari metode SVM dan NBC berupa skor *accuracy*, *precision* dan *recall*. Skor *accuracy*, *precision* dan *recall* tertinggi dengan data *testing* sebanyak 50% dan data *training* 50% adalah 0,897. Sedangkan untuk pembagian data *testing* sebanyak 40% dan data *training* 60% adalah 0,486 menggunakan metode SVM.

Berdasarkan uraian yang telah dijabarkan, peneliti tertarik untuk melakukan analisis sentimen pengguna *YouTube* terhadap topik vaksinasi Covid-19 dengan Metode *Naive Bayes Classifier* dan *Support Vector Machine* untuk mengukur dan membandingkan tingkat akurasi dengan bahasa pemrograman *Python*. Dataset dalam penelitian ini adalah data komentar berbahasa Indonesia, maka dari itu peneliti menggunakan *tools Google Colaboratory* dengan bahasa

pemrograman *Python* untuk mengolah data. Peneliti mengangkat tema tersebut dalam bentuk skripsi dengan judul “**Analisis Sentimen Vaksinasi Covid-19 pada Komentar Youtube dengan Menggunakan Algoritma Naive Bayes Classifier (NBC) dan Support Vector Machine (SVM)**”.

1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah diuraikan, maka dapat diidentifikasi masalah dalam penelitian ini adalah:

- a. Opini dan berita mengenai Vaksinasi Covid-19 yang sedang berkembang di masyarakat melalui media sosial, salah satunya pada *platform YouTube*.
- b. Banyak penelitian mengenai analisis sentimen yang menggunakan algoritma *Naive Bayes Classification (NBC)* dan *Support Vector Machine (SVM)* pada media sosial *Twitter*.
- c. *Accuracy*, *recall* dan *precision* pada algoritma *Naive Bayes Classification (NBC)* dan *Support Vector Machine (SVM)* menghasilkan kinerja yang baik.

1.3 Rumusan Masalah

Berdasarkan identifikasi masalah, dapat dirumuskan masalah dalam penelitian ini yaitu:

- a. Bagaimana implementasi algoritma *Naive Bayes Classifier (NBC)* dan *Support Vector Machine (SVM)* untuk analisis sentimen pada sosial media pengguna *YouTube*?
- b. Bagaimana hasil analisis sentimen komentar di *YouTube* tentang vaksinasi Covid-19?

- c. Bagaimana perbandingan nilai akurasi pada penerapan algoritma *Naive Bayes Classifier* (NBC) dan *Support Vector Machine* (SVM)?

1.4 Batasan Masalah

Adapun ruang lingkup penelitian ini adalah sebagai berikut:

- Data yang digunakan dalam penelitian ini berasal dari komentar masyarakat pada kanal berita tvOneNews dan KOMPASTV di *YouTube*, terkait dengan Vaksinasi Covid-19 di Indonesia pada bulan Maret sampai Desember 2021.
- Klasifikasi sentimen terbagi menjadi dua kelas yaitu sentimen positif, dan negatif.
- Penelitian berfokus pada klasifikasi opini masyarakat pada komentar *YouTube* dengan membandingkan 2 metode klasifikasi yaitu *Naive Bayes Classifier* (NBC) dan *Support Vector Machine* (SVM) pada tingkat *accuracy*, *precision* dan *recall*.
- Tools* yang digunakan adalah *Google Colaboratory* dengan bahasa pemrograman *Python*.

1.5 Tujuan Penelitian

Berdasarkan penjelasan di dalam latar belakang, maka tujuan umum dari penelitian ini adalah menganalisis sentimen masyarakat terhadap Vaksinasi Covid-19 di Indonesia menggunakan metode *Naive Bayes Classifier* (NBC) dan *Support Vector Machine* (SVM) pada kolom komentar *YouTube* pada periode tertentu.

Adapun penelitian ini memiliki tujuan khusus, adalah untuk mengukur dan membandingkan kinerja algoritma *Naive Bayes Classifier* (NBC) dan *Support Vector Machine* (SVM) berupa nilai *accuracy*, *precision*, dan *recall*.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini dapat digunakan dari beberapa aspek, sebagai berikut:

- a. Secara teoritis, hasil dari penelitian ini sebagai referensi analisis sentimen komentar *YouTube* dengan membandingkan algoritma *Naive Bayes Classifier* (NBC) dan *Support Vector Machine* (SVM) pada nilai *accuracy*, *precision* dan *recall*.
- b. Secara praktis, mengetahui opini masyarakat melalui kolom komentar pada platform *YouTube* terkait perkembangan Vaksinasi COVID-19 di Indonesia, sehingga dapat memberikan wawasan dalam mengetahui perspektif dan pemikiran yang berkembang di masyarakat.

1.7 Metodologi Penelitian

Pada penelitian ini, metode yang dilakukan pada penelitian ini mengacu pada SEMMA *Data Mining Process* (Alizah, 2020). Tahapan yang dilakukan pada penelitian ini sebagai berikut:

a. *Sample*

Pada tahap pengumpulan data, dilakukan pengumpulan data terkait dengan proses vaksinasi Covid-19 pada kolom komentar *YouTube* di kanal TVOneNews dan KompasTV dengan teknik *web scrapping* dari bulan Maret 2021 sampai Desember 2021 yang disimpan ke dalam *local database*.

b. *Explore*

Pada tahap ini dijelaskan deskripsi data dan visualisasi data mengenai apa gambaran besar dari informasi data yang digunakan. Visualisasi data akan memperlihatkan informasi secara visual dari data.

c. *Modify*

Pada tahap ini dilakukan beberapa metode persiapan data (*Data Preprocessing*). Data dimodifikasi dengan membuat, memilih, dan mengubah variabel untuk memusatkan pemilihan model.

d. *Model*

Tahap ini terbagi menjadi dua, yaitu data uji dan data latih. Kemudian mengekstraksi fitur dari data yang sudah diambil sehingga data tersebut dapat dilakukan proses pemodelan *machine learning*. Data latih untuk melakukan prediksi label pada sentimen data yang diuji.

e. *Assess*

Pada tahap ini dilakukan evaluasi terhadap pemodelan yang telah dibuat, lalu membandingkan hasil yang didapat dari model prediksi terhadap data uji dengan label sentimen pada data uji yang sebelumnya sudah diberi label.

Hasil evaluasi yang dilakukan dihitung berdasarkan besaran dari *precision*, *recall* dan akurasi.

1.8 Sistematika Penulisan

Dalam penyusunan laporan penelitian, terbagi dalam lima bab yang secara singkat diuraikan sebagai berikut:

BAB 1 PENDAHULUAN

Dalam bab ini menjelaskan latar belakang, identifikasi masalah, perumusan masalah, batasan masalah, tujuan dan manfaat penelitian, metodologi penelitian, serta sistematika penulisan.

BAB 2 TINJAUAN PUSTAKA

Bab ini menjelaskan dasar-dasar teori dan pengetahuan terkait analisis sentimen secara umum beserta tahapannya, mulai dari pengambilan data (*data collecting*), pembersihan data (*data preprocessing*), konsep dari algoritma atau metode yang terkait, dan penjelasan secara singkat mengenai *tools* yang digunakan guna memberikan gambaran yang lebih jelas tentang penelitian yang dilakukan. Pada bab ini juga mencakup penelitian-penelitian sebelumnya.

BAB 3 METODOLOGI PENELITIAN

Pada bab ini akan menjelaskan dan menguraikan mengenai metodologi yang digunakan, perangkat penelitian, serta tahapan penelitian.

BAB 4 HASIL DAN PEMBAHASAN

Pada bagian ini, membahas implementasi tahapan-tahapan yang digunakan oleh peneliti. Pada bagian ini juga menjelaskan hasil pengumpulan data dan analisis yang telah dilakukan sesuai dengan metodologi yang digunakan.

BAB 5 PENUTUP

Pada bagian ini, peneliti memberikan kesimpulan dan saran terhadap masalah yang diteliti dalam penelitian ini untuk perkembangan penelitian selanjutnya.



BAB 2

TINJAUAN PUSTAKA

2.1 Analisis Sentimen

Analisis sentimen atau *opinion mining* merupakan perpaduan dari *data mining* dan *text mining*, suatu teknik untuk menganalisa pendapat, sentimen, evaluasi, sikap, penilaian, perasaan dan emosi seseorang apakah pembicara atau penulis berkenan dengan suatu topik, produk, layanan, organisasi, individu, ataupun kegiatan tertentu. Dengan analisis sentimen kita bisa mengetahui apakah isi teks itu bersifat positif atau negatif. Sentimen mengacu pada fokus topik tertentu, pernyataan suatu topik mungkin akan berbeda makna dengan pernyataan sama pada subjek berbeda, oleh karena itu, pada beberapa penelitian didahului dengan menentukan elemen dari sebuah produk yang sedang dibicarakan sebelum memulai analisis sentimen (Sipayung *et al.*, 2016).

Ada beberapa jenis Analisis Sentimen, yaitu *Intent Sentiment Analysis* yang bertujuan untuk mengidentifikasi dan menggali motivasi di balik pesan pengguna, apakah itu termasuk keluhan, saran atau pendapat. Tipe berikutnya adalah *Aspect-Best Sentiment Analysis* yang berfokus pada elemen-elemen yang lebih spesifik dari produk dan layanan. Kemudian tipe berikutnya *Fine-Grained Sentiment Analysis* yang merupakan salah satu jenis yang paling umum (Annisa, 2022). Fokusnya ada pada tingkat polaritas pendapat. Pemikiran dasar dari teknik analisis sentimen ini adalah untuk mengelompokkan teks, kalimat, atau dokumen tersebut termasuk kedalam sentimen atau opini yang positif, negatif, atau netral (Sanjaya & Lhaksmana, 2020)

2.2 Naive Bayes Classifier

Naive Bayes Classifier adalah klasifikasi statistik yang dapat digunakan untuk memprediksi probabilitas keanggotaan suatu kelas. *Naive Bayes Classifier* terbukti memiliki akurasi dan kecepatan yang tinggi saat diterapkan pada *database* dengan data yang besar (Tiffani, 2020). *Naive Bayes* dikembangkan oleh Reverend Thomas Bayes pada abad ke 18. *Naive Bayes* menerapkan fungsi statistik sederhana berdasarkan teorema *Bayes* dengan asumsi keberadaan dari suatu fitur tertentu terhadap suatu kelas yang tidak berhubungan dengan fitur lainnya. *Naive Bayes* merupakan suatu metode yang menggunakan perhitungan probabilitas (Sari & Hayuningtyas, 2019). Tahap klasifikasi menggunakan *Naive Bayes Classifier* dibagi menjadi 2 proses, yaitu pelatihan dan pengujian. Proses pelatihan dilakukan untuk menghasilkan probabilistik model fitur yang nantinya akan digunakan sebagai acuan perhitungan untuk mengklasifikasikan data pengujian (Tiffani, 2020).

Untuk menentukan perhitungan *Naive Bayes* dibutuhkan rumus *Naive Bayes*. Berikut rumus *Naive Bayes* pada persamaan 2.1 (Sari & Hayuningtyas, 2019) persamaan 2.2 (Listiowarni & Setyaningsih, 2018), dan Persamaan 2.3.

$$P(H) = \frac{N_j}{N} \quad (2.1)$$

Dengan N_j adalah jumlah data pada suatu *class*, sedangkan N adalah jumlah total data.

$$P(H|X) = \frac{P(X|H).P(H)+1}{P(X)+|V|} \quad (2.2)$$

$$P(X) = \sum Y P(X|H)P(H) \quad (2.3)$$

Keterangan:

X = Data *class* belum diketahui

H = Hipotesis data *class* spesifik

$P(H|X)$ = Probabilitas hipotesis H terhadap kondisi X

$P(H)$ = Probabilitas hipotesis H

$P(X|H)$ = Probabilitas X terhadap kondisi hipotesis H

$P(X)$ = Probabilitas X

2.2.1 Karakteristik *Naive Bayes*

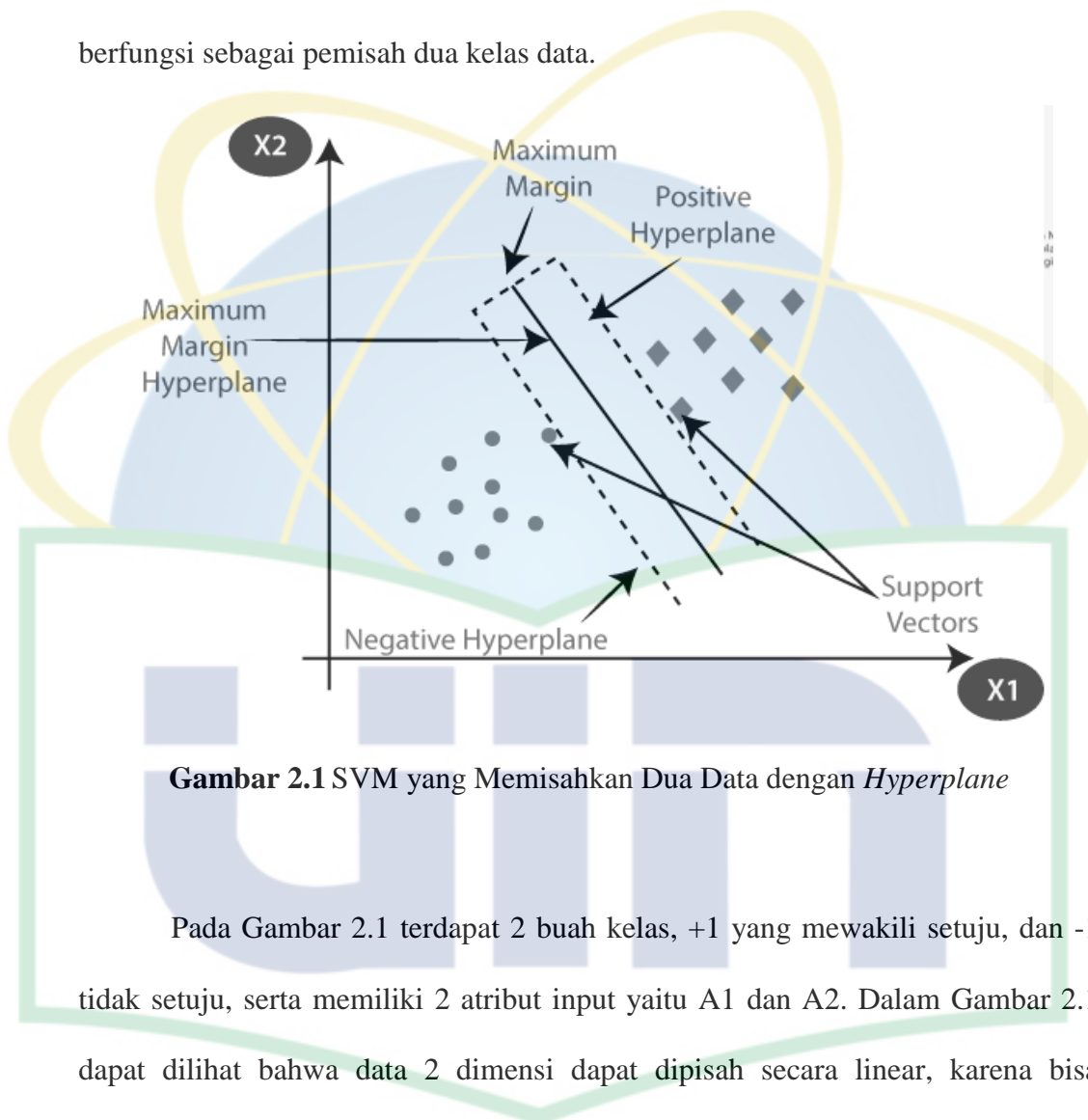
Klasifikasi dengan *Naive Bayes* bekerja berdasarkan teori probabilitas yang memandang semua fitur dari data sebagai bukti dalam probabilitas (Prasetyo, 2012). Hal ini memberikan karakteristik *Naive Bayes* sebagai berikut:

- a. Metode *Naive Bayes* terhadap data-data yang terisolasi yang biasanya merupakan data dengan karakteristik berbeda (*outlier*). *Naive Bayes* juga bisa menangani nilai atau yang salah dengan mengabaikan data latih selama proses pembangunan model dan prediksi.
- b. Tangguh menghadapi atribut yang tidak relevan.
- c. Atribut yang mempunyai korelasi bisa mendegradasi kinerja klasifikasi *Naive Bayes* karena asumsi independensi atribut tersebut sudah tidak ada.

2.3 *Support Vector Machine*

Support Vector Machine merupakan salah satu algoritma klasifikasi yang masuk kelas *supervised learning* yang pertama kali diperkenalkan oleh Vapnik pada tahun 1992 sebagai rangkaian konsep-konsep unggulan dalam bidang *pattern recognition* (Susilowati *et al.*, 2015). *Supervised learning*, merupakan

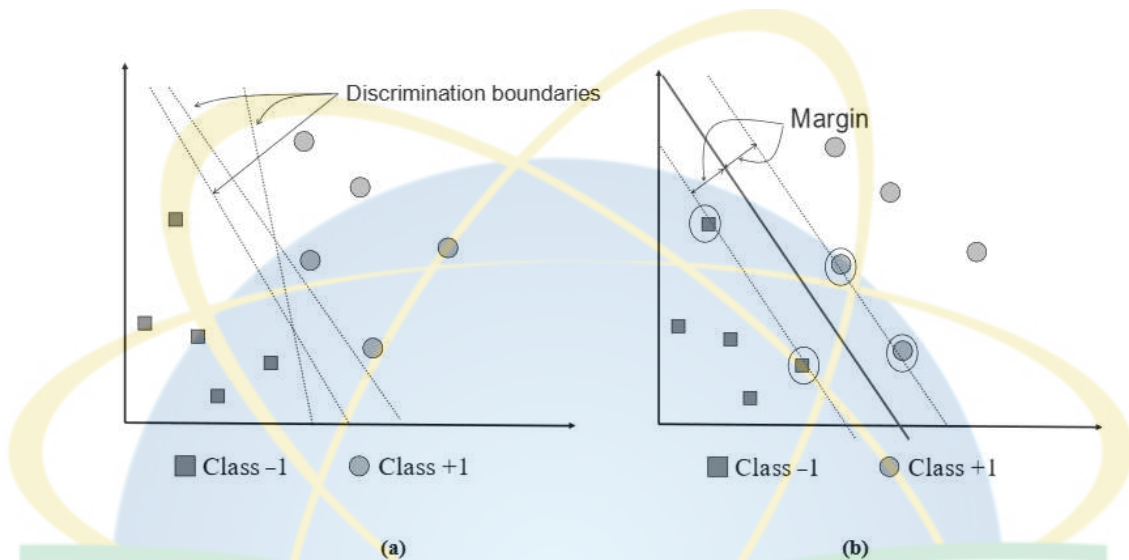
jenis kelas yang di mana dalam implementasinya perlu adanya tahap pelatihan dan pengenalan terhadap suatu objek yang akan dianalisis serta disusul tahap pengujian. Konsep dari SVM adalah dengan mencari *hyperplane* terbaik yang berfungsi sebagai pemisah dua kelas data.



Gambar 2.1 SVM yang Memisahkan Dua Data dengan *Hyperplane*

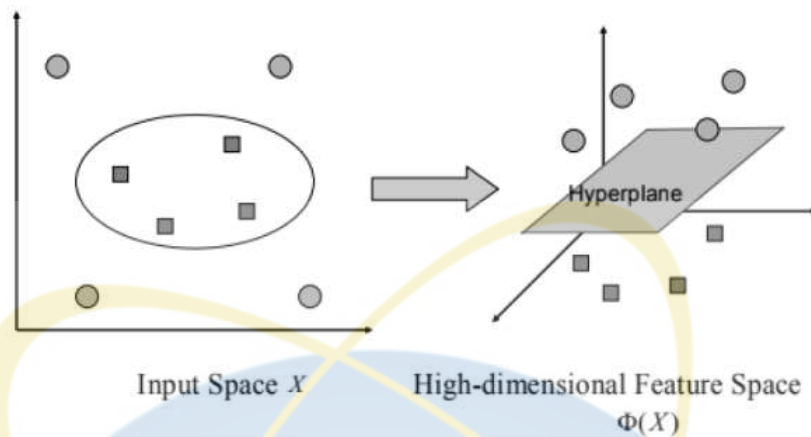
Pada Gambar 2.1 terdapat 2 buah kelas, +1 yang mewakili setuju, dan -1 tidak setuju, serta memiliki 2 atribut input yaitu A1 dan A2. Dalam Gambar 2.1 dapat dilihat bahwa data 2 dimensi dapat dipisah secara linear, karena bisa menarik garis lurus untuk memisahkan semua data tuple kelas +1 dari semua tuple kelas -1, garis hitam ditengah tersebut merupakan *hyperplane* (Gambar 2.2) (Han *et al.*, 2014). *Hyperplane* merupakan garis yang dibentuk sebagai pemisah terbaik antara kedua kelas -1 dan +1 dan dapat dimaksimalkan dengan mengukur *margin* dan mencari titik maksimalnya. *Margin* (garis hitam putus-putus) adalah jarak

antara *hyperplane* tersebut dengan tupel terdekat dari masing-masing *class* (Gambar 2.2) (Han *et al.*, 2014). Tupel yang paling dekat ini disebut sebagai *support vector*



Gambar 2.2 Dua *Hyperplane* di Satu Data

Oleh karena banyaknya garis pemisah maka perlu dicari *hyperplane* terbaik, SVM mendekati masalah ini dengan mencari *Maximum Marginal Hyperplane* (MMH). Kedua *hyperplane* dapat dengan benar mengklasifikasikan semua data tuple yang diberikan pada Gambar 2.2 merupakan gambar jika *hyperplane* digambar terpisah. Permasalahan klasifikasi pada kenyataannya bersifat *non linear*. Untuk menyelesaikan *problem non linear*, SVM dimodifikasi dengan memasukkan fungsi Kernel. Konsep dari *Kernel Trick* dalam SVM adalah dengan membuat dimensi baru, jadi data akan dimasukkan kedalam dimensi baru dan kemudian diberikan *hyperplane* untuk memisahkan data tersebut (Susilowati *et al.*, 2015). Gambar 2.3 merupakan modifikasi SVM untuk *problem non linear*.



Gambar 2.3 Dua *Hyperplane* di Satu Data

Berikut merupakan beberapa fungsi kernel yang populer dan sering digunakan (Ningrum, 2018) sebagai berikut:

a. *Linear Kernel*

Linear kernel merupakan fungsi kernel yang paling sederhana. *Linear* kernel digunakan ketika data yang dianalisis sudah terpisah secara *linear*. *Linear* kernel cocok ketika terdapat banyak fitur dikarenakan pemetaan ke ruang dimensi yang lebih tinggi tidak benar-benar meningkatkan kinerja seperti pada klasifikasi teks.

b. *Polynomial Kernel*

Polynomial kernel merupakan fungsi kernel yang digunakan ketika data tidak terpisah secara *linear*. *Polynomial* kernel sangat cocok untuk permasalahan dimana semua *training* dataset dinormalisasi

c. *Radial Basis Function (RBF)*

RBF kernel merupakan fungsi kernel yang biasa digunakan dalam analisis ketika data tidak terpisah secara *linear*.

Parameter Cost atau biasa disebut sebagai C merupakan parameter yang bekerja sebagai pengoptimalan SVM untuk menghindari misklasifikasi di setiap sampel dalam training dataset (Ningrum, 2018).

2.3 *Text Mining*

Text mining merupakan suatu proses menggali informasi dimana seorang *user* berinteraksi dengan sekumpulan dokumen menggunakan *tools analysis* yang merupakan komponen-komponen dalam *data mining* yang salah satunya adalah kategorisasi (Feldman, James, & Sanger, 2006). *Text mining* dapat memberikan solusi dari permasalahan seperti pemrosesan, pengorganisasian atau pengelompokkan dan menganalisa *unstructured text* dalam jumlah besar. Dalam memberikan solusi, *text mining* mengadopsi dan mengembangkan banyak teknik dari bidang lain, seperti *Data mining*, *Information Retrieval*, Statistik dan Matematik, *Machine Learning*, *Linguistic*, *Natural Language Processing* (NLP), dan *Visualization*.

Tujuan dari *text mining* adalah untuk mendapatkan informasi yang berguna dari sekumpulan dokumen, tetapi tujuan utama *text mining* adalah mendukung proses *knowledge discovery* pada koleksi dokumen yang besar. Adapun tugas khusus dari *text mining* antara lain yaitu pengkategorisasian teks (*text categorization*) dan pengelompokkan teks (*text clustering*) (Eldman & Sanger, 2006).

2.4 *Machine Learning*

Machine Learning adalah ilmu yang mempelajari bagaimana komputer dapat belajar atau meningkatkan kinerja berdasarkan data agar secara otomatis

mengenali pola data untuk membuat keputusan cerdas berdasarkan data (Han *et al.*, 2014). *Machine Learning* merupakan suatu area dalam *Artificial Intelligence* (AI) yang berhubungan dengan pengembangan teknik-teknik yang bisa diprogramkan dan belajar dari data masa lalu (McGranaghan & Santoso, 2007).

Menurut Mohri *et al.* (2018) *Machine Learning* dapat didefinisikan sebagai metode komputasi berdasarkan pengalaman untuk meningkatkan performa atau membuat prediksi yang akurat. Pengalaman di sini didefinisikan sebagai data historis yang tersedia dan dapat dijadikan sebagai data pembelajaran (*training data*). Dalam pembelajaran *Machine Learning*, terdapat beberapa skenario, seperti:

a. *Supervised Learning*

Penggunaan skenario *supervised learning*, pembelajaran menggunakan masukan data pembelajaran yang telah diberi label. Setelah itu membuat prediksi dari data yang telah diberi label.

b. *Unsupervised Learning*

Penggunaan skenario *unsupervised learning*, pembelajaran menggunakan masukan data pembelajaran yang tidak diberi label. Setelah itu mencoba untuk mengelompokkan data berdasarkan karakteristik-karakteristik yang ditemui.

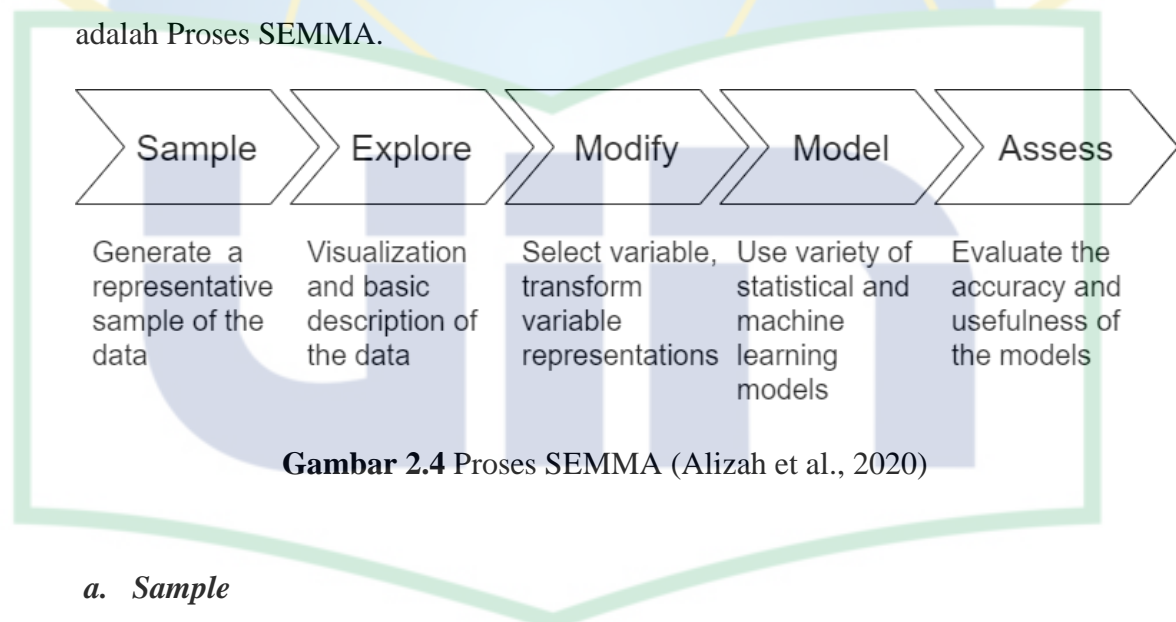
c. *Reinforcement Learning*

Pada skenario *reinforcement learning* fase pembelajaran dan tes saling dicampur. Untuk mengumpulkan informasi pembelajaran secara aktif dengan

berinteraksi ke lingkungan sehingga untuk mendapatkan balasan untuk setiap aksi dari pembelajaran.

2.5 Metode SEMMA

SEMMA merupakan singkatan dari *Sample, Explore, Modify, Model, dan Assess*. Metode ini dapat ditemukan oleh *SAS Institute*. Proses *data mining* SEMMA dapat digunakan dengan mudah dan mudah dipahami proses yang terkait dalam pemeliharaan proyek *data mining*. Proses *data mining* SEMMA memiliki 5 proses tahapan yaitu *Sample, Explore, Modify, Model, dan Assess*, dari masing-masing tersebut memiliki peran sendiri dalam proses *data mining* dan memiliki manfaat dalam proses *data mining* tersebut (*SAS Institute, 2017*). Gambar 2.4 adalah Proses SEMMA.



a. *Sample*

Pada tahap pengumpulan data, dilakukan pengumpulan data terkait dengan proses vaksinasi Covid-19 pada kolom komentar *YouTube* di kanal *TVOneNews* dan *KompasTV* dengan teknik *web crawling* dari bulan Maret 2021 sampai Desember 2021.

b. Explore

Pada tahap ini dijelaskan deskripsi data dan visualisasi data mengenai apa gambaran besar dari informasi data yang digunakan. Visualisasi data akan memperlihatkan informasi secara visual dari data.

c. Modify

Pada tahap ini dilakukan beberapa metode persiapan data (*Data Preprocessing*). Data dimodifikasi dengan membuat, memilih, dan mengubah variabel untuk memusatkan pemilihan model. Tahapan persiapan data (Yulita *et al.*, 2021), diantaranya:

- *Case Folding*

Tahap ini merupakan proses penyeragaman huruf, baik huruf kecil maupun huruf besar.

- *Tokenize*

Suatu proses yang dilakukan untuk memotong atau memecah kalimat menjadi bagian-bagian atau kata-kata. Tokenisasi juga dilakukan dengan menghilangkan tanda baca yang tidak diperlukan.

- *Normalize*

Normalize merupakan tahap dimana dilakukan standarisasi kata yang memiliki makna sama dengan melakukan perubahan penulisan pada suatu kata yang disingkat dan tidak baku agar memiliki arti kata yang seragam. *Stemming* Tahapan ini mengubah kata menjadi kata dasar menurut kaidah bahasa Indonesia yang baik dan benar

- *Stemming*

Stemming merupakan proses yang digunakan untuk mengembalikan kata-kata ke dalam kata dasarnya. Hal ini juga bertujuan untuk membersihkan suatu kata dengan pengejaan yang kurang tepat.

- *Stopword Removal*

Tahap ini merupakan proses untuk melakukan *filter* terhadap kata-kata umum seperti “dan”, “atau”, dan lainnya, yang tidak diperlukan saat pemrosesan data.

d. ***Model***

Tahap ini terbagi menjadi dua, yaitu data uji dan data latih dengan rasio 3:7.

Kemudian mengekstraksi fitur dari data yang sudah diambil sehingga data tersebut dapat dilakukan proses pemodelan *machine learning*. Data latih untuk melakukan prediksi label pada sentimen data yang diuji.

e. ***Assess***

Pada tahap ini dilakukan evaluasi terhadap pemodelan yang telah dibuat, lalu membandingkan hasil yang didapat dari model prediksi terhadap data uji dengan label sentimen pada data uji yang sebelumnya sudah diberi label.

Hasil evaluasi yang dilakukan dihitung berdasarkan besaran dari *precision*, *recall*, dan akurasi.

2.6 *Text Preprocessing*

2.6.1 *Pengertian Text Preprocessing*

Text preprocessing merupakan tahapan dari proses awal terhadap teks untuk mempersiapkan teks menjadi data yang akan diolah lebih lanjut. Suatu teks

tidak dapat diproses langsung oleh algoritma pencarian, oleh karena itu dibutuhkan *preprocessing text* untuk mengubah teks menjadi data *numeric*. Sebuah teks yang ada harus dipisahkan, hal ini dapat dilakukan dalam beberapa tingkatan yang berbeda. Suatu dokumen dapat di pecah menjadi bab, sub-bab, paragraf, kalimat dan pada akhirnya menjadi potongan kata/token. Selain itu pada tahapan ini keberadaan digit angka, huruf kapital, atau karakter-karakter yang lainnya dihilangkan dan dirubah (Eldman & Sanger, 2006).

Preprocessing berfungsi untuk proses awal sebelum dokumen teks diolah pada tahap selanjutnya dimana akan dilakukan proses seleksi data yang akan diproses pada setiap dokumen. Proses ini terdiri atas beberapa proses pembersihan dokumen, yaitu *case folding*, *cleansing*, *tokenizing*, *filtering* atau *stopword removal*, dan *stemming* (Nugroho & Rilvani, 2023).

2.6.2 Tahap Preprocessing Data

Secara umum proses yang dilakukan dalam tahapan *preprocessing* adalah sebagai berikut:

a. Case Folding

Case Folding merupakan proses penyamaan *case* dalam sebuah dokumen. Hal ini dilakukan untuk mempermudah pencarian. Tidak semua dokumen teks konsisten dalam penggunaan huruf kapital. Oleh karena itu peran *case folding* dibutuhkan dalam mengkonversi keseluruhan teks dalam dokumen menjadi suatu bentuk standar, dalam hal ini huruf kecil atau *lowercase* (Kulkarni & Shivananda, 2019).

b. *Cleaning*

Tahapan *cleaning* merupakan langkah awal yang dilakukan dalam pra-pemrosesan data terutama pada jenis data bersifat tekstual. Tahapan ini sangat penting dilakukan untuk memastikan data yang akan diolah telah bersih dan tidak terlalu banyak *noisy*, sehingga memudahkan mesin dalam memproses langkah selanjutnya. Tahapan ini dapat mencakup penghapusan tanda baca (mis: . , ? !), nilai angka, karakter spesial non-ASCII (mis: @ \$ # &), link URL, dan penggunaan spasi berlebih. Pemilihan atribut yang akan digunakan juga termasuk dalam tahapan ini (Kulkarni & Shivananda, 2019).

c. *Tokenize*

Tokenize adalah proses penguraian teks yang semula berupa kalimat-kalimat yang berisi kata-kata. Proses ini diawali dengan menghilangkan *delimiter* yaitu simbol dan tanda baca yang ada pada teks tersebut. *Tokenize* (dikenal juga sebagai penganalisis leksikal atau segmentasi kata) adalah proses mengelompokkan teks ke dalam unit yang berarti menjadi potongan-potongan yang disebut token agar dapat dilakukan analisis. Secara umum, *tokenization* melakukan segmentasi kata dari setiap kalimat secara langsung dengan membuat masing-masing token diambil sebagai urutan karakter yang biasanya dipisahkan oleh spasi (Kulkarni & Shivananda, 2019).

d. *Normalize*

Normalize merupakan tahap mengubah kata *slang* atau singkatan menjadi kata baku sesuai dengan KBBI. Proses normalisasi perlu dilakukan untuk

dataset yang bersumber dari media sosial dimana banyak pengguna yang memakai kata singkatan atau kata tidak baku agar memiliki penulisan yang seragam dan sama (Ganesan, 2019).

e. *Stemming*

Stemming merupakan proses pengolahan kata untuk mendapatkan kata dasar dari sebuah kata yang telah mengalami imbuhan dengan asumsi bahwa katakata tersebut sebenarnya memiliki makna dan arti yang sama. Algoritma ini bekerja berdasarkan struktural morfologi dalam kalimat Bahasa Indonesia, yang terdiri atas awalan, akhiran, sisipan, dan awalan+akhiran. Tujuan dari tahap ini adalah:

- 1) Keefisiensian, pada *stemming* dilakukan pengurangan jumlah kata dalam dokumen agar mengurangi kebutuhan dalam ruang penyimpanan dan mempercepat dalam melakukan pencarian.
- 2) Keefektifan, *stemming* dilakukan untuk mengurangi *recall* dengan pengurangan bentuk-bentuk kata ke dalam bentuk dasarnya (Putri, 2016). Sebagai contoh adalah kata “duduk-lah”, “minum-lah”, “jikapun”, dan sebagainya.

f. *Stopword Removal*

Stopword Removal merupakan proses untuk melakukan *filter* terhadap kata-kata umum yang sering muncul tapi tidak memiliki arti penting dan maknanya tidak berpengaruh pada sistem (Taufiqurrahman et al., 2021) seperti “dan”, “ke”, “atau”, dan lainnya.

2.7 *Lexicon Based*

Dalam pendekatan analisis sentimen dengan menggunakan *Lexicon Based* yang merupakan metode berdasarkan kamus dengan tujuan untuk mendapatkan bobot kalimat pada data sehingga bisa diketahui label kelas sentimen pada dataset (Herdhianto, 2020). Pada penelitian ini menggunakan kamus *lexicon* yang bersumber dari GitHub (Martua, 2020). Metode *Lexicon* memiliki kelebihan membandingkan secara otomatis kata-kata pada dataset langsung dengan kamus kata yang terdapat pada *lexicon*, sehingga dapat menghemat waktu jika data yang diolah adalah dataset yang dengan jumlah yang besar (Nuri, 2022).

2.8 **Kualitas Hasil Klasifikasi**

Menurut Han & Kamber (2006) klasifikasi didefinisikan sebagai teknik analisis data yang digunakan untuk menghasilkan model prediksi untuk mendeskripsikan label atau kelas data. Sebuah algoritma klasifikasi akan membangun sebuah model klasifikasi dengan cara menganalisis data *training*. Tahap pembelajaran dapat juga dipandang sebagai tahap pembentukan fungsi atau pemetaan $y = f(x)$, dimana y merupakan kelas hasil prediksi dan x adalah *record* yang ingin diprediksi kelasnya (Han & Kamber 2006).

Kualitas hasil kualifikasi dapat dinilai dan dievaluasi berdasarkan beberapa ukuran: (Manning et al., 2009)

a. *Accuracy*

Akurasi adalah jumlah proporsi prediksi yang benar. Akurasi digunakan sebagai tingkat ketepatan antara nilai aktual dengan nilai prediksi.

b. *Precision*

Presisi adalah proporsi jumlah dokumen teks yang relevan terkendali diantara semua dokumen yang dipilih sistem. Presisi digunakan sebagai tingkat ketepatan anata informasi yang diminta dengan jawaban yang diberikan oleh sistem.

c. *Recall*

Recall adalah proporsi jumlah dokumen teks yang relevan terkendali diantara semua dokumen teks relevan yang ada pada koleksi. *Recall* digunakan sebagai ukuran keberhasilan sistem dalam menemukan kembali informasi.

2.9 *Confusion Matrix*

Confusion matrix digunakan untuk mengetahui performa klasifikasi *machine learning* yang dipresentasikan dalam bentuk matriks yang memberikan perbandingan antara nilai aktual dan nilai prediksi (Fatihin, 2022). *Confusion matrix* adalah sebuah tabel yang menyatakan jumlah data uji yang benar diklasifikasikan dan jumlah data uji yang salah diklasifikasikan. Contoh *confusion matrix* untuk klasifikasi biner ditunjukkan pada Tabel 2.3 (Vaid et al., 2020).

Tabel 2.1 *Confusion Matrix*

Kelas Sebenarnya	Kelas hasil prediksi	
	Positif = 1	Negatif = 0
Positif = 1	TP (<i>True Positive</i>)	FN (<i>False Negative</i>)
Negatif = 0	FP (<i>False Positive</i>)	TN(<i>True Negative</i>)

- a. **True Positive (TP)** = jumlah dokumen dari kelas 1 yang benar dan diklasifikasikan sebagai kelas 1. Data dari klasifikasi yang memiliki label positif dan label klasifikasi tersebut sesuai dengan nilai aktual.
- b. **True Negative (TN)** = jumlah dokumen dari kelas 0 yang benar diklasifikasikan sebagai kelas 0. Data dari klasifikasi yang memiliki label negatif dan label klasifikasi tersebut sesuai dengan nilai aktual.
- c. **False Positive (FP)** = jumlah dokumen dari kelas 0 yang salah diklasifikasikan sebagai kelas 1. Data dari klasifikasi yang memiliki label positif dan label tidak sesuai dengan nilai aktual.
- d. **False Negative (FN)** = jumlah dokumen dari kelas 1 yang salah diklasifikasikan sebagai kelas 0. Data dari klasifikasi yang memiliki label negatif dan label tidak sesuai dengan nilai aktual.

2.10 ***K-Fold Cross Validation***

Cross-validasi adalah teknik validasi model untuk menilai bagaimana hasil statistik analisis akan menggeneralisasi kumpulan data independen. Teknik ini utamanya digunakan untuk melakukan prediksi model dan memperkirakan seberapa akurat sebuah model prediktif ketika dijalankan dalam praktiknya. Salah satu teknik dari validasi silang adalah *k-fold cross validation*, yang mana memecah data menjadi k bagian set data dengan ukuran yang sama.

Nilai *k-folds* yang biasa digunakan adalah 3, 5, 10 dan 20 (Hilmiyah, 2017). Simulasi mengenai pembagian data train dan data test pada gambar 2.5



Gambar 2.5 *K-Folds Cross Validation* (Sanjay, 2018)

2.11 *YouTube*

YouTube merupakan situs terpopuler kedua di dunia pada Juli 2020 menurut sebuah perusahaan analisis lalu lintas web, *Alexa Internet*, setelah *Google* yang menempati posisi pertama. Sedangkan untuk Indonesia sendiri, *YouTube* merupakan media sosial terpopuler dengan sebesar 93,8% pengguna aktif media sosial memakainya berdasarkan data dari “*Hootsuite (We are Social, 2021): Indonesian Digital Report 2021*” di *We Are Social* Indonesia, pada tahun 2021 jumlah pengguna media sosial di Indonesia yaitu 170 juta (61,8% dari jumlah populasi di Indonesia).

Sebagai salah satu dari media sosial, *YouTube* dianggap sebagai medium yang dapat menyediakan informasi yang lengkap sekaligus murah, bahkan tidak berbayar. Itu lah sebabnya platform ini dipercaya sebagai televisi versi terkini yang membutuhkan perangkat lebih sederhana dibandingkan dengan perangkat untuk menonton televisi. Alternatif konten di *YouTube* sangat beragam, namun masih juga terdapat audiens yang ingin menonton program televisi melalui platform *YouTube* dengan melakukan *streaming*. Ragam program televisi yang

masih dikejar oleh audiens hingga ke *YouTube* biasanya terkait dengan informasi olahraga atau berita terkini (Kusuma & Prabayanti, 2022).

Terdapat beberapa *channel* yang memberikan informasi kepada masyarakat terkait perkembangan vaksinasi covid-19 di Indonesia, seperti tvOneNews dan Kompas TV. Kedua *channel* tersebut memiliki jumlah *subscriber* yang banyak, yaitu 10,4 juta pada tvOneNews dan 14,6 juta pada Kompas Tv. Sehingga teks yang dihasilkan oleh pengguna melalui komentar menjadi besar dan memberi peluang untuk dilakukan analisis (Vlachos & Tan, 2018).

2.11 Bahasa Pemrograman *Python*

Python merupakan sebuah bahasa pemrograman, *Python* sendiri mulai dikembangkan pada tahun 1989 oleh *Guido van Rossum*, seorang *engineer* di *Google Inc.* pada saat itu. Nama *Python* sendiri bukan berasal dari nama spesies ular, namun berasal dari sebuah grup komedi di Inggris yang bernama *Monty Python*. Versi *stable* perdana *Python*, yaitu *Python 1.0*, dirilis pada tahun 1991. *Python* memiliki *high-level* struktur data, *dynamic typing* dan *dynamic binding*. *Python* memiliki sintaks sederhana dan mudah dipelajari untuk penekanan pada kemudahan membaca dan mengurangi biaya perbaikan program. *Python* banyak digunakan oleh *non-programmer* dan ilmuwan. *Python* mendukung modul dan paket untuk mendorong kemodularan program dan *code reuse*. Interpreter *Python* dan *standard library*-nya tersedia secara gratis untuk semua *platform* dan dapat secara bebas disebar (Python Software Foundation, 2021).

2.12 *Google Colaboratory*

Colaboratory, atau “Colab” merupakan produk dari Google Research. Colab memungkinkan siapa saja menulis dan mengeksekusi kode python melalui browser, dan sangat cocok untuk machine learning, analisis data, serta pendidikan. Secara lebih teknis, Colab merupakan layanan notebook Jupyter yang dihosting dan dapat digunakan tanpa penyiapan, serta menyediakan akses gratis ke resource komputasi. Resource Colab tidak dijamin dan sifatnya terbatas, serta batas penggunaannya terkadang berfluktuasi. Hal ini diperlukan agar Colab dapat menyediakan resource secara gratis (Irfon & Soen, 2022).

2.13 *Library*

2.13.1 *Selenium*

Selenium adalah *library* bersifat *open source* yang digunakan untuk *web scraping*. *Selenium* bekerja dengan mengotomatisasi *browser* untuk memuat situs *web*, mengambil data yang diperlukan, dan bahkan mengambil tangkapan layar atau menyatakan bahwa tindakan tertentu terjadi di situs *web*. *Selenium* tidak mengandung *browser web* sendiri, sehingga membutuhkan integrasi dengan *browser* pihak ketiga untuk dapat berjalan. Jika menjalankan *Selenium* dengan *Firefox*, misalnya, akan melihat contoh *Firefox* terbuka di layar, navigasikan ke situs *web*, dan lakukan tindakan yang telah ditentukan dalam *script* (Mitchell, 2018).

2.13.2 *Pandas*

Pandas adalah sebuah *library* di *Python* yang berlisensi BSD dan *open source* yang menyediakan struktur data dan analisis data yang mudah digunakan.

Pandas biasanya digunakan untuk membuat tabel, mengubah dimensi data, mengecek data, dan lain sebagainya. Struktur data dasar pada *Pandas* dinamakan *DataFrame*, yang memudahkan kita untuk membaca sebuah *file* dengan banyak jenis format seperti *file* .txt, .csv, dan .tsv (McKinney, 2011).

2.13.3 Numpy

NumPy (*Numerical Python*) merupakan sebuah bagian dari *library Python* yang fokus pada *scientific computing*. *NumPy* memiliki kemampuan untuk membentuk objek *N dimensional array*, yang mirip dengan *list* pada *Python*. Keunggulan *NumPyarray* dibandingkan dengan *list* pada *Python* adalah konsumsi memory yang lebih kecil serta *runtime* yang lebih cepat (Scipy, 2021).

2.13.4 Sastrawi

Salah satu *library* yang bisa digunakan dalam melakukan proses *stemming* bahasa Indonesia adalah dengan menggunakan *library python Sastrawi*. *Library* ini merupakan pengembangan dari *library PHP Sastrawi*, dimana *library* tersebut menerapkan algoritma NA (*Nazief & Adriani stemmer*). Tahapan algoritma tersebut meliputi:

- a. Langkah pertama adalah memeriksa apakah kata tersebut merupakan kata dasar (*root*) terdapat dalam daftar akar kata (*root*). Ketika kata tersebut merupakan kata dasar, maka proses dihentikan pada tahap pertama ini.
- b. Menghilangkan *Inflection Suffixes* (“-lah”, “-kah”, “-ku”, “-mu”, atau “-nya”). Jika kata berupa partikel (“-lah”, “-kah”, “-tah” atau “-pun”) maka langkah ini diulangi lagi untuk menghapus *Possesive Pronouns* (“-ku”, “-mu”, atau “-nya”).

- c. Menghilangkan *derivational suffix* (imbuhan turunan). Hilangkan imbuhan -i, -kan, -an.
- d. Menghilangkan *derivational prefix* (awalan turunan). Hilangkan awalan be-, di-, ke-, me-, pe-, se- dan te-
- e. Bila dari langkah 4 di atas belum ketemu juga. Maka lakukan analisis apakah kata tersebut masuk dalam tabel diambiguitas kolom terakhir atau tidak.
- f. Bila semua proses di atas gagal, maka algoritma mengembalikan kata aslinya.

2.14 Penelitian Sejenis

Penelitian sejenis atau literatur sejenis adalah tinjauan pustaka di mana penulis mengumpulkan berbagai penelitian terdahulu yang berkaitan dengan topik yang sedang peneliti lakukan, kemudian melakukan perbandingan antara peneliti dengan penelitian sebelumnya serta sebagai referensi dalam penelitian. Pada Tabel 2.3 merupakan kumpulan literatur sejenis dalam analisis sentimen dengan Algoritma *Naive Bayes Classifier* dan *Support Vector Machine* dalam berbagai penelitian.

Tabel 2.2 Penelitian Sejenis *Algorithma Naive Bayes Classifier* dan *Support Vector Machine*

No	Penulis	Metode	Tools	Hasil	Dataset	Kelebihan	Kekurangan
1.	Romadoni <i>et al.</i> (2020)	<i>Support Vector Machine</i>	Bahasa Program: <i>R Studio</i>	Klasifikasi dengan empat skenario <i>splitting</i> data yaitu rasio 60:40, 70:30, 80:20, 90:10 dan empat <i>kernel</i> yaitu <i>kernel linear</i> , <i>rbf</i> , <i>sigomid</i> , dan <i>polynomial</i> , serta mendapatkan nilai akurasi sebesar 98,7%.	Data <i>Tweet</i> uang elektronik (OVO, LinkAja) sebanyak 3852 data, 2034 data diklasifikasikan ke dalam kelas positif, dan 1818 data diklasifikasikan ke dalam kelas negatif.	- Memiliki tingkat akurasi yang baik - Jumlah data yang digunakan termasuk banyak yaitu total 3852 data.	- Tidak menjelaskan <i>library</i> yang digunakan - Kurangnya informasi spesifikasi <i>hardware</i> dalam <i>preprocessing data</i> .
2.	Herdhianto (2020)	<i>Naive Bayes Classifier</i>	Bahasa Program <i>PHP</i>	Hasil akurasi Algoritma <i>Naive Bayes Classifier</i> (NBC) dengan seleksi fitur	Data <i>tweet</i> mengenai zakat sebanyak 1000	- Hasil <i>output</i> klasifikasi digambarkan	Pada tahap <i>Preprocessing</i> data kurang dipaparkan

		(NBC)	<p><i>Database:</i> XAMPP</p> <p><i>OS:</i> Windows 10 64-bit</p>	<p><i>Term-Frequency</i> sebesar 74%, hasil presisi mendapatkan nilai 79,3% untuk sentimen positif dan 66,7% untuk sentimen negatif.</p>	<p>data. Dengan data latih sebanyak 950 dan data uji sebanyak 50</p>	<p>dengan jelas oleh penulis dengan menggunakan tabel dan grafik</p> <ul style="list-style-type: none"> - Jumlah data yang digunakan termasuk banyak 	secara <i>detail</i>
3.	Zalyhaty (2021)	<i>Support Vector Machine</i>	<p>Bahasa Program Python</p> <p><i>Software:</i> OS Windows 10 Home, Ms. Excel 2019 dan</p>	<p>Hasil persentase sentimen positif sebanyak 66,8% dan negatif 33,2%</p> <p>Hasil rata-rata validasi menggunakan 5-fold cross validation menunjukkan nilai 74,05% mengartikan model klasifikasi masih belum sempurna</p>	<p>Berita <i>online</i> dengan topik vaksin Covid-19 sebanyak 283 data</p>	<ul style="list-style-type: none"> - Tahapan <i>preprocessing data</i> dipaparkan oleh penulis secara rinci - Tahapan pada pembobotan seleksi fitur juga dijelaskan dalam penelitian 	<ul style="list-style-type: none"> - Jumlah data yang digunakan kurang banyak - Pelabelan positif dan negatif masih dengan cara manual, tidak dijelaskan standar pelabelan positif dan negatifnya

			<i>Google Collaboratory</i>				- Penggambaran hasil klasifikasi masih kurang mudah untuk dipahami
4.	Villavicencio et al. (2021)	<i>Naive Bayes Classifier</i>	<i>Rapid Miner</i>	Diantara 993 tweet, sebanyak 83,38% mengandung sentimen positif, 8,26% memiliki polaritas negatif dan 8,36% memiliki polaritas netral.	11.974 tweet bahasa Inggris dan bahasa Tagalog mengenai vaksin Covid-19, dengan hashtag #covidvaccineph, #covid19vaccineph, #resbakuna, #BIDABakunatio n, #BIDASolusyon, #WeHealAsOneP	- Data yang dikumpulkan cukup banyak, menggunakan berbagai kata kunci yang beragam untuk pengambilan data di twitter - Pada tahap preprocessing sampai evaluation dijelaskan secara	- Waktu pengumpulan data hanya memanfaatkan selama 1 bulan - Hasil analisis, dan grafik yang dihasilkan dari tools RapidMiner perlu untuk ditingkatkan kembali

					<i>H, #covaxph</i>	detail oleh penulis.	
5.	Novendri <i>et al.</i> (2020)	<i>Naive Bayes Classifier</i>	<i>Bahasa Program Python</i>	<ul style="list-style-type: none"> - 80% data untuk training, dan 20% data untuk testing - Seleksi fitur menggunakan TF-IDF - Nilai akurasi NBC sebesar 81%, <i>Precision</i> 74,83%, <i>Recall</i> 75,22% 	Data komentar <i>YouTube</i> trailer film <i>Money Heist</i> <i>session</i> 4 sebanyak 998 data	Model evaluasi dengan seleksi fitur dijelaskan secara detail	<ul style="list-style-type: none"> - Persentase data testing kurang banyak, yaitu hanya sebesar 20% dari data <i>tweet</i> yang didapat - Tidak dipaparkan dengan detail data <i>Preprocessing</i>
6.	Abdulloh & Pambudi (2021)	<i>Naive Bayes Classifier</i>	<i>Python</i>	Hasil pengujian sentimen negatif sebesar 41% dan sentimen positif 17%. Setelah peristiwa vaksinasi, sentimen negatif terhadap vaksin menjadi 24% dan sentimen positif sebesar 36%.	1448 data mengenai vaksinasi <i>covid</i> pada platform <i>YouTube</i> , dengan 504 dataset negatif, 584 dataset netral, 360	Pengujian dilakukan dengan 2 tahapan, pertama menguji dengan 3 kelas sentimen positif, netral dan negatif. Kemudian	Dataset yang digunakan masih kurang, sehingga pelabelan yang dilakukan kurang akurat

				<p>nilai akurasi, yaitu:</p> <ul style="list-style-type: none"> - <i>Kernel linear</i>, yaitu 75,7% - <i>Kernel poly</i>, yaitu 76,12% - <i>Kernel rbf</i>, yaitu 77,75% - <i>Kernel sigmoid</i>, yaitu 75,75% 	dataset positif	<p>pegujian kedua dilakukan dengan 2 kelas sentimen berupa positif dan negatif.</p>	
7.	Odin <i>et al.</i> (2020)	<i>Naive Bayes Classifier</i>	<i>Python</i>	<p>Menunjukkan 62,53% sentimen negatif dan 37,47% positif dengan akurasi klasifikasi lebih dari 0,97.</p> <p>Nilai akurasi:</p> <ul style="list-style-type: none"> - 3000 <i>data training</i> menghasilkan nilai akurasi 57% - 5000 <i>data training</i> 	<p>25.013 sentimen negatif dan 14.986 sentimen positif terhadap maskapai penerbangan</p>	<p>percobaan klasifikasi akurasi dengan sejumlah <i>data training</i> menunjukkan garis lurus hubungan antara <i>data training</i> dan akurasinya</p>	<p>Kurang mempertimbangkan pengembangan mekanisme yang optimal untuk penentuan ukuran <i>data training</i>.</p>

				<p>menghasilkan nilai akurasi 68%</p> <p>- 7000 data <i>training</i> menghasilkan nilai akurasi 75%</p> <p>- 11000 data <i>training</i> menghasilkan nilai akurasi 90</p>			
8.	Prastyo et al. (2020)	Support Vector Machine	Jupyter notebook, python, tweetscrap per library	<p>Penggunaan algoritma SVM dengan <i>Normalized Poly Kernel</i> memberikan hasil terbaik dalam memprediksi sentimen dengan memberikan akurasi tertinggi pada sentimen berdasarkan aspek umum, yaitu 787 <i>tweet</i> positif, 482 netral, dan 934 negatif.</p>	<p>- 2.203 <i>tweet</i> untuk <i>general aspect</i>: 787 positif, 482 netral, dan 934 sentimen negatif.</p> <p>- <i>Economic aspect</i>, 1.941 <i>tweet</i> yang terdiri dari 973</p>	<p>Dalam penelitian ini dibangun algoritma <i>machine learning</i> untuk memprediksi analisis sentimen dari data yang tidak terlihat dengan membandingkan</p>	<p>Pemilihan model dan fitur masih belum digabung, sehingga menyebabkan sehingga waktu komputasi yang tidak sedikit</p>

				<p>Nilai akurasi berdasarkan aspek umum menggunakan dua kelas dengan akurasi rata-rata 82,00%, <i>precisi</i> 82,24%, <i>recall</i> 82,01%, dan <i>f-measure</i> 81,84%.</p>	<p>sentimen positif, 385 netral, dan 585 negatif.</p>	<p>dua algoritma, SVM dengan <i>Normalized Poly Kernel</i> dan MNB</p>	
9.	Ahmad <i>et al.</i> (2018)	Support Vector Machine	Python	<p>Hasil dari proses pengujian 1.236 <i>tweet</i> (404 positif dan 832 negatif) menggunakan SVM. Nilai akurasi yang didapat, yaitu akurasi 96,68%, <i>precision</i> 95,82%, <i>recall</i> 94,04% dan AUC 0,979</p>	<p>Data <i>output</i> dari proses <i>handling duplicate</i> berjumlah 1.236 <i>tweet</i> mengenai pemindahan Ibukota Baru pada <i>Twitter</i></p>	<p>- <i>Perfomance</i> diukur dengan akurasi dan <i>Area Under Curve</i> (AUC) yang ditampilkan dalam bentuk kurva ROC</p> <p>- Akurasi yang didapat mendapatkan</p>	<p>Pada data sentimen wacana pemerintah terkait topik pemindahan ibu kota dibutuhkan usaha yang besar pada tahap <i>text processing</i> awal yakni pada bagian <i>labeling</i>.</p>

						hasil yang tinggi	
10.	Imamah et al. (2020)	Support Vector Machine	Python, Google Colaboratory	Persentase sentimen positif sebesar 89% sedangkan sentimen negatif sebesar 11%. kemudian diklasifikasikan, memiliki akurasi 70,22% , Presisi 65,38% dan Recall 75,36%	Data review TripAdvisor selama bulan Oktober 2019 pada Bangkalan Regency sebanyak 1394 data	<ul style="list-style-type: none"> - Penelitian menggunakan <i>Confusion matrix</i> yang memudahkan pembaca untuk membaca hasil pengukuran - Menggunakan 4 <i>term</i> sebagai representasi dari hasil proses klasifikasi 	Nilai akurasi yang dihasilkan masih terbilang kecil yaitu hanya 70,22%

Peneliti telah melakukan tinjauan pustaka dari banyak jurnal yang berhubungan dengan Analisis Sentimen serta algoritma yang digunakan dalam klasifikasi *machine learning*. Dari banyaknya jurnal yang ditinjau, peneliti memasukkan 10 jurnal dalam penelitian ini sebagai penelitian sejenis. Berdasarkan Tabel 2.9 dengan topik penelitian yang penulis lakukan mendekati penelitian sejenis nomor 3, 5, 6 dan 10. Namun terdapat beberapa kelebihan pada penelitian yang dilakukan yaitu:

- a. Menggunakan dua algoritma klasifikasi yaitu: *Naive Bayes Classifier* dan *Support Vector Machine*.
- b. Penulis menggunakan metodologi SEMMA.
- c. Menggunakan metode *Lexicon Based* untuk pelabelan kelas.
- d. Menggunakan metode validasi *K-Fold Cross Validation*.

2.15 Perumusan Hipotesis Komparatif

Hipotesis adalah jawaban sementara dari suatu masalah komparatif yang dihadapi dan perlu diuji kebenarannya dengan data yang lebih lengkap dan menunjang. Penelitian ini dilakukan untuk mengetahui apakah akurasi dari algoritma *naive bayes classifier* menghasilkan nilai yang lebih besar atau lebih kecil dari algoritma *support vector machine*. Berikut ini perumusan hipotesis dalam penelitian ini:

Ho:

- a. Penggunaan data *testing* yang kecil dibanding data *training* mempengaruhi akurasi algoritma *naive bayes classifier* lebih dari algoritma *support vector machine*.

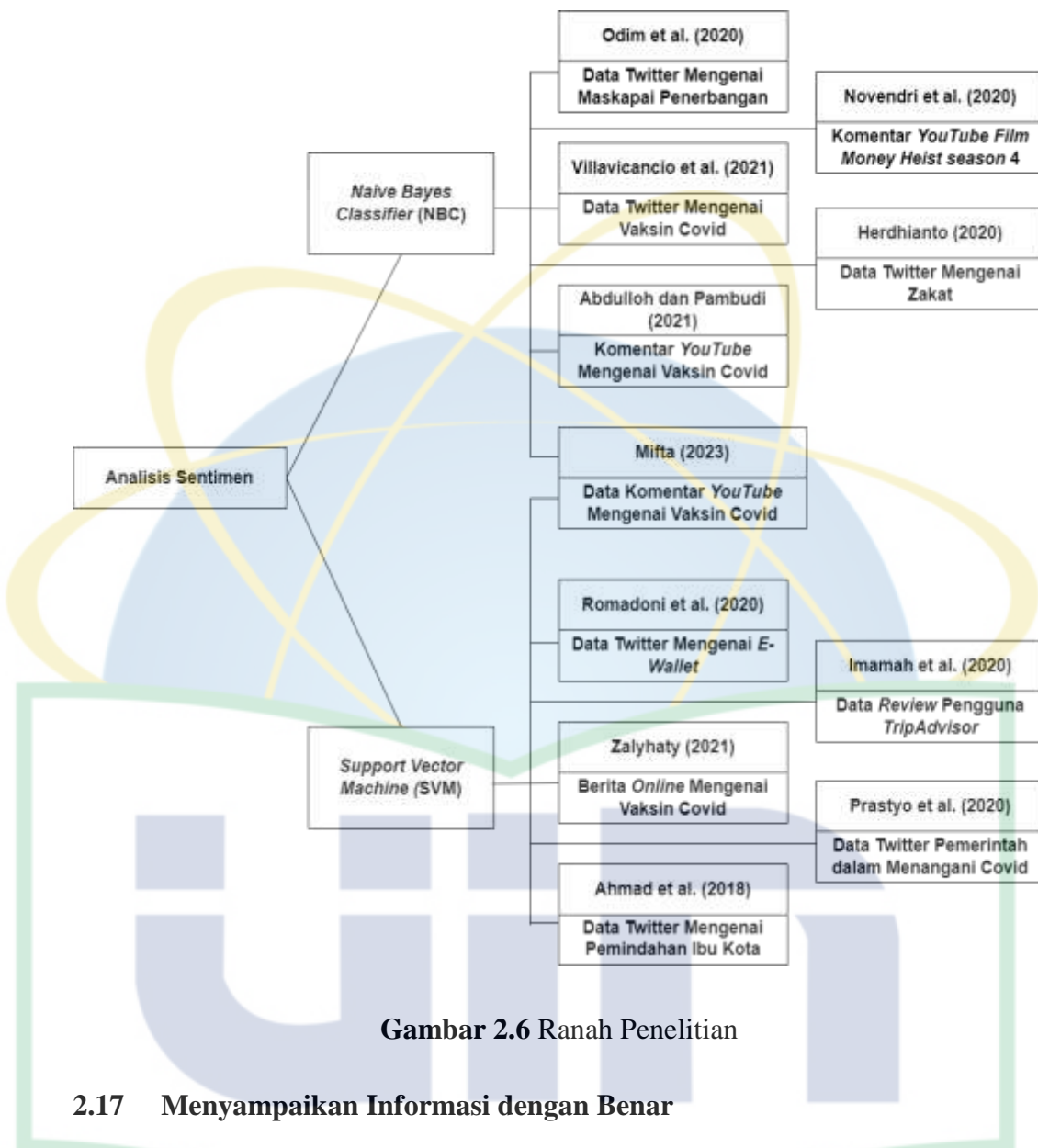
- b. Penggunaan data *testing* yang besar dibanding data *training* mempengaruhi akurasi algoritma *naive bayes classifier* lebih dari algoritma *support vector machine*.

H1:

- a. Penggunaan data *testing* yang kecil dibanding data *training* mempengaruhi akurasi algoritma *support vector machine* lebih dari algoritma *naive bayes classifier*.
- b. Penggunaan data *testing* yang besar dibanding data *training* mempengaruhi akurasi algoritma *support vector machine* lebih dari algoritma *naive bayes classifier*.

2.16 Ranah Penelitian

Pada tahap ini menggambarkan ranah penelitian sejenis yang dilakukan oleh penulis berdasarkan literatur yang penulis bandingkan berdasarkan topik penelitian mengenai analisis sentimen dengan 2 metode yaitu: *Naive Bayes Classifier* dengan *Support Vector Machine*. Berdasarkan penelitian sejenis sebelumnya, maka ranah penelitian yang penulis ambil adalah berkaitan dengan platform *YouTube* dengan mengambil data komentar dari video *YouTube* dengan topik berita mengenai Vaksinasi Covid-19 dengan membandingkan 2 metode utama yaitu *Naive Bayes Classifier* dengan *Support Vector Machine*. Gambar 2.6 adalah ilustrasi dari ranah penelitian.



Gambar 2.6 Ranah Penelitian

2.17 Menyampaikan Informasi dengan Benar

Media sosial mengajak siapa saja yang tertarik untuk berpartisipasi secara terbuka, memberi komentar, serta membagi informasi dalam waktu yang cepat dan tak terbatas. Kebebasan berpendapat merupakan hak setiap insan. Namun, berpendapat sering disalahgunakan untuk membuat fitnah, opini palsu, dan menebar kebencian yang sering diutarakan melalui media sosial (Maarif, 2021).

Dalam QS. Al-Hujurat ayat 6 disebutkan bagaimana etika serta tata cara menyikapi sebuah berita yang kita terima, sebagai berikut:

يَا أَيُّهَا الَّذِينَ آمَنُوا الَّذِينَ إِن جَاءَكُمْ فَاسِقٌ بِنَبَأٍ فَتَبَيَّنُوا أَن تُصِيبُوا قَوْمًا بِجَهْلَةٍ عَلَفْتُمْ صِحْرًا مَا مَفَعَلْتُ
نَدِيمِينَ

Artinya: “Hai orang-orang yang beriman, jika datang kepadamu orang fasik membawa suatu berita, maka periksalah dengan teliti agar kamu tidak menimpakan suatu musibah kepada suatu kaum tanpa mengetahui keadaannya yang menyebabkan kamu menyesal atas perbuatanmu itu.”

Islam mengajarkan opini yang jujur dan didasarkan pada bukti dan fakta serta diungkapkan dengan tulus. Tidak menyebarkan informasi yang belum diketahui kebenarannya di media sosial. Istilah ini disebut qaul zur yang berarti perkataan buruk atau kesaksian palsu (Maarif, 2021).

Firman Allah SWT, dalam QS. Al-Hajj ayat 30:

ذَلِكَ وَمَنْ يُعْظَمْ حُرْمَتِ اللَّهِ فَهُوَ خَيْرٌ لَهُ عِنْدَ رَبِّهِ وَأُحِلَّتْ لَكُمْ الْأَنْعَامُ مَا لَا عَلَيْكُمْ يُنْفِلُ يَبُوءَ أَفَاجَةً
جَسَ الرِّجْسَ مِنَ الْأَوْتَانِ وَاجْتَنِبُوا قَوْلَ الزُّورِ

Artinya: “Demikianlah (perintah Allah). Dan barang siapa mengagungkan apa yang terhormat di sisi Allah (hurumat) maka itu lebih baik baginya di sisi Tuhannya. Dan dihalalkan bagi kamu semua hewan ternak, kecuali yang diterangkan kepadamu (keharamannya), maka jauhilah olehmu (penyembahan) berhala-berhala yang najis itu dan jauhilah perkataan dusta.”

Pada kutipan, Juminem (2019), terdapat beberapa tuntunan dalam penggunaan media sosial sebagai berikut:

- Menyampaikan informasi dengan benar, tidak merekayasa atau memanipulasi fakta, serta menahan diri untuk tidak menyebarkan informasi tertentu yang fakta atau kebenarannya belum diketahui secara pasti.
- Menghindari prasangka *suudzon* atau buruk sangka, *gibah*, fitnah, dan *tajassus*. Prasangka yang tidak berdasar dapat membahayakan, karena dapat memicu *bullying* dan pembunuhan karakter.
- Meneliti fakta yang diperoleh agar tidak terjadi *gibah*, fitnah, dan *tajassus*.
- Menghindari *namimah* atau mengadu domba dan provokasi dengan pihak lain.
- Menghindari *Sukriyah* yang berarti merendahkan atau mengolok-ngolok orang lain yang dapat menumbuhkan kebencian.
- Bijak dalam bersosial media dengan mengedepankan etika, logika, dan perasaan serta berbagi nasihat yang baik, bijak, dan ikhlas.

BAB 3

METODE PENELITIAN

Pada bab ini akan dijelaskan mengenai tahapan-tahapan didalam metodologi yang digunakan dalam penelitian ini. Segala hal yang dilakukan dalam penelitian, mencakup tentang metodologi pengumpulan data, material eksperimen dan metode SEMMA yang menjelaskan mengenai proses dari algoritma yang digunakan dalam penelitian akan dijabarkan pada bab ini.

3.1 Metodologi Pengumpulan Data

a. Observasi

Kegiatan observasi ini dilakukan dalam upaya untuk mengumpulkan data dan informasi dengan melakukan peninjauan dan pengamatan secara langsung terhadap objek penelitian. Objek yang dimaksud adalah komentar pengguna *YouTube* terkait berita mengenai Vaksinasi Covid-19 pada *channel* Kompas TV dan tvOneNews.

b. Tinjauan Pustaka

Tinjauan Pustaka ini dilakukan dengan cara membaca dan mempelajari teori-teori yang terkait dalam pembahasan penelitian ini yang dilakukan sebelumnya untuk menunjang pelaksanaan penelitian. Dalam penelitian ini juga menggunakan referensi baik dari artikel jurnal, skripsi atau tesis dan *browsing internet* dalam upaya mendukung pemecahan masalah.

c. Studi Literatur Sejenis

Studi literatur sejenis dilakukan untuk mengumpulkan informasi dan data yang berhubungan dengan judul penelitian. Informasi dan data yang terkumpul

kemudian dijadikan data pembanding dan pendukung untuk penelitian yang sedang dilakukan.

3.2 Material

Material-material yang digunakan pada penelitian ini meliputi hal-hal berikut:

- a. Dataset yang digunakan merupakan hasil dari *crawling* pada kolom komentar pengguna masing-masing konten video di *channel* Kompas TV dan tvOneNews pada *YouTube* terkait berita Vaksinasi Covid-19 setiap bulannya dari bulan Maret 2021 sampai Desember 2021.
- b. Data untuk tahapan *Normalize* menggunakan kamus NLP yang bersumber dari GitHub, yang berisi kumpulan kata-kata informal menjadi formal berbahasa Indonesia.
- c. Data yang digunakan pada tahapan *Lexicon Based* menggunakan *Indonesian Sentiment (InSet) Lexicon* yang bersumber dari Koto dan Rahmanningtyas (2018) dan dimodifikasi dengan penambahan beberapa kata yang bersumber dari GitHub (Martua, 2020)

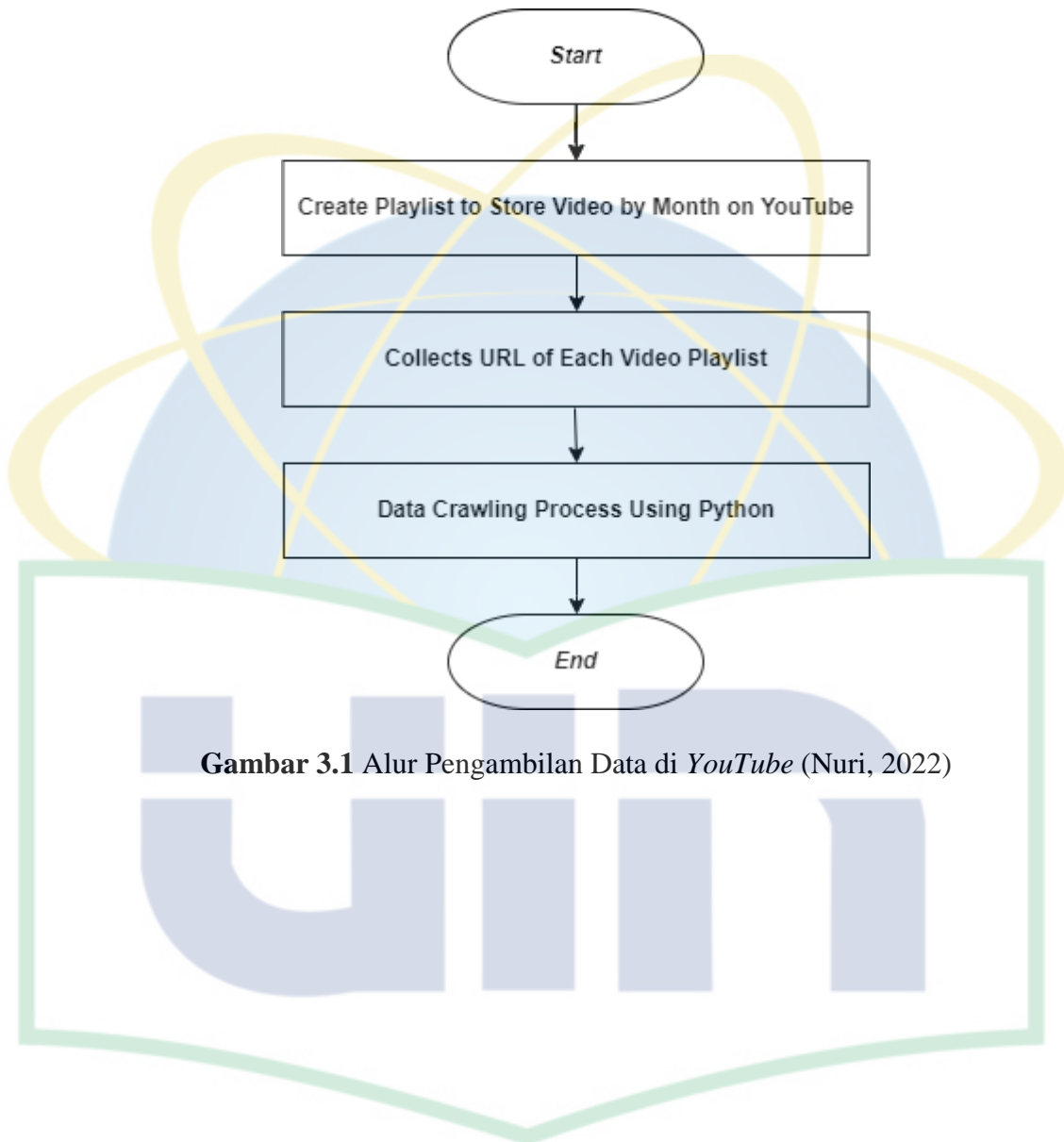
3.3 Metode SEMMA

Penelitian ini menggunakan metode SEMMA pada *data mining* yang terdiri atas *sample*, *explore*, *modify*, *model*, dan *assess*.

a. *Sample*

Pada tahap ini dilakukan pengumpulan data terkait tema dari penelitian terdahulu sebagai dasar dari penelitian ini. Kemudian dilakukan pengumpulan dataset yang bersumber dari *YouTube*. Dataset yang digunakan berupa tanggapan

pengguna di akun resmi *YouTube* Kompas TV dan tvOneNews terkait vaksinasi COVID-19 pada *channel* dari bulan Maret sampai Desember 2021.



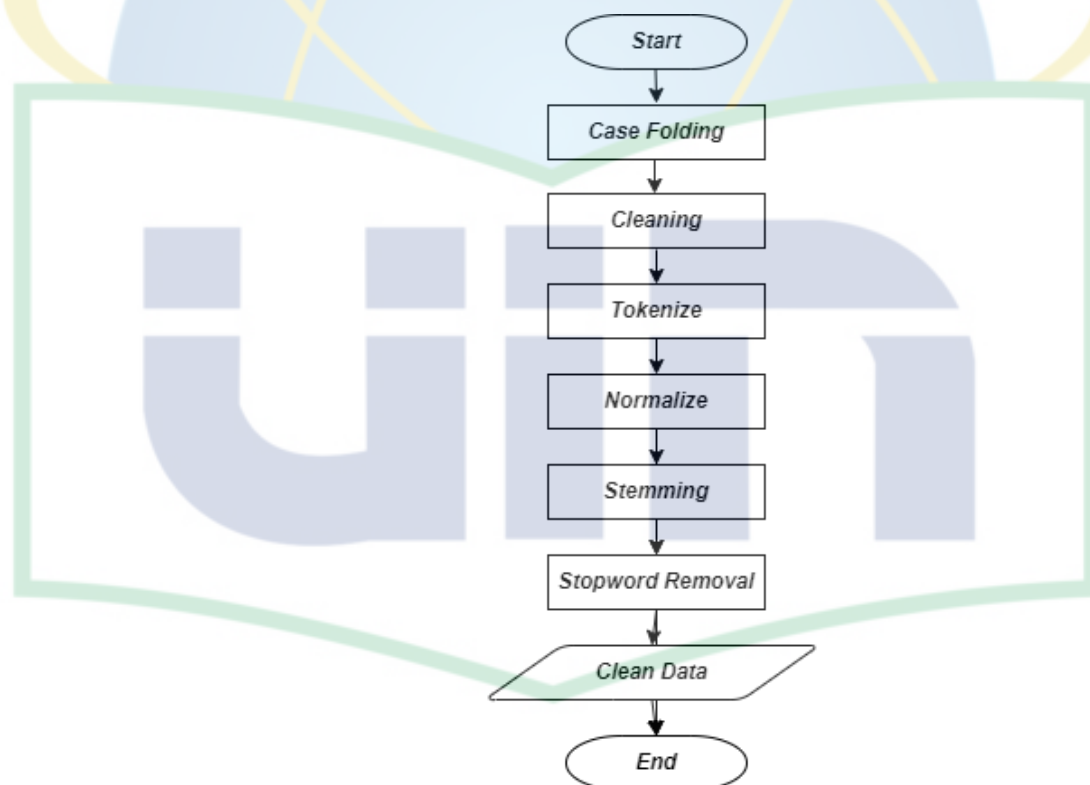
Gambar 3.1 Alur Pengambilan Data di *YouTube* (Nuri, 2022)

b. *Explore*

Pada tahap ini peneliti mendeskripsikan data yang didapat dari hasil *web scrapping* berupa tanggapan pengguna pengguna YouTube di akun Kompas TV dan tvOneNews terkait vaksinasi COVID-19 dari bulan Maret sampai Desember 2021.

c. *Modify*

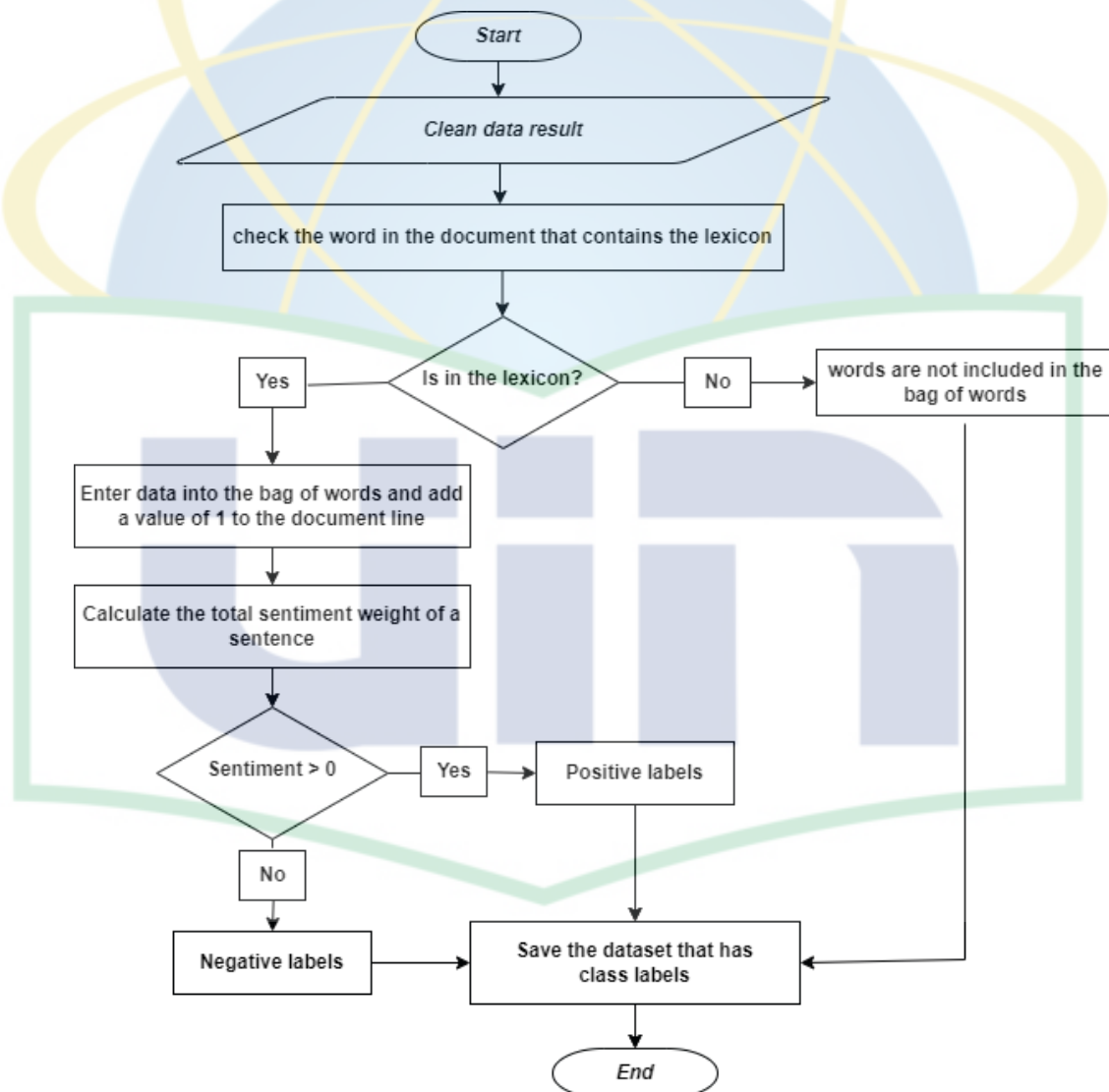
Pada tahap ini dilakukan persiapan data atau *preprocessing* data yang terdiri atas *case folding*, *cleaning*, *tokenize*, *normalize*, *stemming*, dan *stopword removal* agar data lebih terstruktur.



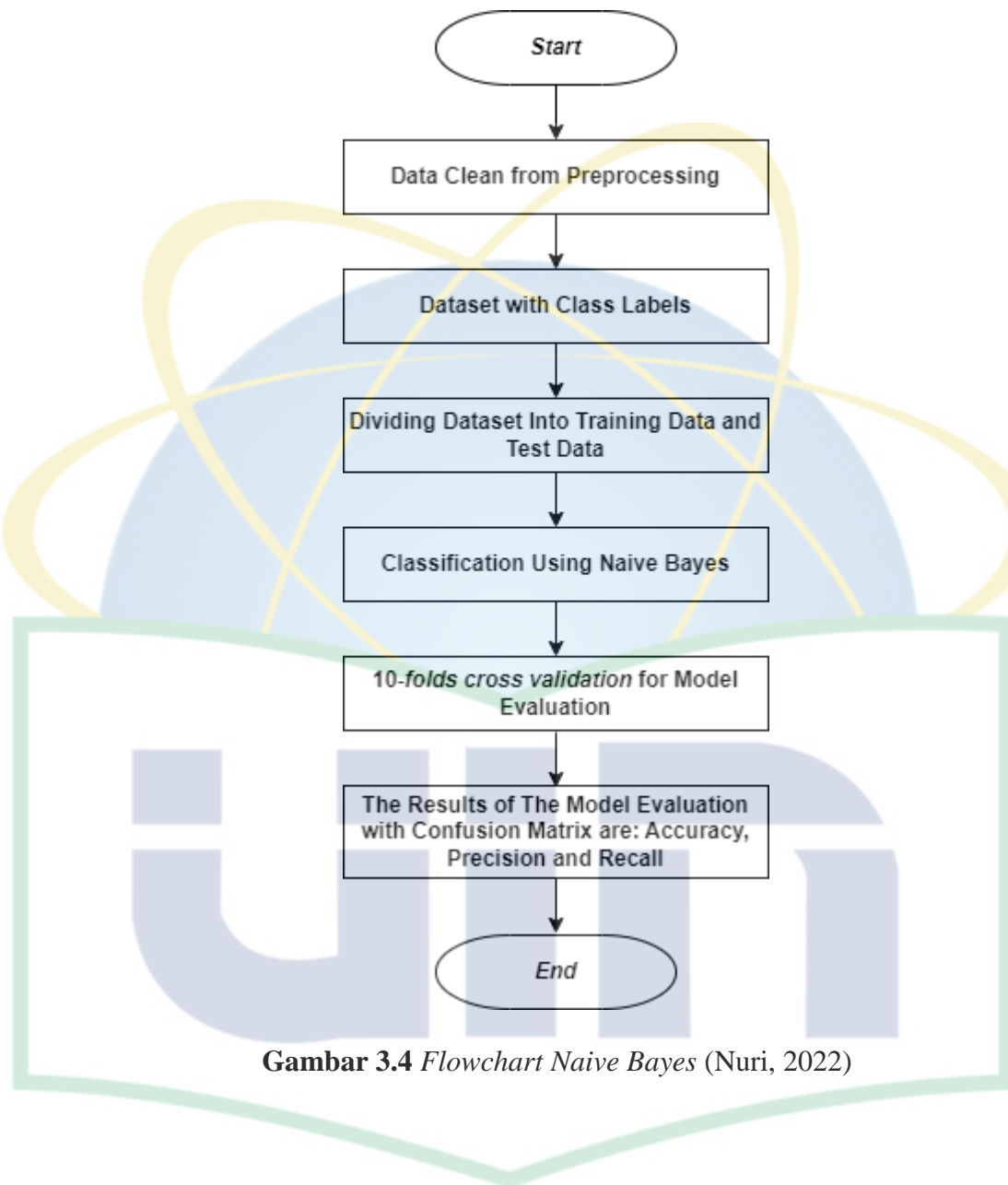
Gambar 3.2 Alur Tahap *Modify* (Nuri, 2022)

d. *Model*

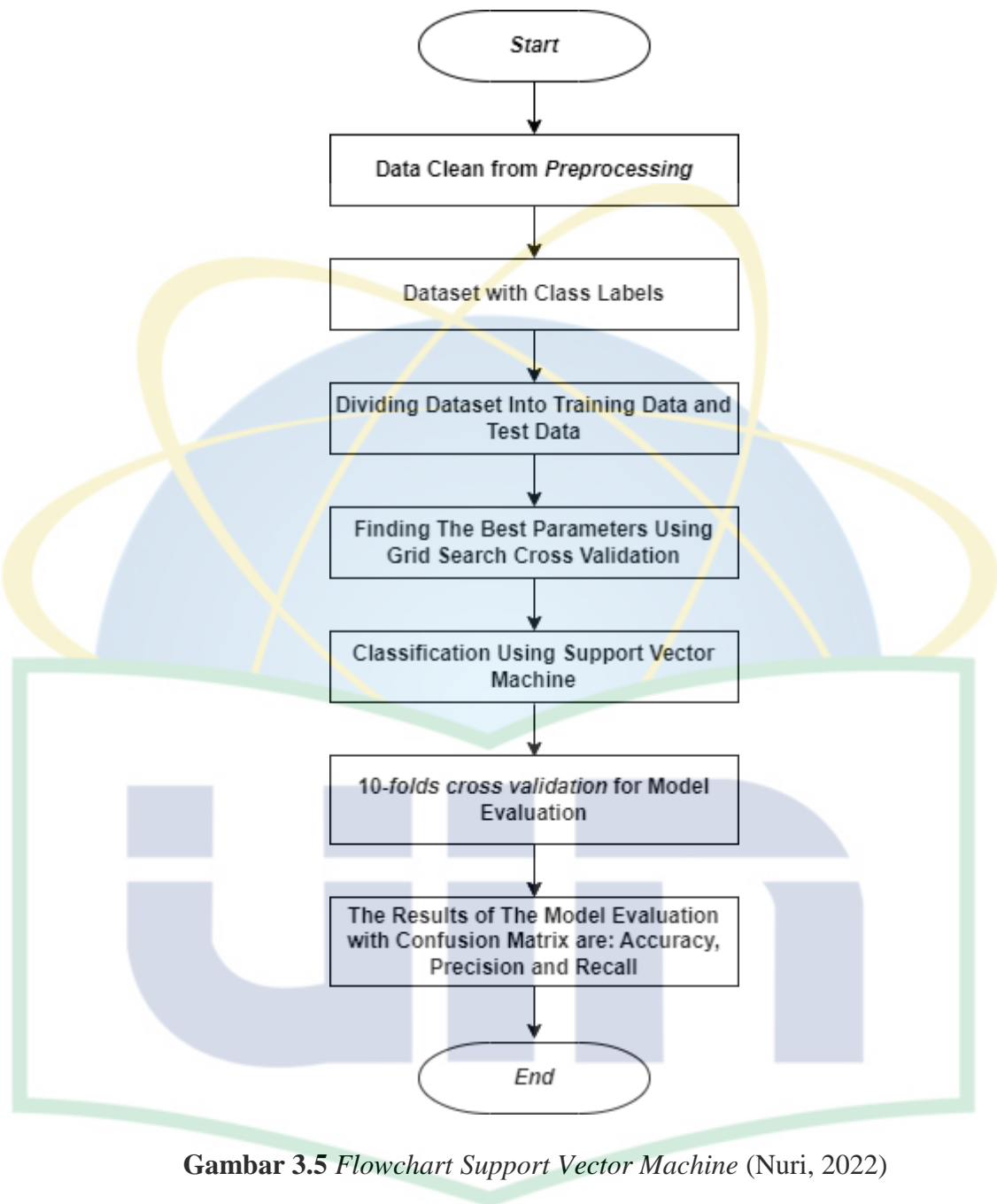
Memberikan label pada dataset bersih berdasarkan kelasnya untuk menentukan opini positif dan negatif dengan metode *lexicon*.. Tahap selanjutnya dataset dibagi menjadi data *train* dan data *test* untuk diolah dengan metode *Naive Bayes Classifier* dan *Support Vector Machine*. Gambar 3.3, 3.4 dan 3.5 adalah *flowchart* dari metode yang digunakan pada penelitian.



Gambar 3.3 *Flowchart Lexicon Based* (Nuri, 2022)



Gambar 3.4 Flowchart Naive Bayes (Nuri, 2022)



Gambar 3.5 Flowchart Support Vector Machine (Nuri, 2022)

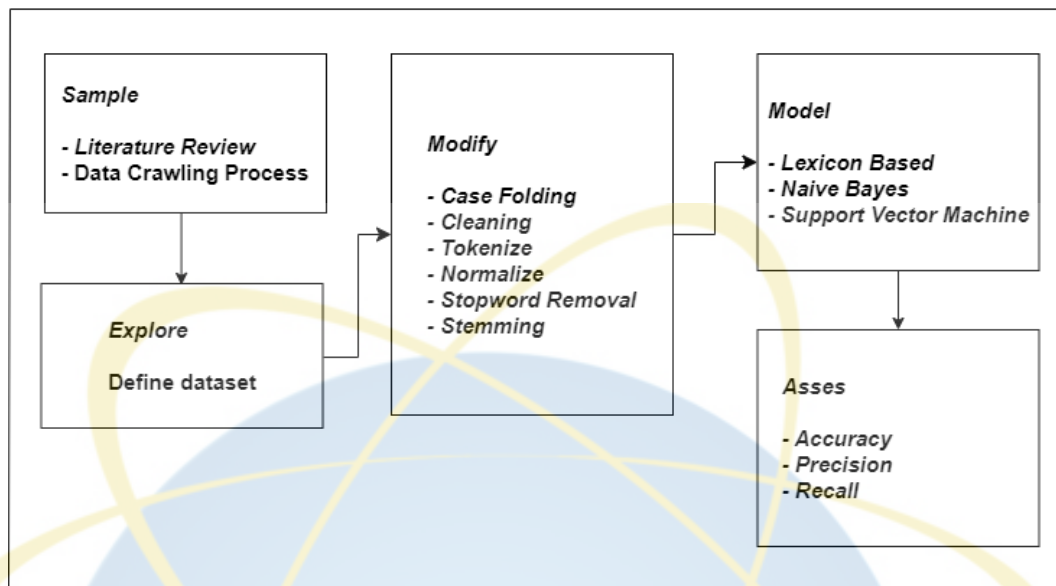
e. Assess

Pada tahap ini dilakukan evaluasi dari model. Pada metode *Naive Bayes Classifier* digunakan *10-folds cross validation* untuk mengetahui akurasi, presisi, dan *recall* dari model yang digunakan. Pada metode *Support Vector Machine*

digunakan *grid search cross validation* untuk mencari parameter terbaik dan *10-folds cross validation* untuk mengetahui akurasi, presisi, *recall* dari model yang digunakan.

3.4 Tahapan Metode SEMMA

Tahapan metode SEMMA yang terdiri atas *sample*, *explore*, *modify*, *modes* dan *asses*. Tahap *sample* penulis melakukan tinjauan pustaka dan *scraping* data pada *YouTube*. Tahap *explore* berupa menyeleksi atribut yang digunakan dan mendefinisikan data hasil *scraping* untuk dataset serta membuat grafik berdasarkan waktu dari data komentar *YouTube*. Tahap *modify* dilakukan pada data tidak terstruktur yang terdiri atas *case folding*, *cleaning*, *tokenize*, *normalize*, *stemming*, dan *stopword removal* agar menjadi data yang lebih terstruktur. Tahap *model* dilakukan pengolahan dataset terstruktur, dimana data set diberikan label kelas dengan metode *lexicon* kemudian diolah dengan metode *Naive Bayes* dan *Support Vector Machine*. Tahap *asses* mengevaluasi penilaian terhadap pemodelan berupa akurasi, presisi, dan *recall*. Setelah itu, tahapan terakhir adalah kesimpulan dan saran penelitian. Tahapan penelitian ini dijelaskan pada Gambar 3.6.



Gambar 3.6 Tahapan Metode SEMMA (Afifi, 2022)

3.5 Perangkat Penelitian

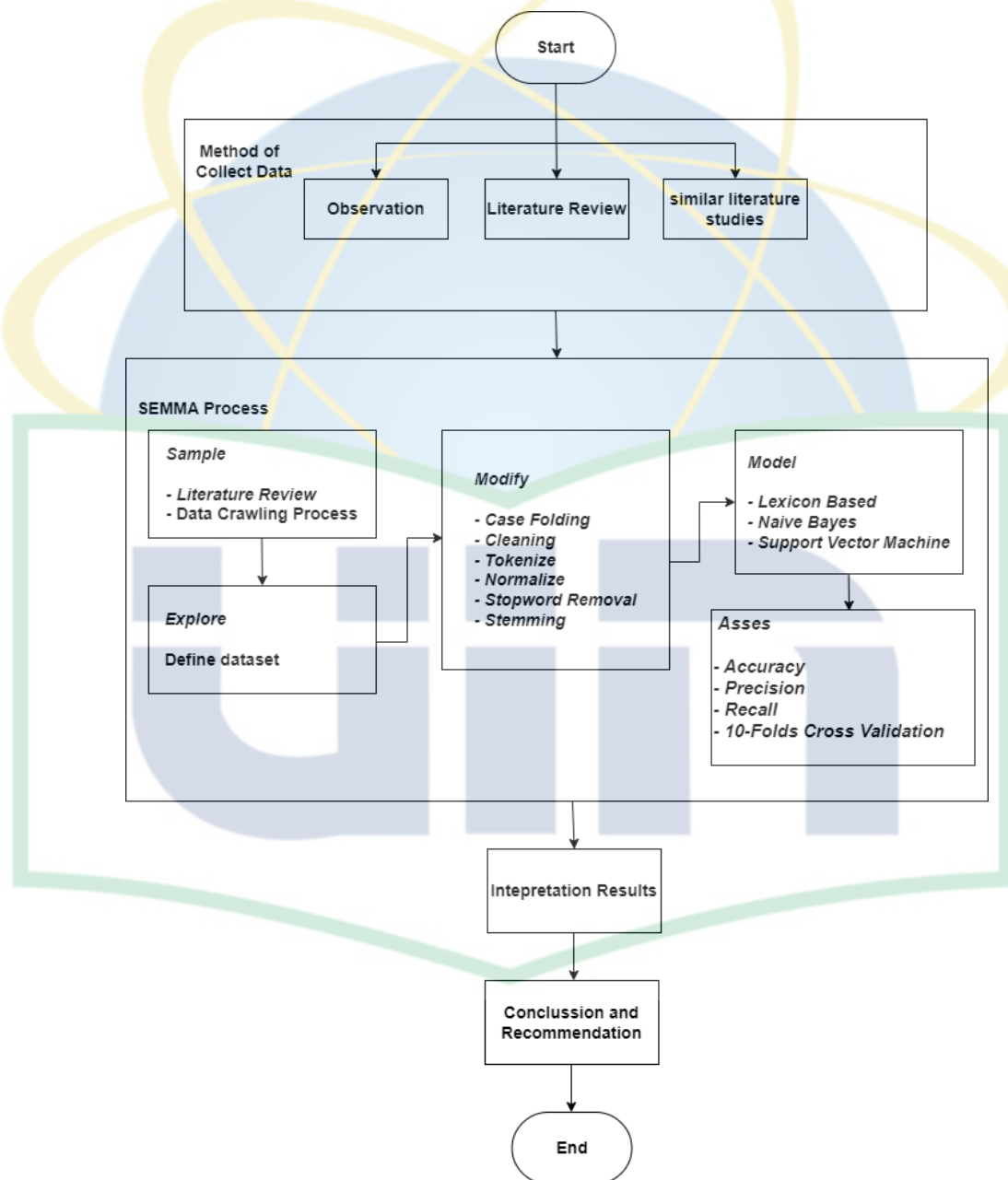
Spesifikasi perangkat keras (*hardware*) dan perangkat lunak (*Software*) yang digunakan oleh peneliti dalam penelitian ini Tabel 3.1.

Tabel 3.1 *Software dan Hardware*

Hardware	Lenovo Thinkpad X240	Intel Core i5 4200U speed 1.6 GHz 2.6 GHz
		4 GB RAM
		125 SSD dan 500 HDD
		Monitor 12,5 inch
Software	Sistem Operasi	Windows 10
	Tools	Google Colaboratory
	Bahasa Pemrograman	Python 3.9.13

3.6 Alur Penelitian

Alur penelitian yang menjelaskan proses berjalannya penelitian ini yang dimulai dari tahapan metode pengumpulan data, material penelitian dan metode SEMMA. Dalam penelitian ini, peneliti mengacu pada Gambar 3.7 berikut.



Gambar 3.7 Alur Penelitian

BAB 4

HASIL DAN PEMBAHASAN

4.1 *Sample*

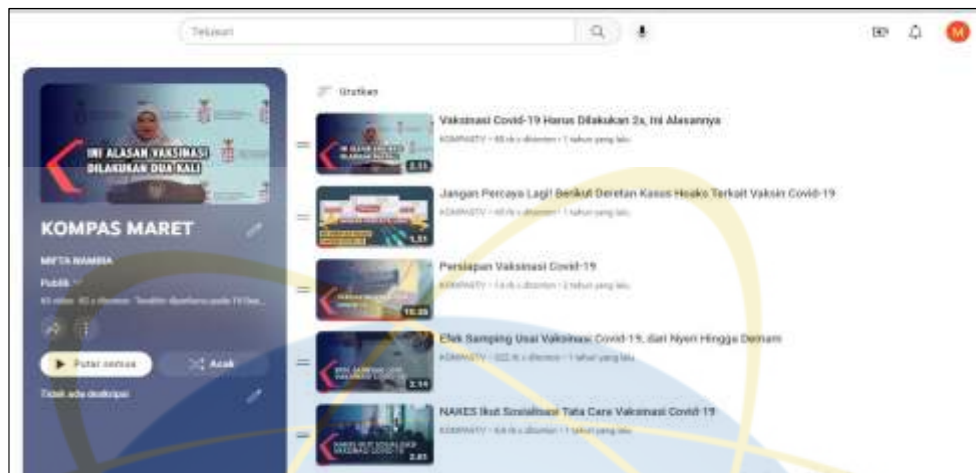
4.1.1 Tinjauan Pustaka

Peneliti melakukan tinjauan pustaka terkait analisis sentimen, metode *Naive bayes* dan *Support Vector Machine* dari jurnal-jurnal yang dipaparkan pada tabel 2.4.

4.1.2 *Crawling Data*

Sumber dataset dalam penelitian ini diperoleh dari platform media sosial *YouTube* dengan menggunakan teknik *web crawling* untuk memperoleh data tekstual berupa komentar pengguna terhadap konten video yang dilihatnya. Selain mendapatkan komentar, beberapa data penunjang juga diperoleh pada tahapan ini seperti nama pengguna, nama *channel*, dan tanggal publikasi. Berikut langkah-langkah untuk *crawling* data di *YouTube*:

- Melakukan penyeleksian video secara manual dan memasukkannya ke dalam playlist berdasarkan *channel* serta tanggal pempublikasian video tersebut pada masing-masing bulan. Contohnya dapat dilihat pada gambar 4.1



Gambar 4.1 Playlist Konten Video Berdasarkan Bulan Publikasi

- Setelah dilakukan pembuatan *playlist*, kemudian mengumpulkan URL dari masing-masing konten video dari bulan Maret 2021 sampai Desember 2021. Pengumpulan URL ini bertujuan agar mempermudah proses *crawling*. Tabel 4.1 merupakan contoh pengumpulan URL konten video.

Tabel 4.1 Contoh URL masing-masing konten video

No.	URL	Bulan
1	https://youtu.be/3oOCz3LfwvQ	Maret
2	https://youtu.be/bMJHm2yN3mk	
3	https://youtu.be/0eIYES3pvqM	
4	https://youtu.be/TQjbgV7wgFo	
5....	https://youtu.be/o0GgpGBbXEA	
28	https://youtu.be/PbGGTg8iAyU	APRIL
29	https://youtu.be/Rawca_9e5t4	
30	https://youtu.be/buKkLnCZcqE	
31....	https://youtu.be/CXgcr71e1bQ	
40	https://youtu.be/lb_Rdfj-NMI	MAY
41	https://youtu.be/iCIBfIVyNzA	
42....	https://youtu.be/wRCKZzS33To	

- Setelah dilakukan URL dari *playlist* terkumpul, kemudian dilakukan proses pengambilan data secara otomatis dengan bahasa pemrograman *python*. Hasil dari pengambilan data tersebut berupa file dalam format *Json* secara terpisah dari masing-masing bulan. Kemudian dikonversi dari *JSON* menjadi *data frame* dan menyimpan hasilnya ke dalam file *CSV* agar lebih mudah untuk diolah diproses selanjutnya. Adapun contoh dari hasil pengambilan data yang berhasil diperoleh dapat dilihat pada tabel 4.1.

Tabel 4.2 Contoh Hasil Pengambilan Data

komentar	time	author	Channel	heart	reply
gimana bisa imunnya terbentuk jika makan sehari hari susah dengan kondisi beginimikirrr	1 tahun yang lalu	aby ardiansy ah	UCgI3V wH8Dyo y3ilVW7 6hjEg	FALSE	FALSE
kebanyakan orang yg sdh vaksin jadi over confidenceso kadang kadang lupa prokes lagian jenis vaksin yg sekarang toh jg gak efektif terhadap varian baru yg akan dtg lagian jg percuma setelah vaksin utk masyarakat miskin gizi nya ga bisa terpenuhi akibat kebutuhan hidup yg pas pasan	1 tahun yang lalu (diedit)	Ridwan Hidayat	UCgI3V wH8Dyo y3ilVW7 6hjEg	FALSE	FALSE

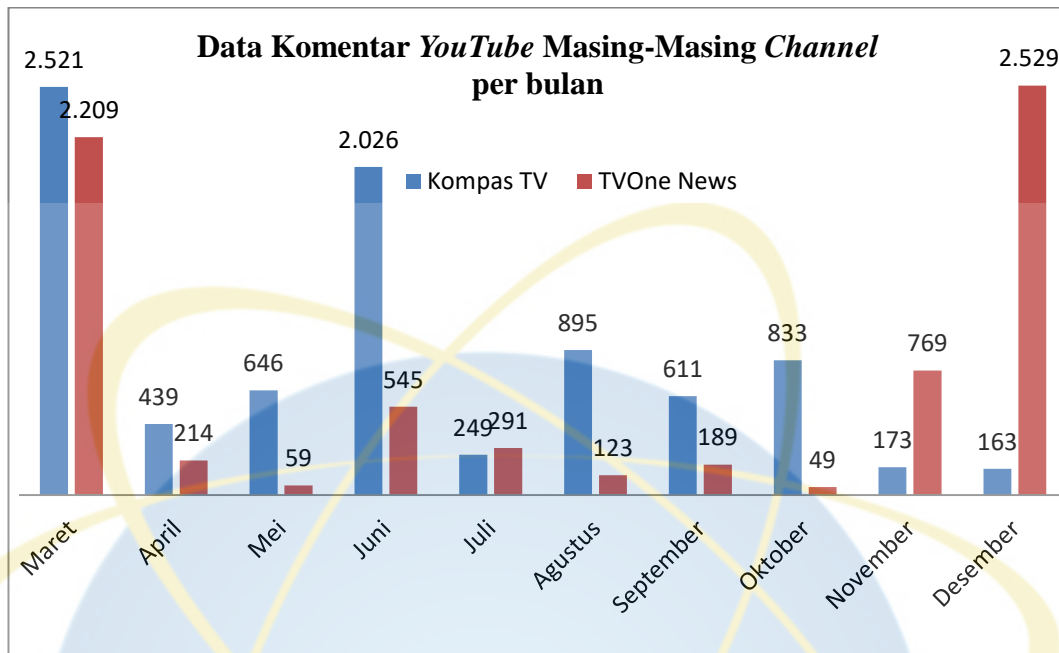
4.2 Explore

Hasil dari *crawling* berisi 5 kolom mengenai rincian dari sebuah komentar video yaitu komentar merupakan isi tanggapan dari pengguna *YouTube*, *time* menunjukkan waktu pengguna dalam memberikan komentar terhadap video, *author* merupakan nama pengguna yang memberikan komentar, *channel* merupakan *url channel* dari video yang dikomentari pengguna, *heart* merupakan jumlah *like* yang diberikan pada komentar pengguna oleh pengguna lainnya, *reply* merupakan jumlah pengguna lain yang memberikan balasan pada komentar video yang bersangkutan. Agar mempermudah dalam menganalisis data, maka hanya atribut komentar saja yang ditampilkan Gambar 4.2.

Tabel 4.3 Hasil filter kolom pada dataset

komentar
gimana bisa imunnya terbentuk jika makan sehari hari susah dengan kondisi beginimikirrr
kebanyakan orang yg sdh vaksin jadi over confidenceso kadang kadang lupa prokes lagian jenis vaksin yg sekarang toh jg gak efektif terhadap varian baru yg akan dtg lagian jg percuma setelah vaksin utk masyarakat miskin gizi nya ga bisa terpenuhi akibat kebutuhan hidup yg pas pasan

Dataset diurutkan berdasarkan bulan untuk selanjutnya dibuat grafik seperti pada Gambar 4.2. Berdasarkan konten video mengenai berita vaksinasi covid-19 yang di *crawling* pada *YouTube*, komentar terbanyak ada pada bulan Desember di *channel* tvOneNews yaitu sebanyak 2.529 komentar.



Gambar 4.2 Data Komentar Berdasarkan Waktu

4.3 *Modify*

Modify dilakukan pada data set berupa *text preprocessing*. Tahapan ini bertujuan agar dataset menjadi lebih terstruktur dan lebih dikenali bentuknya oleh sistem komputer untuk diolah lebih lanjut. Tahap *text preprocessing* yang digunakan terdiri atas *case folding*, *cleaning*, *tokenize*, *stopword removal*, dan *stemming*.

4.3.1 *Case Folding*

Langkah pertama dalam mengolah data tidak terstruktur menjadi data terstruktur adalah *case folding* dimana dilakukan perubahan huruf besar pada dataset menjadi huruf kecil. Berikut merupakan *script* yang digunakan pada tahap *Case Folding* serta hasil *Case Folding* yang dapat dilihat pada Tabel 4.3.

```
# ----- Case Folding -----
# gunakan fungsi Series.str.lower() pada Pandas
TWEET_DATA['komentar'] = TWEET_DATA['komentar'].str.lower()
print('Case Folding Result : \n')
print(TWEET_DATA['komentar'])
print('\n\n\n\n')
```

Tabel 4.4 Hasil *Case Folding*

Sebelum	Sesudah
Setelah Vaksin Pertama, Kena Covid 19, Kemudian Divaksin Lagi, Kalo Ternyata Masih Terkena Positif Covid19, Lalu Harus Vaksin Lagi Kalau Mau Sehat, Dan Seterusnya.... Terimakasih,	setelah vaksin pertama, kena covid19, kemudian divaksin lagi, kalo ternyata masih terkena positif covid19, lalu harus di vaksin lagi kalau mau sehat, dan seterusnya.... terimakasih,

4.3.2 *Cleaning*

Pada proses ini menghapus atribut yang tidak dibutuhkan dan tidak relevan untuk pengklasifikasian data. Atribut yang di antaranya *hashtag*, *URL*, *link*, tanda baca, angka, *ascii*, dan *white space*. Berikut merupakan *script* yang digunakan pada tahap *Cleaning* serta hasil *Cleaning* yang dapat dilihat pada Tabel 4.4.

```
# ----- Cleaning -----
def remove_komentar_special(text):
    # remove tab, new line, and back slice
    text = text.replace('\t', " ").replace('\n', " ")
    .replace('\u', " ").replace('\\', "")
    # remove non ASCII (emoticon, chinese word, .etc)
    text = text.encode('ascii', 'replace').decode('ascii')
    # remove mention, link, hashtag
    text = ' '.join(re.sub("([@#][A-Za-z0-9]+)|(\w+:\/\/\S+)", " ", text).split())
```

```

#remove number
def remove_number(text):
    return re.sub(r"\d+", "", text)

#remove punctuation
def remove_punctuation(text):
    return text.translate(str.maketrans("", "", string.punctuation))

#remove whitespace leading & trailing
def remove_whitespace_LT(text):
    return text.strip()

#remove multiple whitespace into single whitespace
def remove_whitespace_multiple(text):
    return re.sub('\s+', ' ', text)

# remove single char
def remove_singl_char(text):
    return re.sub(r"\b[a-zA-Z]\b", "", text)

```

Tabel 4.5 Tabel hasil *cleaning*

Sebelum	Sesudah
setelah vaksin pertama, kena covid19, kemudian divaksin lagi, kalo ternyata masih terkena positif covid19, lalu harus di vaksin lagi kalau mau sehat, dan seterusnya, terimakasih,	setelah vaksin pertama kena covid kemudian divaksin lagi kalo ternyata masih terkena positif covid lalu harus di vaksin lagi kalau mau sehat dan seterusnya terimakasih

4.3.3 Tokenize

Selanjutnya proses *tokenize* merupakan tahapan yang bertujuan untuk memecah kata menjadi token dari masing-masing komentar berdasarkan karakter spasi sebagai pemisah. *Tokenize* ini bertujuan agar data dapat diproses pada

tahapan selanjutnya. Berikut merupakan *script* yang digunakan pada tahap *Tokenize* serta hasil *Tokenize* yang dapat dilihat pada Tabel 4.5.

```
# NLTK word tokenize
def word_tokenize_wrapper(text):
    return word_tokenize(text)

TWEET_DATA['komentar_tokens'] = TWEET_DATA['komentar']
    .apply(word_tokenize_wrapper)
```

Tabel 4.6 Komentar Sebelum Dan Sesudah Dilakukan Proses *Tokenize*

Sebelum	Sesudah
setelah vaksin pertama kena covid kemudian divaksin lagi kalo ternyata masih terkena positif covid lalu harus di vaksin lagi kalau mau sehat dan seterusnya terimakasih	['setelah', 'vaksin', 'pertama', 'kena', 'covid', 'kemudian', 'divaksin', 'lagi', 'kalo', 'ternyata', 'masih', 'terkena', 'positif', 'covid', 'lalu', 'harus', 'di', 'vaksin', 'lagi', 'kalau', 'mau', 'sehat', 'dan', 'seterusnya', 'terimakasih']

4.3.4 *Normalize*

Normalize merupakan tahap dimana dilakukan standarisasi kata yang memiliki makna sama dengan melakukan perubahan penulisan pada suatu kata singkatan atau tidak baku agar memiliki arti kata yang seragam. Berikut langkah yang dilakukan dalam proses *normalize*.

1. Menyiapkan terlebih dahulu kamus yang akan digunakan sebagai kamus normalisasi. Pada penelitian ini menggunakan kamus normalisasi dari kamus NLP Bahasa Indonesia *Resource* di Github. Kamus ini berisi ribuan kosa kata

slang words dan merubahnya ke dalam kata-kata formal sebagai acuan proses normalisasi.

2. Melakukan proses normalisasi dengan mengganti kata tidak baku atau *slang* berdasarkan kamus yang telah diinput.

bntr	: bentar	smpe	: sampai
ajh	: saja	tnya	: tanya
beud	: sangat	lgsg	: langsung
bsa	: bisa	knp	: kenapa
bsk	: besok	cuan	: uang
skrg	: sekarang	kaga	: tidak
yg	: yang	udh	: sudah
gmna	: bagaimana	sndri	: sendiri
pkai	: pakai	krn	: karena
stiap	: setiap	klo	: kalau

Gambar 4.3 Contoh Normalisasi Kata

Berikut merupakan *script* yang digunakan pada tahap Normalisasi serta hasil Normalisasi yang dapat dilihat pada Tabel 4.6.

```

normalized_word = pd.read_excel("/content/kamusnlp.xlsx")

normalized_word_dict = {}

for index, row in normalized_word.iterrows():
    if row[0] not in normalized_word_dict:
        normalized_word_dict[row[0]] = row[1]

def normalized_term(document):
    return [normalized_word_dict[term] if term in normalized_word_dict else term for term in document]
```

```
TWEET_DATA['komentar_normalized'] = TWEET_DATA['komentar_tokens_WSW'].apply(normalized_term)
```

Tabel 4.7 Komentar Sebelum Dan Sesudah Dilakukan Proses Normalisasi

Sebelum	Sesudah
['setelah', 'vaksin', 'pertama', 'kna', 'covid', 'kemudian', 'divaksin', 'lagi', 'klo', 'ternyata', 'masih', 'terkena', 'positif', 'covid', 'lalu', 'harus', 'di', 'vaksin', 'lagi', 'klo', 'mau', 'sehat', 'dan', 'seterusnya', 'terimakasih']	['setelah', 'vaksin', 'pertama', 'kena', 'covid', 'kemudian', 'divaksin', 'lagi', 'kalau', 'ternyata', 'masih', 'terkena', 'positif', 'covid', 'lalu', 'harus', 'di', 'vaksin', 'lagi', 'kalau', 'mau', 'sehat', 'dan', 'seterusnya', 'terimakasih']

4.3.5 Stopword Removal

Stopword removal adalah tahapan selanjutnya dalam *text-preprocessing* untuk menghilangkan kata-kata yang tidak memiliki makna dan tidak memberikan pengaruh sentimen pada suatu kalimat. Proses *stopword* yang digunakan pada penelitian ini memanfaatkan *library* Sastrawi yang di dalamnya terdapat *corpus stopwords* bahasa indonesia. Penentuan kata-kata pada kamus *stopwords* juga dapat ditambahkan sesuai kebutuhan. Berikut merupakan *script* yang digunakan pada tahap *Stopword Removal* serta hasil *Stopword Removal* yang dapat dilihat pada Tabel 4.7.

```
from nltk.corpus import stopwords

# get stopwords indonesia
list_stopwords = stopwords.words('indonesian')

# append additional stopword
list_stopwords.extend("/content/slangword.xlsx")
```

```
def stopwords_removal(words):
    return [word for word in words if word not in list_stopwords]

TWEET_DATA['komentar_tokens_WSW'] = TWEET_DATA['komentar_tokens'].apply(stopwords_removal)

print(TWEET_DATA['komentar_tokens_WSW'].head())
```

Tabel 4.8 Komentar Sebelum Dan Sesudah Dilakukan Proses *Stopword Removal*

Sebelum	Sesudah
['setelah', 'vaksin', 'pertama', 'kena', 'covid', 'kemudian', 'divaksin', 'lagi', 'kalau', 'ternyata', 'masih', 'terkena', 'positif', 'covid', 'lalu', 'harus', 'di', 'vaksin', 'lagi', 'kalau', 'mau', 'sehat', 'dan', 'seterusnya', 'terimakasih']	['setelah', 'vaksin', 'pertama', 'kena', 'covid', 'divaksin', 'masih', 'positif', 'covid', 'vaksin', 'mau', 'sehat', 'terus', 'terimakasih']

4.3.6 Stemming

Pada tahapan *stemming* ini bertujuan untuk mengubah dan menyeragamkan kata-kata yang ada pada dokumen dataset dengan melakukan pemetaan dari bentuk varian kata yang memiliki imbuhan didalamnya, baik itu awalan, sisipan, akhiran maupun kombinasi dari awalan dan akhiran. *Stemming* pada dataset berbahasa Indonesia dilakukan dengan memanfaatkan *library* Sastrawi yang ada pada bahasa pemrograman *python*. Berikut merupakan *script* yang digunakan pada tahap *Stemming* serta hasil *Stemming* yang dapat dilihat pada Tabel 4.8.

```

# import Sastrawi package
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
import swifter

# create stemmer
factory = StemmerFactory()
stemmer = factory.create_stemmer()

# stemmed
def stemmed_wrapper(term):
    return stemmer.stem(term)

term_dict = {}

for document in TWEET_DATA['komentar_normalized']:
    for term in document:
        if term not in term_dict:
            term_dict[term] = ' '

print(len(term_dict))
print("-----")

for term in term_dict:
    term_dict[term] = stemmed_wrapper(term)
    print(term, ":", term_dict[term])

print(term_dict)
print("-----")

# apply stemmed term to dataframe
def get_stemmed_term(document):
    return [term_dict[term] for term in document]

TWEET_DATA['komentar_tokens_stemmed'] = TWEET_DATA['komentar_normalized'].swifter.apply(get_stemmed_term)
print(TWEET_DATA['komentar_tokens_stemmed'])

```

Tabel 4.9 Komentar Sebelum dan Sesudah Dilakukan Proses Stemming.

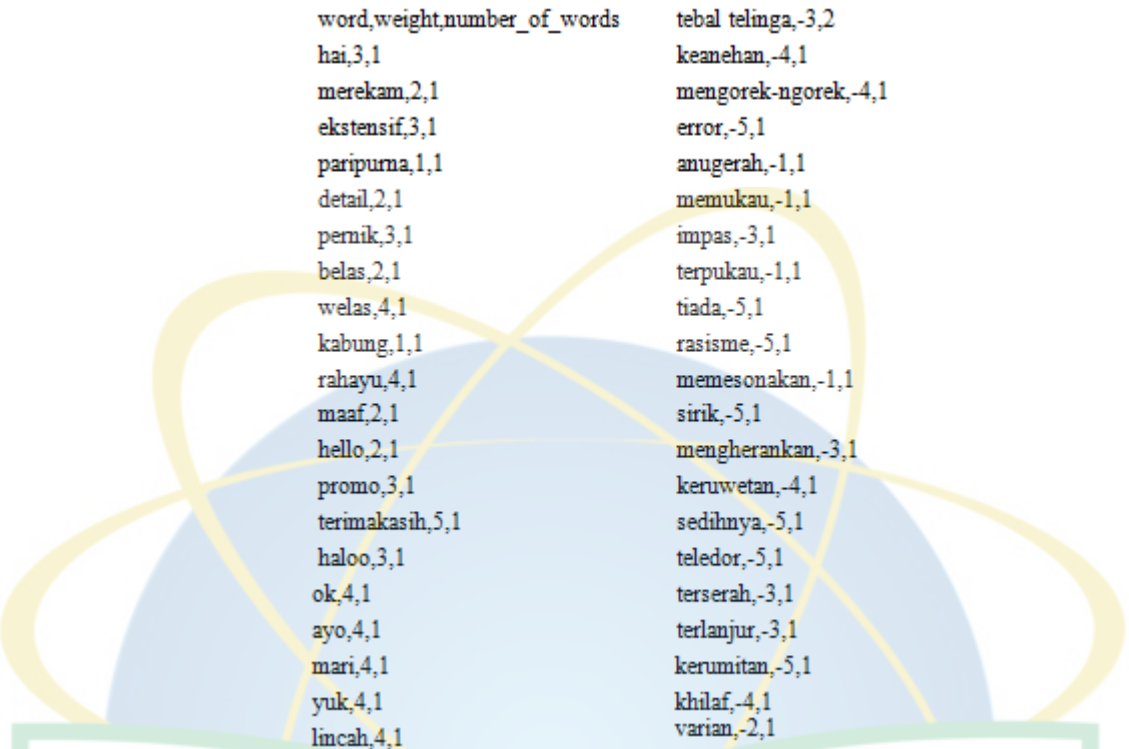
Sebelum	Sesudah
['setelah', 'vaksin', 'pertama', 'terkena', 'covid', 'divaksin', 'masih', 'positif', 'covid', 'vaksin', 'mau', 'sehat', 'seterusnya', 'terimakasih']	['setelah', 'vaksin', 'pertama', 'terkena', 'covid', 'vaksin', 'masih', 'positif', 'covid', 'vaksin', 'mau', 'sehat', 'terus', 'terimakasih']

4.4 Model

Tahap pemodelan pada penelitian ini menggunakan tiga model yaitu metode *lexicon based* untuk memberikan label kelas pada dataset, selanjutnya digunakan metode *Naive Bayes* dan *Support Vector Machine* untuk klasifikasi.

4.4.1 Lexicon Based

Metode *lexicon based* pada penelitian ini digunakan untuk melakukan klasifikasi kelas pada dataset. Suatu dokumen pada dataset diklasifikasikan ke dalam dua kelas, yaitu positif dan negatif. Kata-kata yang ada pada suatu komentar akan dibandingkan dengan dokumen pada kamus *lexicon*. Jika kata dalam komentar terdapat di dalam kamus *lexicon*, maka kata tersebut akan diberikan nilai *score*. Jumlah nilai *score* pada suatu komentar akan menentukan label positif atau negatif. Kamus *lexicon* yang digunakan adalah *Indonesian Sentiment (InSet) Lexicon* yang bersumber dari Koto dan Rahmaningtyas (2018). *InSet lexicon* berisi kurang lebih 10.250 kata yang diberi nilai -5 sampai dengan +5. *InSet lexicon* kemudian dimodifikasikan dengan penambahan beberapa kata yang bersumber dari GitHub (Martua, 2020). Gambar 4.2 adalah beberapa kata sentimen dalam kamus *lexicon*.



word,weight,number_of_words	tebal telinga,-3,2
hai,3,1	keanehan,-4,1
merekam,2,1	mengorek-ngorek,-4,1
ekstensif,3,1	error,-5,1
paripurna,1,1	anugerah,-1,1
detail,2,1	memukau,-1,1
pernik,3,1	impas,-3,1
belas,2,1	terpukau,-1,1
welas,4,1	tiada,-5,1
kabung,1,1	rasisme,-5,1
rahayu,4,1	memesonakan,-1,1
maaf,2,1	sirik,-5,1
hello,2,1	mengherankan,-3,1
promo,3,1	keruwetan,-4,1
terimakasih,5,1	sedihnya,-5,1
haloo,3,1	teledor,-5,1
ok,4,1	terserah,-3,1
ayo,4,1	terlanjur,-3,1
mari,4,1	kerumitan,-5,1
yuk,4,1	khilaf,-4,1
lincah,4,1	varian,-2,1

Gambar 4.4 Kamus *Lexicon*

Berikut merupakan *script* yang digunakan pada metode *Lexicon* serta hasil sentimen dengan menggunakan *Lexicon* dapat dilihat pada Tabel 4.9.

```
#lexiconbased
!pip install VaderSentiment

from vaderSentiment.vaderSentiment import
SentimentIntensityAnalyzer
analyser = SentimentIntensityAnalyzer()

df = pd.read_excel('/content/siaplexicon.xlsx') #data
crawling
df.head()

scores = [analyser.polarity_scores(x) for x in
df['komentar']]
print(scores)
df['Compound_Scores'] = [x['compound'] for x in scores]
```



```

df.nsmallest(10, ['Compound_Scores']) #melihat score
terkecil

df.nlargest(10, ['Compound_Scores']) #melihat score
terbesar

df.loc[df['Compound_Scores'] < 0, 'Sentiments'] =
'Negative'
df.loc[df['Compound_Scores'] > 0, 'Sentiments'] =
'Positive'
df.head()

```

Tabel 4.10 Hasil Sentimen Komentar dengan Metode *Lexicon*

Sentimen	Jumlah
Positif	1327
Negatif	853

Tabel 4.9 merupakan hasil dari pemberian label dengan menggunakan metode *lexicon based*. Didapatkan bahwa sebanyak 1327 data komentar masuk ke dalam label sentimen positif, dan sebanyak 853 data komentar masuk ke dalam label sentimen negatif. Berdasarkan hasil pemberian label dengan menggunakan metode *lexicon based* didapatkan bahwa sentimen pengguna *YouTube* pada bulan Maret 2021 sampai desember 2021 terhadap vaksinasi covid-19 adalah positif.

Dataset yang sudah diklasifikasikan ke dalam kelas positif dan negatif selanjutnya divisualisasikan dalam bentuk *wordcloud*. Gambar 4.5 merupakan frekuensi kemunculan kata dari kelas positif dan Gambar 4.6 merupakan visualisasi *wordcloud* dari kelas positif.

Pada tahap ini dataset dibagi menjadi dua bagian, yaitu data latih dan data uji dengan beberapa percobaan rasio *test* dan *train* yaitu 90% data latih dan 10% data uji, 80% data latih dan 20% data uji, 70% data latih dan 30% data uji, berdasarkan penelitian (Gormantara, 2020). Tabel 4.10 adalah perbandingan data *train* dan data *test* yang digunakan dalam penelitian.

Tabel 4.11 Perbandingan Data Train dan Data Test Metode *Naive Bayes*

Data Train	Data Test
90%	10%
80%	20%
70%	30%

4.4.3 *Support Vector Machine*

Pada tahap ini dataset dibagi menjadi dua bagian, yaitu data latih dan data uji dengan beberapa percobaan rasio *test* dan *train* yaitu 90% data latih dan 10% data uji, 80% data latih dan 20% data uji, 70% data latih dan 30% data uji, berdasarkan penelitian (Gormantara, 2020). Tabel 4.11 adalah perbandingan data *train* dan data *test* yang digunakan dalam penelitian.

Tabel 4.12 Perbandingan Data Train dan Data Test Metode *Naive Bayes*

Data Train	Data Test
90%	10%
80%	20%
70%	30%

Untuk mencari parameter terbaik pada setiap kernel digunakan *grid search cross validation*. Pada kernel *linear*, variasi nilai *cost* yang diujikan adalah 0,001, 0,1, 1, 10, dan 1000. Pada kernel RBF variasi nilai *cost* yang diajukan adalah 1, 10, 50, 100 dan variasi *gamma* yang diujikan adalah 1, 2, 3, 4, dan 5. Pada kernel *Polynomial* variasi nilai *cost* yang diujikan adalah 100, 200, 300, 400, dan 500 dan variasi nilai *degree* yang diujikan adalah 1 dan 2. Tabel 4.13 berikut adalah

kernel dan parameter yang diujikan dengan *grid search cross validation* (Ningrum, 2018).

Tabel 4.13 Kernel dan Parameter yang diujikan

Kernel	Parameter yang Diujikan
Linear	$C = [0,01, 0,1, 1, 10, 100, 1000]$
RBF	$C = [1,10, 50, 100]$, $\gamma = [1, 2, 3, 4, 5]$
Polynomial	$C = [100, 200, 300, 400, 500]$, $\text{degree} = [1, 2]$

4.5 Assess

Tahap ini merupakan tahap dimana dilakukan evaluasi model penelitian.

Pada metode *Support Vector Machine* digunakan *grid search cross validation* dalam mencari nilai untuk parameter terbaik dimana selanjutnya digunakan pada model. Hasil evaluasi untuk metode *Naive Bayes* dan *Support Vector Machine* menggunakan *confusion matrix* yang berisi nilai akurasi, *precision*, *recall* dari data *test*. Untuk memaksimalkan nilai *confusion matrix* digunakan *k-folds cross validation* dengan nilai $k = 10$ (Hilmiyah, 2017).

4.5.1 Naive Bayes

Pada metode *Naive Bayes* digunakan beberapa skenario pembagian data uji yang tertulis pada tabel 4.11. Data *train* dan data *test* selanjutnya diolah menggunakan metode *Naive Bayes*. Hasil Akurasi dengan pembagian dataset 90:10 akurasi adalah 90% untuk pembagian data 80:20 akurasi adalah 92% dan untuk pembagian data 70:30 mendapatkan akurasi sebesar 88% Hasil akurasi terbaik adalah pada rasio dataset 80:20. Berikut adalah *script* akurasi yang

digunakan pada *Naive Bayes* dan Tabel 4.14 adalah hasil *accuracy*, *precision*, dan *recall* dari beberapa rasio dataset.

```
from sklearn.feature_extraction.text import
CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB

#vectorizer
dataset['text'] = dataset['text'].astype(str)

vec = CountVectorizer().fit(dataset['text'])
vec_transform = vec.transform(dataset['text'])
print(vec_transform)

#split data
x = vec_transform.toarray()
y = dataset['sentiment']
x_train, x_test, y_train, y_test = train_test_split(x,
y, test_size=0.2)

#akurasi NBC
metodeBN = MultinomialNB().fit(x_train, y_train)
predictNB = metodeBN.predict(x_test)

print('Accuracy=>')
print('Naive Bayes : ', metodeBN.score(x_test, y_test))

#evaluasi model NBC
y_pred = metodeBN.predict(x_test)
print('Accuracy of NB classifier on test set:
{:.2f}'.format(metodeBN.score(x_test, y_test)))

confusion_matrix = confusion_matrix(y_test, y_pred)
print(confusion_matrix)
print(classification_report(y_test, y_pred))
```


Tabel 4.14 Hasil Accuracy, Precision dan Recall

Data train : data test	Accuracy	Precision	Recall
90:10	90%	90%	89%
80:20	92%	93%	95%
70:30	88%	89%	87%

Hasil *confusion matrix* dengan metode *Naive Bayes* pada rasio 80:20 didapatkan bahwa prediksi benar pada sentimen positif atau *true positif* sebanyak 259. Prediksi benar pada sentimen negatif atau *true negatif* sebanyak 144. Tabel 4.15 adalah *confusion matrix*.

Tabel 4.15 Hasil *Confusion Matrix*

Hasil Akurasi	Nilai Prediksi	
	Negatif	Positif
Negatif	144	20
Positif	13	259

Persamaan 4.1 adalah perhitungan nilai akurasi metode *Naive Bayes*.

$$Accuracy = \frac{\text{total prediksi benar}}{\text{total keseluruhan data}} = \frac{403}{436} = 92\% \quad (4.1)$$

Persamaan 4.2 dan 4.3 adalah perhitungan presisi kelas positif dan negatif.

$$Precision \text{ kelas positif} = \frac{\text{true positif}}{\text{total prediksi positif}} = \frac{259}{20 + 259} = 93\% \quad (4.2)$$

$$Precision \text{ kelas negatif} = \frac{\text{true negatif}}{\text{total prediksi negatif}} = \frac{144}{144 + 13} = 92\% \quad (4.3)$$

Persamaan 4.4 dan 4.5 adalah perhitungan *recall* kelas positif dan negatif.

$$Recall \text{ kelas positif} = \frac{\text{true positif}}{\text{data positif aktual}} = \frac{259}{13 + 259} = 95\% \quad (4.4)$$

$$Recall \text{ kelas negatif} = \frac{\text{true negatif}}{\text{data negatif aktual}} = \frac{144}{144 + 20} = 88\% \quad (4.5)$$

Setelah mendapatkan rasio dengan nilai akurasi terbaik, kemudian dilakukan *cross validation* untuk mendapatkan akurasi yang maksimal. Pada penelitian ini digunakan 10-folds cross dengan hasil tertinggi *accuracy*, *precision*, dan *recall* adalah 96%, 97% dan 96%. Berikut adalah *script* untuk 10-Folds Cross Validation dan Tabel 4.16 adalah hasil dari 10-folds cross validation.

```
#10-Fold Cross Val NBC
akurasi_3_cross = cross_val_score(metodeBN, x, y,
scoring='accuracy', cv=10)
print("akurasi 10 CV = {}".format(akurasi_3_cross))
print("rataaan akurasi 10 CV = {}".format(akurasi_3_cross.mean()*100))
```

Tabel 4.16 Hasil 10-Folds Cross Validation

<i>n</i>	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>
1	88%	89%	88%
2	87%	88%	88%
3	86%	86%	85%
4	85%	84%	84%
5	96%	97%	96%
6	89%	88%	88%
7	84%	84%	83%
8	89%	88%	89%
9	86%	89%	88%
10	86%	86%	85%

4.5.2 Support Vector Machine

Pada metode *Support Vector Machine* menggunakan tiga kernel yaitu linear, rbf, dan *poly* serta menggunakan *grid search cross validation* untuk

mencari nilai parameter yang terbaik. Program *Python* untuk model *Support Vector Model* dijalankan sebanyak lima kali. Pada kernel linear, nilai parameter terbaik adalah 1 dengan hasil *accuracy*, *precision*, dan *recall* sebesar 94%, 93%, 97%. Pada kernel RBF, nilai parameter C terbaik adalah 10 dan nilai parameter gamma terbaik adalah 1 dengan *accuracy*, *precision*, dan *recall* sebesar 92%, 87%, 91%. Pada kernel *poly*, nilai parameter C terbaik adalah 1 dan nilai parameter *degree* terbaik adalah 1 dengan hasil *accuracy*, *precision*, dan *recall* sebesar 87%, 91%, 91%. Kinerja terbaik adalah pada kernel *linear* dimana hasil akurasi adalah 93% dengan nilai parameter dari *linear* yaitu C=1. Berikut adalah *script* untuk *Support Vector Machine*.

```
from sklearn.feature_extraction.text import
CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.svm import LinearSVC

from vaderSentiment.vaderSentiment import
SentimentIntensityAnalyzer
analyser = SentimentIntensityAnalyzer()

# split x dan y
x = TWEET_DATA['komentar']
y = TWEET_DATA['Sentiments']
x_train, x_test, y_train, y_test = train_test_split(x,
y, test_size=0.3)

# perform countvectorizer
vectorizer = CountVectorizer()
vectorizer.fit(x_train)

# x_train
x_train = vectorizer.transform(x_train)
x_test = vectorizer.transform(x_test)
```

```
#SVM dengan kernel
from math import gamma
for c in [0.001, 0.01, 0.1, 1, 10, 100]:
    svm = LinearSVC(C=1)
    svm.fit(x_train, y_train)
    print('Akurasi untuk c = %s: %s' % (c,
accuracy_score(y_test, svm.predict(x_test))))
```

Tabel 4.18 adalah hasil dari *grid search cross validation* setiap kernel, dan

Tabel 4.19 Hasil Grid Search Cross Validation pada Kernel Linear.

Tabel 4.17 Hasil *Grid Search Cross Validation* Tiap Kernel

Kernel	Nilai Parameter Terbaik	Accuracy	Precision	Recall
Linear	C=1	94%	93%	97%
RBF	C=10, gamma=1	92%	87%	91%
Poly	C=100, degree=1	87%	91%	91%

Tabel 4.18 Hasil *Grid Search Cross Validation* pada Kernel Linear

Parameter	Accuracy	Precision	Recall
C=0,001	91%	86%	88%
C=0,01	92%	87%	91%
C=0,1	93%	88%	92%
C=1	94%	93%	97%
C=10	90%	86%	92%
C=100	92%	85%	90%

Dilakukan beberapa pembagian rasio dataset dengan menggunakan kernel linear dimana nilai parameter C=1. Dimana dengan pembagian dataset 90:10 akurasi adalah 87% Untuk pembagian dataset 80:20 akurasi adalah 91%

Untuk pembagian dataset 70:30 menghasilkan akurasi sebesar 94%. Berikut merupakan *script* untuk akurasi model *Support Vector Machine* dan Tabel 4.20 adalah hasil akurasi dari kernel linear.

```
#Evaluasi model SVM
y_pred = svm.predict(x_test)
print('Accuracy of SVM classifier on test set:
{:.2f}'.format(svm.score(x_test, y_test)))

confusion_matrix = confusion_matrix(y_test, y_pred)
print(confusion_matrix)
print(classification_report(y_test, y_pred))
```

Tabel 4.19 Accuracy, Precision, Recall Kernel Linear

Data train : data test	Accuracy	Precision	Recall
90:10	87%	90%	92%
80:20	91%	93%	90%
70:30	94%	93%	97%

Kinerja terbaik pada kernel rbf adalah pada rasio 70:30 dengan nilai akurasi adalah 94% Hasil *confusion matrix* dari metode *Support Vector Machine* pada rasio 70:30 didapatkan bahwa prediksi benar pada sentimen positif atau *true positif* sebanyak 376. Prediksi benar pada sentimen negatif atau *true negatif* sebanyak 236. Tabel 4.21 adalah hasil *confusion matrix*-nya.

Tabel 4.20 Hasil *Confusion Matrix*

Hasil Akurasi	Nilai Prediksi	
	Negatif	Positif
Negatif	236	29
Positif	13	376

Persamaan 4.6 merupakan perhitungan akurasi metode *Support Vector Machine*.

$$Accuracy = \frac{\text{total prediksi benar}}{\text{total keseluruhan data}} = \frac{612}{654} = 94\% \quad (4.6)$$

Persamaan 4.7 dan 4.8 adalah perhitungan *precision* kelas positif dan negatif.

$$Precision \text{ kelas positif} = \frac{\text{true positif}}{\text{total prediksi positif}} = \frac{376}{29 + 376} = 93\% \quad (4.7)$$

$$Precision \text{ kelas negatif} = \frac{\text{true negatif}}{\text{total prediksi negatif}} = \frac{236}{13 + 236} = 95\% \quad (4.8)$$

Persamaan 4.9 dan 4.10 adalah perhitungan *recall* kelas positif dan negatif.

$$Recall \text{ kelas positif} = \frac{\text{true positif}}{\text{total prediksi positif}} = \frac{376}{13 + 376} = 97\% \quad (4.9)$$

$$Recall \text{ kelas negatif} = \frac{\text{true negatif}}{\text{total prediksi negatif}} = \frac{236}{236 + 29} = 89\% \quad (4.10)$$

Setelah mendapatkan rasio dengan hasil rata-rata *accuracy*, *precision*, dan *recall* terbesar, kemudian dilakukan *cross validation* untuk mendapatkan akurasi yang maksimal. Pada penelitian ini digunakan *10-folds cross* dengan hasil tertinggi *accuracy*, *precision*, dan *recall* sebesar 98%, 97% dan 95%. Berikut adalah *script* untuk *10-Folds Cross Validation* dan Tabel 4.22 adalah hasil dari *10-folds cross validation*.

```
from sklearn.model_selection import cross_validate

cross_val_score,
scores = cross_val_score(svm, x_train, y_train,
scoring='recall_weighted', cv=10)
scores
```

Tabel 4.21 Hasil *Cross Validation*

<i>n</i>	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>
1	94%	91%	94%
2	91%	92%	90%
3	93%	94%	93%
4	97%	96%	97%
5	94%	94%	93%
6	98%	97%	95%
7	91%	91%	92%
8	95%	94%	93%
9	91%	92%	91%
10	93%	92%	90%

4.6 Interpretasi Hasil

Hasil *crawling* data sejumlah 15.557 data mentah lalu dilakukan *text preprocessing* sehingga menjadi 5205 data bersih. Selanjutnya, dilakukan pelabelan 5205 dataset dengan metode *lexicon based*. Setelah dilakukan pelabelan dataset maka dihasilkan sebanyak 1327 data komentar masuk ke dalam kelas positif dan 853 data masuk ke dalam kelas negatif. Tabel 4.23 adalah contoh komentar yang berisi kelas positif dan negatif.

Tabel 4.22 Komentar Positif dan Negatif

Komentar Positif	Komentar Negatif
kena virus langsung sembuh karena pakai vaksin imun dikuatkan ayo vaksin untuk cegah sakit syarat hanya kartu keluarga vaksin gratis	lebih meresahkan pemaksaan vaksin untuk politik dengan cara syarat administrasi apapun kalau sudah meninggal setelah vaksin alasannya gak ada kaitannya dengan vaksin

perintah tenaga medis untuk tanggulangi wabah virus perkuat vaksinasi dan protokol kesehatan agar imun kuat dan sehat	resiko disuntik vaksin bukan berarti sehat selamanya artinya kita akan disuntik rutin bulan sekali rakyat mau di bikin ketergantungan zat kimia padahal tubuh orang itu berbeda beda ada yang kuat menerima ada yang tidak
presiden perintah masyarakat tingkatkan prokes jaga imunitas cegah penyebaran covid	Ngapain sehat negatif vaksin sudah banyak
vaksin sudah gratis tinggal paksa rakyat ikut vaksinasi saja	Pihak swasta apa ada peran jual vaksin pemerintah bisa saja kekurangan ini swasta bisa ada lalu jual ke rakyat bahaya ini negara bisa parah
Allah kasih pikiran positif agar hati tenang tingkatkan stamina imun terhindar dari covid	suntikan kedua sebagai vaksin pembentuk antibody secara optimal tapi masih ada resiko kena covid lagi aneh ini ga ada permainan kan

Berdasarkan proses pengujian klasifikasi dengan metode *Naive Bayes* dan *Support Vector Machine* pada data komentar didapatkan hasil akurasi dari kedua metode tersebut pada Tabel 4.22. Hasil dari penelitian menggunakan klasifikasi teks data komentar tentang vaksinasi covid-19 menggunakan metode *Naive Bayes* mendapatkan akurasi 92%. Sementara itu, akurasi yang didapatkan dengan menggunakan metode *Support Vector machine* adalah 94%.

Tabel 4.23 Kinerja *Naive Bayes Classifier* dan *Support Vector Machine*

Rasio	<i>Naive Bayes Classifier</i>			<i>Support Vector Machine</i>		
	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>
90:10	90%	90%	89%	87%	90%	92%
80:20	92%	93%	95%	91%	93%	90%
70:30	88%	89%	87%	94%	93%	97%

Berdasarkan tabel 4.24, dapat disimpulkan bahwa *Naive Bayes Classifier* dapat menghasilkan nilai *accuracy*, *precision* dan *recall* tertinggi yaitu 92%, 93% dan 95%, dengan penggunaan rasio data *testing* yang lebih sedikit yaitu 80:20. Sedangkan pada *Support Vector Machine* menghasilkan nilai *accuracy*, *precision* dan *recall* yang lebih tinggi apabila menggunakan data *testing* dengan rasio yang lebih besar yaitu 70:30 dengan nilai *accuracy* sebesar 94%, *precision* sebesar 93% dan *recall* sebesar 97%.

Penggunaan jumlah rasio data *testing* yang lebih besar berpengaruh pada tingginya nilai akurasi algoritma *Support Vector Machine* dibandingkan dengan algoritma *Naive Bayes Classifier* yang menghasilkan nilai akurasi tinggi dengan penggunaan data *testing* lebih kecil.

4.6.1 Analisa dengan Peneliti Terdahulu

Penelitian Munthe, Ansori dan Setiawan (2021) dengan menggunakan 1702 dataset mengenai komentar video *YouTube Food Vlogger* menggunakan metode *Naive Bayes* menghasilkan akurasi sebesar 97%. Penelitian Apriani dan Gustian (2019) dengan 1500 dataset mengenai Review Tokopedia pada Google Playstore menggunakan metode *Naive Bayes Classifier* menghasilkan akurasi

sebesar 97%. Penelitian Hakim (2021) dataset sejumlah 2539 data ulasan myIndiHome pada Google Playstore dengan metode *Naive Bayes Classifier* dan *Support Vector Machine* menghasilkan akurasi sebesar 85% dan 88%. Penelitian Dwianto dan Sadikin (2021) dengan 2000 data tweet mengenai Transportasi Online menggunakan metode *Naive Bayes Classifier* dan *Support Vector Machine* menghasilkan akurasi sebesar 84% dan 70%. Tabel 4.25 adalah hasil perbandingan kinerja.

Tabel 4.24 Hasil Perbandingan Kinerja

Peneliti	Objek	Data	Label Sentimen	Metode	
				Naive Bayes Classifier	Support Vector Machine
Munthe et al. (2021)	Komentar Saluran Video <i>YouTube Food Vlogger</i>	1702 data komentar	Positif dan Negatif	97%	-
Apriani et al. (2019)	Review Tokopedia pada <i>Google Playstore</i>	1500 data	Positif dan Negatif	97%	-
Hakim (2021)	Review <i>myIndiHome</i> pada <i>Google Playstore</i>	2539 data ulasan	Positif dan Negatif	85%	88%
Dwianto & Sadikin (2021)	<i>Tweet</i> mengenai Transportasi <i>Online</i>	2000 data tweet	Positif dan Negatif	84%	70%
Namira (2023)	Komentar <i>YouTube</i> Vaksinasi Covid-19	15.557 data komentar	Positif dan negatif	92%	94%

Pada penelitian ini dilakukan dengan pengujian 15.557 komentar gabungan dari *channel YouTube* KompasTv dan tvOneNews tentang Vaksinasi Covid-19. Tahap *preprocessing* dilakukan untuk membersihkan dataset dari karakter yang tidak memiliki sentimen untuk selanjutnya dilakukan pelabelan ke

dalam kelas positif dan negatif dengan metode *Lexicon*. Implementasi metode klasifikasi diterapkan pada dataset dimana hasil akurasi untuk metode *Naive Bayes* sebesar 92% dan akurasi metode *Support Vector Machine* dengan kernel paling optimal adalah linear (parameter $C=1$) sebesar 94%.



BAB 5

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan maka diperoleh beberapa kesimpulan sebagai berikut:

- a. Dari 15.557 data komentar pengguna *YouTube* mengenai vaksinasi covid-19 lalu dihasilkan sebanyak 6.545 data bersih setelah dilakukan *preprocessing* untuk selanjutnya diberikan label menggunakan *lexicon based*. Berdasarkan hasil pelabelan tersebut, mendapatkan data kelas positif sebanyak 1327 komentar dan kelas negatif sebanyak 835 komentar.
- b. Hasil penerapan metode *Naive Bayes Classifier* pada klasifikasi komentar dengan rasio 80:20 mendapatkan *accuracy* tertinggi pada pembagian data latih dan data uji 20:80 sebesar 92%, *precision* sebesar 93% dan *recall* sebesar 95%.
- c. Hasil penerapan metode *Support Vector Machine* menggunakan kernel linear dengan nilai parameter terbaik dari $C=1$, pada klasifikasi komentar mendapatkan nilai *accuracy* tertinggi pada pembagian data latih dan data uji 70:30 sebesar 94%, *precision* sebesar 93%, dan *recall* sebesar 97%.
- d. Penggunaan jumlah rasio data *testing* yang lebih besar berpengaruh pada tingginya nilai akurasi algoritma *Support Vector Machine* dibandingkan dengan algoritma *Naive Bayes Classifier* yang menghasilkan nilai akurasi tinggi dengan data *testing* lebih kecil.

Keterbatasan penelitian meliputi:

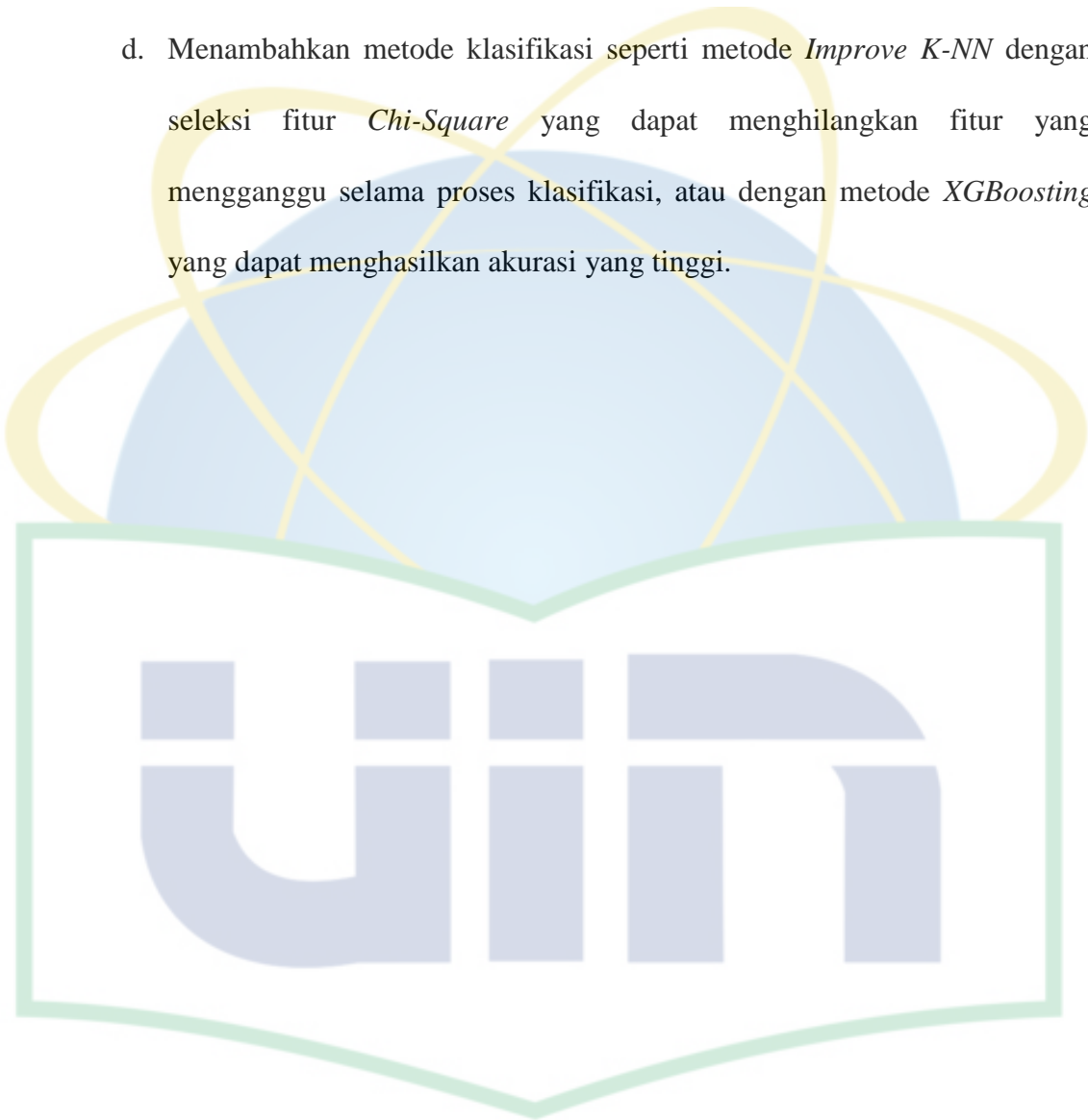
- a. Banyak penggunaan bahasa daerah dan bahasa yang tidak baku dalam data komentar, menyebabkan peneliti tidak memahami apa yang dimaksud dalam komentar tersebut.
- b. Pada proses *crawling* data komentar *YouTube*, juga terambil komentar yang tidak membicarakan vaksinasi, yang menyebabkan peneliti harus menghapus data komentar yang tidak berhubungan dengan topik penelitian.
- c. Pada proses normalisasi terbatas pada normalisasi kata yang terdapat pada kamus normalisasi saja, dimana pada komentar pengguna banyak digunakan kata tidak baku yang bermacam-macam dan terkadang ada huruf ganda sehingga tidak semua kata pada dataset ternormalisasi.

5.2 Saran

Berdasarkan hasil penelitian yang dilakukan, peneliti memiliki beberapa saran yang bisa menjadi masukan dan bahan pertimbangan untuk penelitian selanjutnya sebagai berikut:

- a. Dalam penelitian ini hanya menganalisis data berbahasa Indonesia. Untuk penelitian selanjutnya, dapat menambahkan data dalam bahasa asing seperti data berbahasa Inggris.
- b. Dalam penelitian ini, data diambil dari media sosial *YouTube*, pada penelitian berikutnya bisa mengambil data dari media sosial lainnya seperti *Instagram* atau *TikTok*.

- c. Adanya penambahan kamus bahasa gaul pada proses *stopword removal*, karena pada komentar pengguna *YouTube* masih banyak penggunaan bahasa-bahasa yang tidak baku.
- d. Menambahkan metode klasifikasi seperti metode *Improve K-NN* dengan seleksi fitur *Chi-Square* yang dapat menghilangkan fitur yang mengganggu selama proses klasifikasi, atau dengan metode *XGBoosting* yang dapat menghasilkan akurasi yang tinggi.



DAFTAR PUSTAKA

- Abdulloh, F. F., & Pambudi, I. R. (2021). Analisis Sentimen Pengguna Youtube Terhadap Program Vaksin Covid-19. *CSRID (Computer Science Research and Its Development Journal)*, 13(3), 141. <https://doi.org/10.22303/csrid.13.3.2021.141-148>
- Afifi, W. (2022). Analisis sentimen pengguna twitter terhadap layanan internet pt indosat tbk dengan metode k-nearest neighbor (k-nn) dan naive bayes classifier (nbc).
- Ahmad, M., Aftab, S., Bashir, M. S., & Hameed, N. (2018). Sentiment analysis using SVM: A systematic literature review. *International Journal of Advanced Computer Science and Applications*, 9(2), 182–188. <https://doi.org/10.14569/IJACSA.2018.090226>
- Alizah, M. D., Nugroho, A., Radiyah, U., & Gata, W. (2020). Sentimen Analisis Terkait Lockdown pada Sosial Media Twitter. *Indonesian Journal on Software Engineering (IJSE)*, 6(2), 223–229. <https://doi.org/10.31294/ijse.v6i2.8991>
- Anjasmos, M. T., Istiadi, I., & Marisa, F. (2020). Analisis Sentimen Aplikasi Go-Jek Menggunakan Metode SVM Dan NBC (Studi Kasus: Komentar Pada Play Store). *Conference on Innovation and Application of Science and Technology (CIASTECH 2020)*, *Ciastech*, 489–498.
- Annisa, T. (2022). *Mengenal peran sentiment analysis beserta cara kerjanya*. Ekrut Media. <https://www.ekrut.com/media/sentiment-analysis-adalah>
- Aziz, M. A. (2021). *ANALISIS TOPIK MENGGUNAKAN METODE LATENT*

*DIRICHLET ALLOCATION (LDA) PADA KOLOM KOMENTAR YOUTUBE
(STUDI KASUS : PERKEMBANGAN COVID-19 DI INDONESIA).*

Basch, C. H., Hillyer, G. C., Meleo-Erwin, Z. C., Jaime, C., Mohlman, J., &

Basch, C. E. (2020). Preventive behaviors conveyed on YouTube to mitigate transmission of COVID-19: Cross-sectional study. *JMIR Public Health and Surveillance*, 6(2), 3–8. <https://doi.org/10.2196/18807>

Dwianto, E., & Sadikin, M. (2021). Analisis Sentimen Transportasi Online pada Twitter Menggunakan Metode Klasifikasi Naïve Bayes dan Support Vector Machine. *Format : Jurnal Ilmiah Teknik Informatika*, 10(1), 94. <https://doi.org/10.22441/format.2021.v10.i1.009>

eldman, R., & Sanger, J. (2006). *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*No Title.

Fatihin, A. (2022). *Analisis Sentimen Terhadap Ulasan Aplikasi Mobile dengan Menggunakan Metode Support Vector Machine dan Pendekatan Lexicon Based.*

Ganesan, K. (2019). *All you need to know about text preprocessing for NLP and Machine Learning.* <https://www.kdnuggets.com/2019/04/text-preprocessing-nlp-machinelearning.html>

Gormantara, A. (2020). *Analisis Sentimen Terhadap New Normal Era di Indonesia pada Twitter Analisis Sentimen Terhadap New Normal Era di Indonesia pada Twitter Menggunakan Metode Support Vector Machine.*

Hakim, S. N. (2021). *Analisis Sentimen Persepsi Pengguna MyIndihome Menggunakan Metode Support Vector Machine (SVM) dan Naive Bayes*

Classifier (NBC). 3(March), 6.

Han, J., Kamber, M., & Pei, J. (2014). *Data mining: Data mining Concepts and Techniques.*

Herdhianto, A. (2020). *Sentiment Analysis Menggunakan Naïve Bayes Classifier (NBC) Pada Tweet Tentang Zakat.*
<http://repository.uinjkt.ac.id/dspace/handle/123456789/53661>

Hilmiyah, F. (2017). *Prediksi Kinerja Mahasiswa Menggunakan Support Vector Machine untuk Pengelola Program Studi di Perguruan Tinggi (Studi Kasus: Program Studi Magister Statistika ITS).* <https://repository.its.ac.id/>

Imamah, Husni, Malasari Rachman, E., Oktavia Suzanti, I., & Ayu Mufarroha, F. (2020). Text Mining and Support Vector Machine for Sentiment Analysis of Tourist Reviews in Bangkalan Regency. *Journal of Physics: Conference Series*, 1477(2). <https://doi.org/10.1088/1742-6596/1477/2/022023>

Kamruzzaman, S. M., & Rahman, C. M. (2004). Text Categorization using Association Rule and Naive Bayes Classifier. *Asian Journal of Information Technology*, 3.

Kibriya, A. M., Frank, E., Pfahringer, B., & Holmes, G. (2004). Multinomial Naive Bayes for Text Categorization Revisited. *Advances in Artificial Intelligence.*

Kulkarni, A., & Shivananda, A. (2019). Advanced Natural Language Processing. *International Journal of Media and Information Literacy.*

Kusuma, Y., & Prabayanti, H. R. (2022). Content Creator Yang Berkarakter Berdasarkan Analisis Video Youtube Ningsih Tinampi. *WACANA: Jurnal*

Ilmiah Ilmu Komunikasi, 21(2), 210–225.

<https://doi.org/10.32509/wacana.v21i2.2111>

Li, H. O. Y., Bailey, A., Huynh, D., & Chan, J. (2020). YouTube as a source of information on COVID-19: A pandemic of misinformation? *BMJ Global Health*, 5(5). <https://doi.org/10.1136/bmjgh-2020-002604>

Listiowarni, I., & Setyaningsih, E. R. (2018). Analisis Kinerja Smoothing pada Naive Bayes untuk Pengkategorian Soal Ujian. *Jurnal Teknologi Dan Manajemen Informatika*. <https://doi.org/10.26905/jtmi.v4i2.2080>

Maarif, S. D. (2021). *Adab Bersosial Media dalam Pandangan Islam*. Tirto.Id. <https://tirto.id/adab-bersosial-media-dalam-pandangan-islam-gch5>

Manning, C. D., Raghavan, P., & Schütze, H. (2009). *An Introduction to Information Retrieval*.

McGranaghan, M. F., & Santoso, S. (2007). Challenges and trends in analyses of electric power quality measurement data. *Eurasip Journal on Advances in Signal Processing*, 2007. <https://doi.org/10.1155/2007/57985>

McKinney, W. (2011). pandas: a Foundational Python Library for Data Analysis and Statistics. *Python for High Performance and Scientific Computing*, December, 1–9.

Mitchell, R. (2018). *Web Scraping with Python: Collecting More Data from the Modern Web*.

Munthe, M. P., Siswo, A., Ansori, R., & Septiawan, R. R. (2021). *Analisis Sentimen Komentar Pada Saluran Youtube Food Vlogger Berbahasa Indonesia Menggunakan Algoritma Naïve Bayes Sentiment Analysis*

Comment on Indonesian Youtube Channel About Food Vlogger Using Naïve Bayes Algorithm. 8(6), 11909–11916.

Nuri, A. (2022). *Implementasi Naive Bayes dan Support Vector Machine dengan Lexicon Based untuk Analisis Sentimen pada Twitter.*

Odim, M. O., Ogunde, A. O., Oguntunde, B. O., & Phillips, S. A. (2020). Exploring the performance characteristics of the naïve bayes classifier in the sentiment analysis of an Airline's social media data. *Advances in Science, Technology and Engineering Systems*, 5(4), 266–272. <https://doi.org/10.25046/aj050433>

Pelaksanaan Vaksinasi COVID-19 di Indonesia Membutuhkan Waktu 15 Bulan. (2021). Kementerian Kesehatan.

Prasetyo, E. (2012). *Data Mining - Konsep dan Aplikasi Menggunakan MATLAB.*

Prastyo, P. H., Sumi, A. S., Dian, A. W., & Permanasari, A. E. (2020). Tweets Responding to the Indonesian Government's Handling of COVID-19: Sentiment Analysis Using SVM with Normalized Poly Kernel. *Journal of Information Systems Engineering and Business Intelligence*, 6(2), 112. <https://doi.org/10.20473/jisebi.6.2.112-122>

Putra, M. S., Dharma Wati, S., & Solikin, I. (2021). *PADA MEDIA SOSIAL YOUTUBE MENGGUNAKAN ALGORITMA. 02, 99–105.*

PUTRI, D. U. K. (2016). *Implementasi Inferensi Fuzzy Mamdani untuk Keperluan Sistem Rekomendasi Berita Berbasis Konten.*

Putri, M. I. (2021). *Social Media Journalism : Monetisasi Berita di YouTube melalui News Vlog Packaging. 9(1), 64–77.*

- Romadoni, F., Umaidah, Y., & Sari, B. N. (2020). Text Mining Untuk Analisis Sentimen Pelanggan Terhadap Layanan Uang Elektronik Menggunakan Algoritma Support Vector Machine. *Jurnal Sisfokom (Sistem Informasi Dan Komputer)*, 9(2), 247–253. <https://doi.org/10.32736/sisfokom.v9i2.903>
- Sanjay, M. (2018). *Why and how to Cross Validate a Model?* <https://towardsdatascience.com/why-and-how-to-cross-validate-a-model-d6424b45261f>
- Sanjaya, G., & Lhaksmana, K. M. (2020). *Lexicon Based*). 7(3), 9698–9710.
- Sari, R., & Hayuningtyas, R. (2019). No Title. *Indonesian Journal on Software Engineering (IJSE)*, 5(Penerapan Algoritma Naive Bayes untuk Analisis Sentimen pada Wisata TMII Berbasis Website), 51–60.
- Sigmawaty, D., & Adriani, M. (2019). Jurnal Ilmu Komputer dan Informasi (Journal of Computer Science and Information) 12/2 (2019), 91-102. DOI: <http://dx:doi:org/10:21609/jiki:v12i2:745>. *Jurnal Ilmu Komputer Dan Informasi (Journal of ...*, 2, 75–84. <https://core.ac.uk/download/pdf/296598902.pdf>
- Sipayung, E., Maharani, H., & Zefanya, I. (2016). Perancangan Sistem Analisis Sentimen Komentar Pelanggan Menggunakan Metode Naive Bayes Classifier. *Jurnal Sistem Informasi*, Vol. 8.
- Srivastava, D. K., & Bhambhu, L. (2005). DATA CLASSIFICATION USING SUPPORT VECTOR MACHINE. *Journal of Theoretical and Applied Information Technology*.
- Susilowati, E., Sabariah, M. K., & Gozali, A. A. (2015). Implementasi Metode

Support Vector Machine untuk Melakukan Klasifikasi Kemacetan Lalu Lintas Pada Twitter. *E-Proceeding of Engineering*, 2(1), 1478–1484.

Taufiqurrahman, F., Faraby, S. Al, & Purbolaksono, M. D. (2021). Klasifikasi Teks Multi Label pada Hadis Terjemahan Bahasa Indonesia Menggunakan Chi Square dan SVM. *E-Proceeding of Engineering*, 8(5), 10650–10659.

Villavicencio, C., Macrohon, J. J., Inbaraj, X. A., Jeng, J. H., & Hsieh, J. G. (2021). Twitter sentiment analysis towards covid-19 vaccines in the Philippines using naïve bayes. *Information (Switzerland)*, 12(5). <https://doi.org/10.3390/info12050204>

Vlachos, E., & Tan, Z.-H. (2018). Public perception of android robots: Indications from an analysis of YouTube comments. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. <https://ieeexplore.ieee.org/document/8594058>

Yulita, W., Nugroho, E. D., & Algifari, M. H. (2021). Analisis Sentimen Terhadap Opini Masyarakat Tentang Vaksin Covid-19 Menggunakan Algoritma Naïve Bayes Classifier. *Ejurnal.Teknokrat.Ac.Id*, 2(2), 1–9. <https://ejurnal.teknokrat.ac.id/index.php/JDMSI/article/view/1344>

Zalyhaty, L. Q. (2021). *Analisis Sentimen Tanggapan Masyarakat Terhadap Vaksin Covid-19 Menggunakan Algoritma Support Vector Machine (SVM)*.



Script Code Crawling

```
pip install youtube-comment-downloader

youtube-comment-downloader --url "https://youtu.be/S-HwhWlW5XM" --
output tvonemaret.json #masukkan masing-masing link video
```

Script Code Preprocessing

```
import pandas as pd
import numpy as np
import nltk
nltk.download('punkt')
nltk.download('stopwords')

# ----- Case Folding -----
TWEET_DATA['komentar'] = TWEET_DATA['komentar'].str.lower()
print('Case Folding Result : \n')
print(TWEET_DATA['komentar'])
print('\n\n\n\n\n')

import string
import re #regex library

# import word_tokenize & FreqDist from NLTK
from nltk.tokenize import word_tokenize
from nltk.probability import FreqDist

# ----- Tokenizing -----
def remove_komentar_special(text):
    # remove tab, new line, and back slice
    text = text.replace('\t', " ").replace('\n', " ")
    text = text.replace('\u', " ").replace('\ ', " ")
    # remove non ASCII (emoticon, chinese word, .etc)
    text = text.encode('ascii', 'replace').decode('ascii')
    # remove mention, link, hashtag
    text = ' '.join(re.sub("([@#][A-Za-z0-9]+)|(\w+:\/\/\S+)", " ",
text).split())

#remove number
def remove_number(text):
    return re.sub(r"\d+", "", text)

TWEET_DATA['komentar'] =
TWEET_DATA['komentar'].apply(remove_number)

#remove punctuation
def remove_punctuation(text):
    return text.translate(str.maketrans("", "", string.punctuation))

TWEET_DATA['komentar'] =
TWEET_DATA['komentar'].apply(remove_punctuation)
```

```

#remove whitespace leading & trailing
def remove_whitespace_LT(text):
    return text.strip()

TWEET_DATA['komentar'] =
TWEET_DATA['komentar'].apply(remove_whitespace_LT)

#remove multiple whitespace into single whitespace
def remove_whitespace_multiple(text):
    return re.sub('\s+', ' ', text)

TWEET_DATA['komentar'] =
TWEET_DATA['komentar'].apply(remove_whitespace_multiple)

# remove single char
def remove_singl_char(text):
    return re.sub(r"\b[a-zA-Z]\b", "", text)

TWEET_DATA['komentar'] =
TWEET_DATA['komentar'].apply(remove_singl_char)

# NLTK word rokenize
def word_tokenize_wrapper(text):
    return word_tokenize(text)

TWEET_DATA['komentar_tokens'] =
TWEET_DATA['komentar'].apply(word_tokenize_wrapper)

print('Tokenizing Result : \n')
print(TWEET_DATA['komentar_tokens'].head())
print('\n\n\n')

```

Script Code Lexicon

```

!pip install tweet-preprocessor
!pip install textblob
!pip install sastrawi
!pip install emoji

import os
import pandas as pd
import tweepy
import re
import string
from textblob import TextBlob
import preprocessor as p
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
import datetime
from datetime import timedelta

```

```

import numpy as np
import emoji
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.feature_extraction.text import CountVectorizer
from Sastrawi.StopWordRemover.StopWordRemoverFactory import
StopWordRemoverFactory

#lexiconbased
!pip install VaderSentiment
from vaderSentiment.vaderSentiment import
SentimentIntensityAnalyzer
analyser = SentimentIntensityAnalyzer()

#menghitung score
scores = [analyser.polarity_scores(x) for x in df['komentar']]
print(scores)
df['Compound_Scores'] = [x['compound'] for x in scores]

#pemberian label sentiment
df.loc[df['Compound_Scores'] < 0, 'Sentiments'] = 'Negative'
df.loc[df['Compound_Scores'] > 0, 'Sentiments'] = 'Positive'
df.head()

```

Script Code Naive Bayes Classifier

```

import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report

#vectorizer
dataset['text'] = dataset['text'].astype(str)

vec = CountVectorizer().fit(dataset['text'])
vec_transform = vec.transform(dataset['text'])
print(vec_transform)

#split data
x = vec_transform.toarray()
y = dataset['sentiment']
x_train, x_test, y_train, y_test = train_test_split(x, y,
test_size=0.2)

#akurasi NBC
metodeBN = MultinomialNB().fit(x_train, y_train)

```

```

predictNB = metodeBN.predict(x_test)

print('Accuracy=>')
print('Naive Bayes : ', metodeBN.score(x_test, y_test))

#evaluasi model NBC
y_pred = metodeBN.predict(x_test)
print('Accuracy of NB classifier on test set:
{:.2f}'.format(metodeBN.score(x_test, y_test)))

confusion_matrix = confusion_matrix(y_test, y_pred)
print(confusion_matrix)
print(classification_report(y_test, y_pred))

from sklearn.model_selection import cross_validate

#10-Fold Cross Val NBC
akurasi_3_cross = cross_val_score(metodeBN, x, y,
scoring='recall_weighted', cv=10)
print("akurasi 10 CV = {}".format(akurasi_3_cross))
print("rataaan akurasi 10 CV =
{}".format(akurasi_3_cross.mean()*100))

```

Script Code Support Vector Machine

```

import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.svm import LinearSVC
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import cross_validate
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report

from vaderSentiment.vaderSentiment import
SentimentIntensityAnalyzer
analyser = SentimentIntensityAnalyzer()

# split x dan y
x = TWEET_DATA['komentar']
y = TWEET_DATA['Sentiments']
x_train, x_test, y_train, y_test = train_test_split(x, y,
test_size=0.3)

# perform countvectorizer

```

```
vectorizer = CountVectorizer()
vectorizer.fit(x_train)

# x_train
x_train = vectorizer.transform(x_train)
x_test = vectorizer.transform(x_test)

#SVM kernel
from math import gamma
for c in [0.01, 0.05, 0.25, 0.5, 0.75, 1]:
    svm = LinearSVC(C=100)
    svm.fit(x_train, y_train)
    print('Akurasi untuk c = %s: %s' %(c, accuracy_score(y_test,
svm.predict(x_test))))

#Evaluasi model
y_pred = svm.predict(x_test)
print('Accuracy of SVM classifier on test set:
{:.2f}'.format(svm.score(x_test, y_test)))

confusion_matrix = confusion_matrix(y_test, y_pred)
print(confusion_matrix)
print(classification_report(y_test, y_pred))

#10-folds cross val
cross_val_score,
scores = cross_val_score(svm, x_train, y_train,
scoring='precision_weighted', cv=10)
scores
```



**KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI (UIN)
SYARIF HIDAYATULLAH JAKARTA
FAKULTAS SAINS DAN TEKNOLOGI**

Jl. Ir. H. Juanda No. 95 Ciputat Indonesia 15412
Telp. (62-21) 7401925 Fax. (62-21) 7493315

Email : fst.uinjkt.ac.id
Website: <http://fst.uinjkt.ac.id>

Nomor : B- 77/F9/KM.01.2/02/2022
Lampiran : -
Perihal : Bimbingan Skripsi

Jakarta, 24 Februari 2022

Kepada Yth.

1. **Dr. Qurrotul Aini, MT**
2. **Suci Ratnawati, MTI**

Assalamu 'alaikum Wr. Wb.

Dengan ini diharapkan kesediaan Saudara untuk menjadi pembimbing I/II (Materi/Teknis)* penulisan skripsi mahasiswa:

Nama : Mifta Namira
NIM : 11160930000018
Program Studi : Sistem Informasi
Judul Skripsi : **"Analisis Sentimen Vaksinasi Covid-19 pada Komentar YouTube dengan Algoritma Naive Bayes Classifier dan Support Vector Machine"**

Judul tersebut telah disetujui oleh Program Studi bersangkutan pada tanggal dengan outline, abstraksi dan daftar pustaka terlampir. Bimbingan skripsi ini diharapkan selesai dalam waktu 6 (enam) bulan setelah ditandatanganinya surat penunjukan pembimbing skripsi.

Apabila terjadi perubahan terkait dengan skripsi tersebut selama proses pembimbingan, harap segera melaporkan kepada Program Studi bersangkutan.

Demikian atas kesediaan Saudara, kami ucapkan terima kasih.

Wassalamu 'alaikum Wr. Wb.

a.n Dekan
Wadek Bidang Akademik



Dr. Ir. Siti Rochaeli, M. Si
NIP. 19620308 198903 2 001

Tembusan:
Dekan (sebagai laporan)