

**ANALISIS SENTIMEN MASYARAKAT TERHADAP
KEBIJAKAN PENERAPAN PPKM DI MEDIA SOSIAL
TWITTER DENGAN MENGGUNAKAN METODE
*XGBOOST***

Tugas Akhir

Untuk memenuhi sebagian persyaratan
mencapai derajat Sarjana S-1 Program Studi Teknik Informatika



Oleh:

I Putu Angga Purnama Widiarta

F1D018024

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MATARAM

Juli 2023

HALAMAN PENGESAHAN

Bagian ini ditimpa dengan lembar pengesahan yang dihasilkan dari system
<https://ta.if.unram.ac.id/>

KATA PENGANTAR

Assalamualaikum Wr. Wb.

Segala puji bagi Tuhan Yang Maha Esa yang telah memberikan karunia-NYA sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul “Analisis Sentimen Masyarakat Terhadap Kebijakan Penerapan PPKM Di Media Sosial Twitter Dengan Menggunakan Metode Xgboost” tepat pada waktunya. Pada kesempatan ini pula, penulis menghaturkan terima kasih yang sebesar-besarnya kepada seluruh pihak yang telah mendukung agar terselesaikannya Tugas Akhir ini.

Penulis tentu menyadari Tugas Akhir ini masih jauh dari kata sempurna dan masih terdapat kesalahan dan kekurangan di dalamnya. Untuk itu, diharapkan kritikserta saran dari pembaca untuk Tugas Akhir ini, agar Tugas Akhir ini dapat menjadi lebih baik lagi.

Demikian yang bisa disampaikan, sekali lagi terima kasih atas semua pihak yang telah membantu di dalam penyelesaian Tugas Akhir ini. Semoga Tugas Akhir ini dapat bermanfaat bagi para pembaca.

Wassalamualaikum Wr. Wb.

Mataram, 1 Juli 2023

UCAPAN TERIMA KASIH

Penulis menyadari bahwa terselesaikannya Tugas Akhir ini tentunya bukan hanya dari usaha penulis saja. Tugas Akhir ini bisa selesai tepat waktu tentunya berkat dukungan dari semua pihak yang terlibat juga. Oleh karena itu, pada kesempatan ini penulis sampaikan terima kasih yang sebesar-besarnya kepada:

1. Tuhan Yang Maha Esa, yang telah memberikan berkat dan kesempatan untuk terus bersyukur bisa berada pada tahapan ini.
2. Kedua orang tua dan keluarga yang selama ini telah memberikan doa dan dukungannya dalam menjalani perkuliahan dan pengerjaan Tugas Akhir ini sehingga dapat terselesaikan.
3. Bapak Ramaditia Dwiyanaputra S.T. M.Eng. selaku dosen pembimbing utama yang telah memberikan bimbingan, semangat dan arahan kepada penulis selama penyusunan Tugas Akhir sehingga dapat selesai dengan baik.
4. Bapak Arik Aranta, S.Kom., M.Kom. selaku dosen pembimbing pendamping yang telah memberikan bimbingan dan arahan kepada penulis selama penyusunan Tugas Akhir sehingga dapat selesai dengan baik.
5. Dosen penguji, atas pemberian kritik dan saran yang bersifat membangun serta diperlukan dalam penyelesaian Tugas Akhir ini.
6. Teman-teman terdekat penulis selama menjalani perkuliahan di Program Studi Teknik Informatika dan telah membantu dalam penyelesaian tugas akhir ini.
7. Semua pihak yang tidak dapat penulis sebutkan satu persatu, yang telah memberikan do'a dan dukungan baik moril maupun materil sehingga penulis dapat menyelesaikan pembuatan Tugas Akhir dengan baik.

Semoga Tuhan Yang Maha Kuasa selalu memberikan rahmat dan hidayah- Nya memberikan imbalan yang setimpal atas bantuan yang diberikan kepadapenulis.

ABSTRAK

Corona Virus Disease (Covid-19) merupakan virus yang dapat menyebabkan infeksi pada saluran pernafasan manusia, wabah ini bermula di Wuhan, Provinsi Hubei, China. Indonesia menjadi salah satu negara yang ikut terjangkit virus ini, pemberlakuan pembatasan kegiatan masyarakat (PPKM) diterapkan oleh pemerintah sebagai suatu kebijakan demi menekan angka persebaran dari *covid-19*. Pro dan kontra bermunculan akibat dari dampak yang ditimbulkan oleh kebijakan tersebut, maka dari itu untuk menilai bagaimana sentimen masyarakat terhadap kebijakan tersebut, dilakukanlah analisis sentimen pada opini masyarakat Indonesia. Penelitian ini bertujuan untuk mengimplementasikan algoritma *XGBoost* pada proses klasifikasi sentimen, sentimen analisis menargetkan opini masyarakat di media sosial *twitter*, jumlah *dataset* yang digunakan setelah dilakukan proses *labeling* yaitu 1958 *tweets* positif, dan 3980 *tweets* negatif, augmentasi dilakukan pada *tweets* dengan kategori positif sehingga jumlahnya bertambah menjadi 3916. Pada tahap *preprocessing*, dilakukan proses *casefolding*, *stopwords removal*, *tokenizing*, dan *stemming*. Pemberian bobot pada kata atau *term* menggunakan metode *TF-RF* untuk mengubah setiap *term* menjadi angka. Pada tahapan akhir, klasifikasi dilakukan dengan mengimplementasikan metode *XGBoost* dengan *score hyperparameter* yang optimal. *K-fold cross validation* digunakan untuk mengevaluasi kinerja model. Hasil yang diperoleh pada salah satu skenario pengujian yaitu sebesar 85.27% pada akurasi, 86.07% pada presisi, dan 85.23% *recall*.

Kata kunci – Sentimen Analisis, *Covid-19*, *XGBoost*, *Term Frequency-Relevance Frequency*, *Preprocessing*, *K-Fold Cross Validation*

DAFTAR ISI

HALAMAN PENGESAHAN.....	ii
KATA PENGANTAR.....	iii
UCAPAN TERIMA KASIH.....	iv
ABSTRAK	v
DAFTAR ISI.....	vi
DAFTAR GAMBAR.....	viii
DAFTAR TABEL	ix
DAFTAR KODE SUMBER	xi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian.....	5
1.5 Manfaat Penelitian.....	5
1.6 Sistematika Penulisan.....	5
BAB II TINJAUAN PUSTAKA.....	7
2.1 Penelitian Terkait.....	7
2.2 Teori Penunjang.....	12
2.2.1 Text Mining	12
2.2.2 Sentimen Analisis	12
2.2.3 <i>Web Crawling</i>	13
2.2.4 <i>Tweet</i>	13
2.2.5 Preprocessing.....	14
2.2.6 Term Frequency – Relevance Frequency (TF-RF)	15
2.2.7 <i>XGBoost</i>	16
2.2.8 <i>Confusion Matrix</i>	18
BAB III METODOLOGI PENELITIAN	20
3.1 Alat dan Bahan	20
3.1.1 Alat Penelitian.....	20
3.1.2 Bahan Penelitian	20
3.2 Studi Literatur.....	22
3.3 Alur Penelitian.....	22
3.4 Kebutuhan Sistem.....	24
3.4.1 Analisis Pengguna	25
3.4.2 Analisis Perangkat Keras.....	25
3.4.3 Analisis perangkat lunak	25
3.5 Perancangan Sistem.....	26
3.5.1 <i>Web Crawling Twitter</i>	27
3.5.2 <i>Input Dataset Tweet Training dan Testing</i>	28
3.5.3 <i>Text Preprocessing Tweets Dataset</i>	30
3.5.4 <i>Feature Weighting</i>	36
3.5.5 Klasifikasi Dengan XGBoost	40
3.6 Pengujian	44
3.7 Jadwal Penelitian.....	Error! Bookmark not defined.

BAB IV HASIL DAN PEMBAHASAN	48
4.1 Pengumpulan Data	48
4.2 Text Preprocessing	49
4.2.1 Case Folding	49
4.2.2 Tokenizing	50
4.2.3 Stopwords Filtering	51
4.2.4 Stemming	52
4.3 Term Weighting	52
4.4 Pengujian	55
4.5 Hasil Pengujian	57
BAB V KESIMPULAN DAN SARAN	65
5.1 Kesimpulan	65
5.2 Saran	65
DAFTAR PUSTAKA	66

DAFTAR GAMBAR

Gambar 2.1 Proses <i>text mining</i>	12
Gambar 3.1 Alur penelitian.....	24
Gambar 3.2 Perancangan sistem	27
Gambar 3.3 Contoh <i>decision tree</i>	42
Gambar 3.4 Ilustrasi <i>cross validation 5 fold</i>	45
Gambar 4.1 Bagan proses <i>preprocessing</i>	49
Gambar 4.2 Tahap <i>Case Folding</i>	50
Gambar 4.3 Tahap <i>Tokenizing</i>	51
Gambar 4.4 Tahap <i>Stopwords Filtering</i>	51
Gambar 4.5 Tahap <i>Stemming</i>	52
Gambar 4.6 Tahapan pembobotan <i>term</i>	55
Gambar 4.7 Perbandingan jumlah data sebelum diaugmentasi	57
Gambar 4.8 Perbandingan jumlah data setelah diaugmentasi.....	58
Gambar 4.9 Skenario pengujian pertama hingga kedua.....	58
Gambar 4.10 Wordcloud sentimen kategori positif pada dataset tanpa augmentasi	53
Gambar 4.11 Wordcloud sentimen kategori negatif pada dataset tanpa augmentasi	53
Gambar 4.12 Wordcloud sentimen kategori positif pada dataset augmentasi	54
Gambar 4.13 Wordcloud sentimen kategori negatif pada dataset augmentasi	54

DAFTAR TABEL

Tabel 2.1 Penelitian sebelumnya.....	9
Tabel 2.2 Tabel <i>confusion matrix</i>	18
Tabel 3.1 Kebutuhan perangkat keras	25
Tabel 3.2 Kebutuhan perangkat lunak	26
Tabel 3.3 <i>Tweet training</i>	29
Tabel 3.4 <i>Tweet casefolding</i>	30
Tabel 3.5 <i>Tweet tokenization</i>	32
Tabel 3.6 <i>Tweet stopwords removal</i>	34
Tabel 3.7 <i>Tweet stemming</i>	35
Tabel 3.8 Nilai TF	37
Tabel 3.9 Nilai RF.....	38
Tabel 3.10 Nilai TF-RF Kategori Tweet Positif	39
Tabel 3.11 Nilai TF-RF Kategori Tweet Negatif.....	39
Tabel 3.12 <i>hyperparameter XGBoost</i>	40
Tabel 3.13 Contoh nilai fitur dengan 2 Kelas	41
Tabel 3.14 <i>Confusion matrix</i> yang digunakan pada penelitian	47
Tabel 3.15 Jadwal penelitian.....	Error! Bookmark not defined.
Tabel 4.1 Pengujian pada dataset yang tidak diaugmentasi dan telah distemming dengan k-fold = 5	59
Tabel 4.2 Pengujian pada dataset yang tidak diaugmentasi dan telah distemming dengan k-fold = 10	60
Tabel 4.3 Pengujian pada dataset yang tidak diaugmentasi dan tidak distemming dengan k-fold = 5	60
Tabel 4.4 Pengujian pada dataset yang tidak diaugmentasi dan tidak distemming dengan k-fold = 10	61
Tabel 4.5 Pengujian pada dataset yang diaugmentasi dan telah distemming dengan k-fold = 5.....	62
Tabel 4.6 Pengujian pada dataset yang diaugmentasi dan telah distemming dengan k-fold = 10.....	62

Tabel 4.7 Pengujian pada dataset yang diaugmentasi dan tidak distemming dengan k-fold = 5	63
Tabel 4.8 Pengujian pada dataset yang diaugmentasi dan tidak distemming dengan k-fold = 10.....	64

DAFTAR KODE SUMBER

BAB I

PENDAHULUAN

1.1 Latar Belakang

Corona Virus Disease (Covid-19) merupakan bagian dari keluarga besar virus yang dapat menyebabkan penyakit baik pada hewan maupun manusia. Ditemukan pada akhir tahun 2019 [1]. *Corona* telah dikaitkan dengan infeksi saluran pernafasan pada manusia, mulai dari flu biasa hingga penyakit yang lebih serius seperti *Severe Acute Respiratory Syndrome* (SARS) dan *Middle East Respiratory Syndrome* (MERS), menurut *World Health Organization* (WHO). Wabah ini bermula di Wuhan, Provinsi Hubei, China. Seperti diketahui, masyarakat Tionghoa sering mengonsumsi makanan “aneh” seperti kelelawar, babi, anjing, tikus dan hewan lainnya [2].

Pada *kuartal* awal tahun ini, berdasarkan pada data yang diperoleh dari halaman *website* worldometers, kasus harian *covid-19* dalam skala global mengalami penurunan yang sangat signifikan, tercatat pada tanggal 1 Januari 2022 jumlah kasus harian yang tercatat sebanyak 1.858.097 kemudian per tanggal 15 Mei 2022 jumlah kasus harian yang tercatat menunjukkan angka sebanyak 657.158, menurunnya kasus harian ini sangat dipengaruhi oleh faktor dari pemerataan vaksinasi yang sudah dilakukan, kemudian untuk skala di Indonesia, kasus harian yang tercatat pada tanggal 15 Januari 2022 menunjukkan jumlah sebanyak 1054 kasus, tidak seperti kasus harian yang terjadi di lingkup global pada umumnya, kasus harian di Indonesia malah mengalami kenaikan dari bulan Januari sampai bulan Februari yang dimana puncaknya terjadi pada tanggal 17 Februari dimana kasus harian yang tercatat berjumlah 63.956, lalu kurva tersebut tiba – tiba mengalami penurunan dari bulan tersebut hingga per tanggal 27 April yang menunjukkan jumlah sebanyak 617. Indonesia memang menjadi salah satu negara yang ikut terjangkit virus corona. Oleh karena itu, pemberlakuan pembatasan kegiatan masyarakat (PPKM) diterapkan oleh pemerintah sebagai suatu kebijakan baru yang dimulai pada tanggal 11 Januari 2021 demi menekan angka persebaran dari penyakit *covid-19* yang disebabkan oleh virus corona dengan cara membatasi

pergerakan beserta aktivitas masyarakat. Kebijakan ini terdiri dari beberapa tingkatan, dimana tingkatannya ditentukan oleh seberapa banyak kasus yang telah terjadi di suatu daerah dimana kebijakan ini diterapkan, dimulai dari level 1 (kasus rendah), level 2 (kasus sedang), level 3 (kasus tinggi), level 4 (kasus sangat tinggi).

Kebijakan pemerintah pusat dalam melaksanakan PPKM berdampak signifikan terhadap berbagai sektor kehidupan masyarakat. Kurangnya kerjasama antar pihak, terutama antara pemerintah pusat dan pemerintah daerah menyebabkan pengendalian virus corona menjadi terombang-ambing akibat dari ketidakselarasan koordinasi [3]. Pengaruhnya terhadap sektor ekonomi adalah yang paling terlihat. Pembatasan kemampuan untuk melakukan kegiatan skala besar pasti akan mengakibatkan perekonomian menjadi semakin sulit, dengan beberapa kegiatan ekonomi berhenti. Secara alami, itu memiliki dampak signifikan pada struktur kekuasaan masyarakat. Orang akan memprioritaskan makanan dan kebutuhan penting lainnya, sementara menunda-nunda permintaan sekunder dan tersier, yang mengakibatkan pengurangan substansial dalam tabungan [4].

Pro dan kontra bermunculan di kalangan masyarakat, hal ini dapat dilihat di berbagai lini khususnya media sosial. Media sosial adalah jenis media yang menghubungkan pengguna dan memungkinkan mereka untuk berkomunikasi satu sama lain. Salah satu platform yang sering digunakan selama periode PPKM adalah *twitter*. Dilansir dari *Global Digital Statistic* "Digital, Social & Mobile in 2019", pada tahun 2019 pengguna media sosial di Indonesia berjumlah 150 juta pengguna, dimana *twitter* merupakan media sosial dengan pengguna terbanyak yang mencakup lebih dari 52% dari total pengguna media sosial di Indonesia [5], keunggulan yang dimiliki oleh *twitter* apabila dibandingkan dengan media sosial lainnya yaitu ada pada penyebaran isi pesan dengan cakupan yang lebih luas karena pemilik akun *twitter* bisa melihat *tweet* dari pengguna lainnya tanpa perlu menjadi teman terlebih dahulu [6]. Informasi yang disampaikan di *twitter* juga dapat dijelaskan secara singkat, padat dan mudah dibaca, ini dikarenakan dalam menulis *tweet* diberlakukan pembatasan karakter sebanyak maksimal 140 karakter, pertemanan yang tersedia di *twitter* juga tidak dibatasi [7]. Di Twitter, opini publik memiliki sifat yang tidak dibatasi dan bebas [8]. Artinya, opini yang dibuat bisa

bersifat baik, negatif, atau netral. Berbagai opini pada ranah politik memiliki pengaruh besar terhadap seberapa baik kinerja pemerintah. Dalam PPKM, opini publik dinyatakan sebagai reaksi yang positif, negatif, atau netral terhadap pemerintah. Namun, agar opini dapat digunakan sebagai informasi yang bermakna, diperlukan prosedur analisis sentimen yang dapat menangani semua opini publik untuk memperoleh inferensi tekstual dari isi benak seluruh masyarakat Indonesia.

Analisis sentimen merupakan salah satu bidang studi dari bidang studi dengan lingkup lebih besar yang disebut dengan pemrosesan bahasa alami (*natural language processing*) atau biasa disingkat dengan nama NLP. NLP merupakan serangkaian teknik komputasi yang termotivasi secara teoritis untuk menganalisis dan mewakili teks yang terjadi secara alami pada satu atau lebih tingkat analisis linguistik untuk tujuan mencapai pemrosesan bahasa mirip manusia untuk berbagai tugas atau aplikasi [9]. Sementara analisis sentimen adalah metode untuk memahami, menganalisis, dan memproses *input* tekstual secara otomatis untuk memperoleh informasi sentimen dari suatu opini [10]. Analisis sentimen dilakukan dengan cara mengekstrak kemudian mengolah suatu teks atau kalimat dari sumber tertentu seperti berita dan media sosial untuk memperoleh sentimen yang terkandung pada teks atau kalimat, sentimen tersebut terdiri dari 3 jenis opini, yaitu opini positif, opini negatif, dan opini netral, sehingga dengan dilakukannya sentimen analisis, perusahaan atau instansi memperoleh manfaat yaitu dapat mengetahui respon masyarakat terhadap suatu pelayanan, kebijakan atau produk, melalui *feedback* yang diberikan oleh masyarakat maupun para ahli [11].

Pada sentimen analisis, *input* yang digunakan meliputi suatu kalimat atau teks yang ingin digali emosi atau sentimen yang tersirat didalamnya, sementara *output* yang dihasilkan adalah sentimen atau emosi yang ada pada teks atau kalimat yang digunakan sebagai *input*. Metode yang menonjol untuk memproses sentimen adalah metode yang menggunakan pendekatan *machine learning*. Penelitian mengenai sentimen analisis dengan menggunakan pendekatan *machine learning* terkait dengan kebijakan PPKM sebelumnya dengan metode *Support Vector Machine* (SVM) sudah dilakukan oleh Putra, dkk. Dimana pada penelitian tersebut, nilai akurasi yang diperoleh sebesar 64% [12]. Kemudian pada penelitian yang

dilakukan oleh Krisdiyanto, T dkk. Proses analisis opini diklasifikasikan menjadi 2 sentimen yaitu positif atau negatif, proses klasifikasi menggunakan metode *Naïve Bayes Clasifiers*, diperoleh akurasi sebesar 99% yang termasuk kedalam polaritas positif dan 1% pada polaritas negatif [13]. Pada penelitian ini, penulis akan mengimplementasikan penggunaan dari metode *XGBoost* sebagai algoritma klasifikasi, dan mengimplementasikan metode *TF-RF (Term Frequency – Relevance Frequency)* sebagai metode untuk menentukan bobot dari suatu *term* pada teks. *XGBoost* menghemat waktu, mengoptimalkan sumber daya memori, dan dapat diterapkan secara paralel selama proses implementasi untuk mengelola sentimen.

1.2 Rumusan Masalah

Berdasarkan dari latar belakang yang telah diuraikan, dapat dirumuskan permasalahan penelitian ini adalah sebagai berikut:

1. Bagaimana mengimplementasikan metode *eXtreme Gradient Boosting (XGBoost)* dalam melakukan analisis sentimen masyarakat terhadap penerapan kebijakan PPKM di media sosial *Twitter*?
2. Bagaimana performa pengujian pada analisis sentimen masyarakat terhadap penerapan kebijakan PPKM di media sosial *Twitter*?
3. Bagaimana sentimen mayoritas masyarakat Indonesia terhadap penerapan kebijakan PPKM di media sosial *Twitter*?
4. Apa saja jenis kategori kata yang akan dicari pada analisis sentimen masyarakat terhadap penerapan kebijakan PPKM di media sosial *Twitter*?

1.3 Batasan Masalah

Batasan masalah dalam melakukan proses pada penelitian ini yaitu:

1. Dataset komentar hanya menggunakan komentar berbahasa Indonesia.
2. Dataset yang dikumpulkan dari *twitter* hanya dalam bentuk teks.
3. *Tweet* yang digunakan sebagai data diambil dari *platform Twitter* dengan menggunakan *hashtag* “#ppkm”.
4. Periode pengambilan dataset diambil dari rentang waktu tanggal 1 april 2020 hingga 1 april 2022.

1.4 Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini yaitu:

1. Mengimplementasikan metode *eXtreme Gradient Boosting (XGBoost)* dalam melakukan analisis sentimen masyarakat terhadap penerapan kebijakan PPKM di media sosial *Twitter*.
2. Mengetahui performa pengujian pada analisis sentimen masyarakat terhadap penerapan kebijakan PPKM di media sosial *Twitter*.
3. Mengetahui sentimen mayoritas masyarakat Indonesia terhadap penerapan kebijakan PPKM di media sosial *Twitter*.
4. Mengetahui jumlah klasterisasi kata yang akan dicari yaitu positif dan negatif terhadap penerapan kebijakan PPKM di media sosial *Twitter*.

1.5 Manfaat Penelitian

Manfaat yang diberikan dari penelitian ini yaitu:

1. Menganalisa dan mengklasifikasikan sentimen masyarakat Indonesia di *Twitter* terhadap penerapan kebijakan PPKM yang dibuat oleh pemerintah ke dalam kategori positif dan negatif.
2. Menjadi referensi mahasiswa lain untuk memahami analisis sentimen dan metode *eXtreme Gradient Boost (XGBoost)*.

1.6 Sistematika Penulisan

Penyusunan tugas akhir ini berdasar dari sistematika penulisan berikut yaitu:

1. Bab I. Pendahuluan
Bab ini membahas tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, dan sistematika penulisan.
2. Bab II. Tinjauan Pustaka dan Dasar Teori
Bab ini memuat tentang tinjauan Pustaka yang menjabarkan hasil penelitian yang berkaitan dengan penelitian ini dan dasar teori yang menjabarkan teori-teori penunjang yang berhubungan dengan penelitian ini.
3. Bab III. Metode Perancangan
Memuat tentang metode perancangan, mulai dari pelaksanaan penelitian,

diagram alir penelitian, menentukan alat dan bahan, lokasi penelitian, dan Langkah-langkah penelitian.

4. Bab IV. Hasil dan Pembahasan

Memuat tentang hasil dan pembahasan yang diperoleh berdasarkan hasil pengukuran dan pelaksanaan.

5. Bab V. Penutup

Memuat tentang kesimpulan dan saran berdasarkan hasil pembahasan yang telah diperoleh.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Bryan Pratama, dkk. pada tahun 2019 melakukan studi analisis dengan judul “Sentiment Analysis Of The Indonesian Police Mobile Brigade Corps Based On Twitter Posts Using The SVM And NB Methods” pada studi tersebut dilakukan analisa pada *tweet – tweet* dengan kata kunci “Brimob” dimana total *tweet* yang digunakan sebanyak 1000 *tweets*. Studi ini menggunakan *text mining* dengan didukung oleh *support vector machine* (SVM) untuk mengklasifikasikan sentimen publik terhadap brimob di *twitter*. Akurasi yang diperoleh dengan SVM mencapai 86,96% sedangkan dengan *Naive Bayes* diperoleh akurasi sebesar 86,48% [14]

Tahun 2019, Eka dkk. melakukan studi analisis sentimen pada contoh Gojek dan Grab, menggunakan algoritma *Naive Bayes Classifier*, dan menemukan bahwa akurasi, *recall*, dan presisi metode *Naive Bayes Classifier* masing-masing adalah 72,33%, 73,95%, dan 73,24%. Penelitian tersebut kemudian dilanjutkan oleh (D. A. Al-Qudah et al.) melakukan penelitian analitik sentimen terhadap penyedia layanan *e-payment* menggunakan algoritma yang disebut *XGBoost* dan membandingkan hasilnya dengan J84, *Naive Bayes*, dan KNN. Akurasi maksimum didapatkan oleh KNN dan *XGBoost* yang masing-masing memiliki nilai *recall* 85,2 persen dan 82,8 persen. Sedangkan dengan menggunakan nilai presisi *Naive Bayes* didapatkan akurasi tertinggi sebesar 72 persen.

Tahun 2020, Dana A. Al- Qudah, dkk. dengan penelitian mereka berjudul “Sentiment Analysis for e-Payment Service Providers Using Evolutionary eXtreme Gradient Boosting” melakukan analisa pada pendapat pelanggan dari servis pembayaran elektronik melalui media sosial Arab. Dataset diperoleh dari *twitter* dan *facebook*, kemudian teknik ekstraksi fitur yang digunakan yaitu TF-IDF, dan akurasi yang diperoleh dari penggunaan metode *XGBoost* disini adalah 66,8%, lebih tinggi apabila dibandingkan dari tiga metode lainnya yang coba digunakan juga oleh penulis yaitu K-NN, J48, dan NB [14].

Terkait dengan sentimen analisis, beberapa penelitian telah dilakukan sebelumnya. Fajar Fathur Rachman pada tahun 2020, dalam penelitiannya yang berjudul “Analisis Sentimen Pro dan Kontra Masyarakat Indonesia tentang Vaksin COVID-19 pada media sosial *Twitter*” melakukan penelitian sentimen analisis dengan menggunakan algoritma *Latent Dirichlet Allocation* (LDA) untuk mengelompokkan opini masyarakat dengan tujuan mengetahui topik pembicaraan yang sering dibahas masyarakat terkait dengan wacana vaksinasi, hasil analisis menunjukkan bahwa masyarakat lebih banyak memberikan respon positif terhadap wacana tersebut (30%) dibandingkan dengan respon negatifnya (26%) [15].

Angelina Puput Giovani, dkk. Pada tahun 2020, dalam penelitian dengan judul “Analisis Sentimen Aplikasi Ruang Guru di *Twitter* Menggunakan Algoritma Klasifikasi” melakukan komparasi beberapa algoritma yaitu *Naive Bayes*, *Support Vector Machine*, dan *K-Nearest Neighbour* yang menggunakan *feature selection* dengan yang tidak menggunakan *feature selection*, serta juga membandingkan nilai *Area Under Curve* dari metode – metode tersebut untuk mengetahui algoritma mana yang paling optimal, hasil pengujian menunjukkan bahwa algoritma SVM dengan *feature selection* menjadi algoritma terbaik dengan nilai akurasi 78,55% dan AUC 0,853 [16].

Sulaiman Ainin, dkk. Pada tahun 2020, dengan penelitian berjudul “Sentiment Analyses Of Multilingual Tweets On Halal Tourism” menuliskan tentang penelitian yang mereka lakukan pada *tweet – tweet* dari rentang waktu 2008 - 2018 yang berkaitan dengan multilingual halal *tourism* dimana konten dan sentimen dari *tweet – tweet* tersebut dianalisa, mereka menggunakan 19 kata kunci untuk mengesktrak data dari *tweet* dimana 5 kata kunci tersebut adalah bahasa Malaysia, dan sisanya bahasa Inggris. Setelah dilakukan analisa diperoleh kesimpulan bahwa *tweet* terkait pariwisata halal pada negara non muslim melebihi jumlah *tweet* pada negara muslim, penelitian ini menunjukkan bahwa pariwisata halal mulai populer di negara seperti Inggris, Kanada, dan Spanyol [17].

Elena, Podasca pada tahun 2021 dengan penelitian berjudul “Predicting The Movement Direction Of OMXS30 Stock Index Using *XGBoost* and Sentiment Analysis” melakukan prediksi pada indeks pasar saham Swedia menggunakan

metode *XGBoost* yang disertakan dengan sentimen analisis dari berita keuangan guna membantu meningkatkan kinerja klasifikasi ketika memprediksi tren harga harian dari indeks pasar saham Swedia yaitu OMXS30. Hasil pada penelitian ini menunjukkan bahwa *XGBoost* memiliki kinerja yang baik dalam mengklasifikasikan tren hari OMXS30 dimana akurasi yang diperoleh mencapai 73% [18].

Pada tahun 2021. Aldiansyah Putra, dkk. dalam penelitiannya berjudul “Analisis Sentimen Masyarakat Terhadap Kebijakan PPKM Pada Media Sosial Twitter Menggunakan Algoritma SVM” melakukan penelitian terhadap respons masyarakat di *Twitter* berupa pro dan kontra mereka kepada kebijakan pemerintah dalam pemberlakuan pembatasan kegiatan masyarakat (PPKM), metode yang digunakan pada penelitian tersebut adalah *Support Vector Machine*, dengan memanfaatkan 3000 dataset yang kemudian diperoleh hasil akurasi sebesar 64%. Dari penelitian tersebut, algoritma SVM dapat mengenali *tweet* yang berisikan penolakan PPKM sebagai *tweet* bertendensi negatif dan juga kata – kata yang memiliki hubungan terhadap tendensi negatif tersebut [12].

Tabel 2.1 Penelitian sebelumnya

No.	Peneliti	Judul	Keterangan
1.	Bryan Pratama et al	Sentiment Analysis Of The Indonesian Police Mobile Brigade Corps Based On Twitter Posts Using The SVM And NB Methods	Menggunakan 1000 <i>tweets</i> dengan kata kunci “brimob”, akurasi yang diperoleh untuk masing – masing metode klasifikasi yaitu SVM senilai 86.96% dan <i>naïve baiyes</i> senilai 86,48%
2.	Eka et al	Analisis Sentimen Pada Contoh Gojek dan Grab	Melakukan studi analisis sentimen terhadap gojek dan grab menggunakan algoritma <i>naïve baiyes</i> . Akurasi, <i>recall</i> , dan presisi yang diperoleh yaitu

			72,33%, 73,95%, dan 73,24%.
3.	D. A. Al-Qudah et al.	Sentiment Analysis for e-Payment Service Providers Using Evolutionary eXtreme Gradient Boosting	Analisa pada pendapat pelanggan dari servis pembayaran elektronik melalui media sosial arab, dataset diperoleh dari <i>facebook</i> dan <i>twitter</i> , menggunakan ekstraksi fitur TF-IDF dan memperoleh akurasi senilai 66,8% dengan menggunakan algoritma <i>XGBoost</i> .
4.	Fajar Fathur Rachman	Analisis Sentimen Pro dan Kontra Masyarakat Indonesia tentang Vaksin COVID-19 pada Media Sosial <i>Twitter</i>	Analisis dilakukan dengan menggunakan algoritma Latent Dirichlet Allocation (LDA) dan dataset berupa <i>tweets</i> , hasil menunjukkan respon masyarakat terhadap wacana tersebut (30%) positif dibandingkan dengan respon negatifnya senilai (26%).
5.	Angelina Puput Giovani et al	Analisis Sentimen Aplikasi Ruang Guru Di Twitter Menggunakan Algoritma Klasifikasi	Penelitian ini menekankan komparasi antara metode Naive Bayes, Support Vector Machine, dan K-Nearest Neighbour dengan dan tanpa <i>feature selection</i> ,

			SVM dengan <i>feature selection</i> menghasilkan akurasi terbaik senilai 78,55%
6.	Sulaiman Ainin et al.	Sentiment Analyses Of Multilingual Tweets On Halal Tourism	Dataset merupakan <i>tweet</i> terkait dengan penelitian yang diambil dari rentang tahun 2008-2018, kesimpulan yang diperoleh bahwa <i>tweet</i> terkait pariwisata halal pada negara non muslim melebihi jumlah <i>tweet</i> pada negara muslim.
7.	Elena, Po- dasca	Predicting The Movement Direction Of OMXS30 Stock Index Using XGBoost and Sentiment Analysis	Melakukan prediksi indeks harga pasar saham Swedia, disertai dengan sentimen analisis berita keuangan, dengan menggunakan algoritma XGBoost, akurasi yang diperoleh mencapai 73%.
8.	Aldiansyah Putra et al.	Analisis Sentimen Masyarakat Terhadap Kebijakan PPKM Pada Media Sosial Twitter Menggunakan Algoritma SVM	Penelitian dilakukan dengan menggunakan algoritma SVM, dan <i>dataset</i> sebanyak 3000 <i>tweet</i> dimana akurasi yang diperoleh senilai 64%.

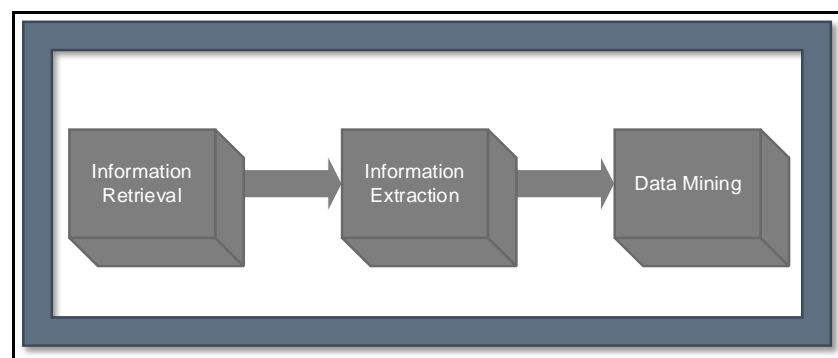
2.2 Teori Penunjang

Teori penunjang berisikan tentang konsep – konsep yang digunakan pada pembuatan dan perancangan sistem akan dibahas pada sub bab berikut :

2.2.1 Text Mining

Text mining merupakan proses penambangan teks yang menggunakan *computer* untuk mengestrak informasi secara otomatis dari berbagai sumber tertulis untuk menemukan informasi baru yang sebelumnya belum pernah ditemukan. Elemen kuncinya adalah dengan menghubungkan informasi yang telah dikumpulkan untuk menciptakan fakta baru atau hipotesis baru yang dapat diuji lebih lanjut dengan menggunakan algoritma komputasional [19].

Text mining merupakan bagian dari data *mining*, perbedaan mendasar dari *text mining* dan data *mining* adalah penambangan teks mengesktrak pola dari teks bahasa alami daripada dari *database* terstruktur yang berisi informasi *factual*. Teks ditulis untuk dibaca orang, sementara *database* dirancang agar program dapat diproses secara otomatis [19]. Untuk mengembangkan model yang belajar dari data pelatihan dan dapat mengantisipasi hasil pada informasi baru berdasarkan pengalaman dalam model pelatihan proses, penambangan teks menggabungkan teknik statistik, linguistik, dan pembelajaran mesin. Berikut adalah langkah – langkah yang terjadi pada *text mining*.



Gambar 2.1 Proses *text mining*

2.2.2 Sentimen Analisis

Sentimen analisis memiliki banyak sebutan, beberapa diantaranya merujuk pada nama – nama seperti subjektif analisis, penggalian opini, dan ekstraksi

penilaian dengan beberapa koneksi ke komputasi afektif [20]. Sentimen analisis adalah studi tentang opini dan sentimen serta evaluasi sikap, penilaian, dan perasaan yang dimiliki orang tentang hal-hal seperti produk, organisasi, isu, tema, dan fitur entitas.

Pada dasarnya sentimen analisis digunakan untuk menentukan opini yang ada pada teks dari suatu kalimat, apakah opini tersebut bersifat positif, negatif, atau netral [21]. Opini berada di pusat hampir semua aktivitas manusia karena mereka memiliki kekuatan untuk mengubah cara orang berperilaku. Berlawanan dengan pengetahuan faktual, opini dan sentimen sama-sama memiliki kualitas atau sifat yang unik karena keduanya subjektif. Karena sudut pandang satu orang hanya mewakili sudut pandang pribadi orang itu, yang seringkali tidak cukup untuk dijadikan dasar pengambilan keputusan, maka penting untuk mempertimbangkan pendapat banyak orang daripada hanya satu itu.

2.2.3 Web Crawling

Istilah *web crawling* atau *web scraping* sering digunakan untuk merujuk pada metode atau teknologi untuk mengumpulkan data yang dapat diakses *public* dari internet untuk fungsi tertentu. Meskipun informasi yang dikumpulkan dari internet seringkali beragam, namun jika dikompilasi dalam satu paket menggunakan metode ini, akan sangat membantu. Analisis sentimen adalah salah satu pengaplikasian dari *web crawling* yang mengidentifikasi perasaan orang tentang topik tertentu [22].

2.2.4 Tweet

Tweet atau dalam bahasa Indonesia disebut dengan kicauan. Pada *twitter*, *tweet* merupakan status yang berisikan tentang segala macam opini yang diberikan oleh suatu individu ataupun kelompok pada *twitter*, *tweet – tweet* tersebut dapat memuat sentimen berupa sentimen positif, negatif, ataupun netral. Sentimen – sentimen tersebut dapat digolongkan kategorinya berdasarkan makna yang tertulis pada sentimen tersebut.

2.2.5 Preprocessing

Teks *preprocessing* digunakan dalam penelitian ini untuk mempersiapkan data untuk analisis sentimen. Data yang diproses akan dikumpulkan dari teks – teks yang memberikan informasi tentang sentimen penulis, apakah itu positif atau negatif. Analisis sentimen terlebih dahulu harus dilakukan secara manual untuk menentukan apakah sebuah sentimen baik atau negatif dengan menganalisis maksud dari garis – garis dalam sentimen tersebut untuk mempermudah pengelolaan data [23]. Teks adalah data tidak terstruktur yang mungkin tidak tersedia dalam bentuk paling mentahnya untuk digunakan oleh program *computer* secara langsung. Selain itu, data teks tidak dapat dikenai operasi numerik. Akibatnya, teks harus diproses terlebih dahulu untuk menghasilkan data yang dapat digunakan dengan komputer. Terdapat beberapa langkah dasar yang dilakukan pada *text preprocessing*, berikut adalah:

2.2.5.1 Cleaning

Cleaning dilakukan untuk menghilangkan karakter, simbol, dan tanda baca yang tidak diperlukan dalam melakukan analisis sentimen, proses ini dilakukan karena data awal yang diperoleh merupakan data mentah yang memiliki banyak *noise* [24]. Proses ini nantinya dapat digabungkan pada saat proses *tokenization* dilakukan.

2.2.5.2 Casefolding

Untuk mempermudah sistem dalam mengenali setiap kata kemudian dalam proses pelatihan, *casefolding* mengubah semua karakter huruf besar dalam teks menjadi huruf kecil. Contoh kasus pada langkah *casefolding* yaitu ada pada proses untuk menghilangkan delimiter, delimiter dapat dianggap sebagai karakter selain huruf, dimana delimiter merupakan urutan satu karakter atau lebih yang dipakai untuk membatasi atau memisahkan data yang disajikan dalam *plain text* [25].

2.2.5.3 Tokenization

Tokenization adalah proses membagi aliran teks menjadi token, yang dapat berupa kata, frasa, simbol, atau komponen bermakna lainnya, kata – kata pada

kalimat yang dipisahkan oleh spasi akan diubah ke dalam bentuk *array* atau susunan kata[19]. Pada *tokenization*, setiap kata dapat ditentukan seberapa sering kata tersebut muncul, penentuan kemunculan frekuensi dari kata – kata tersebut dapat dilakukan dengan menggunakan penghitung frekuensi kemunculan kata [26].

2.2.5.4 Stopword Removal

Stopwords, juga dikenal sebagai *noise words*, *stopwords* adalah kata-kata yang mengandung sedikit informasi yang biasanya tidak diperlukan. Agar algoritma dapat fokus menemukan setiap kalimat, konsep, dan kata apa pun yang tidak terkait dengan nilai emosional, maka kata tersebut akan dihilangkan dari proses analisis sentimen. Untuk membuat proses pelatihan lebih efektif di kemudian hari, penghapusan *stopwords* melibatkan penghapusan konjungsi dan kata lain dari kalimat yang tidak memiliki arti yang sama dengan frasa.

2.2.5.5 Stemming

Stemming adalah metode memperoleh kata dasar dengan menghilangkan imbuhan seperti awalan, akhiran, dan awalan serta akhiran kalimat. *Stemming* merupakan salah satu fungsi krusial pada sistem dengan basis *Natural Language Processing* (NLP), tujuan utama dari fitur ini yakni untuk meningkatkan *recall* dari suatu algoritma yang digunakan dengan memproses akhiran kata secara otomatis dengan memecah kata menjadi akar kata. Peningkatan nilai *recall* dicapai tanpa mengorbankan akurasi pengambilan dokumen. Sebelum istilah indeks benar-benar ditetapkan ke indeks, *stemming* biasanya dilakukan dengan menghilangkan semua sufiks dan awalan (imbuhan) yang melekat [24].

2.2.6 Term Frequency – Relevance Frequency (TF-RF)

Term weighting merupakan metode yang digunakan untuk melakukan proses penghitungan bobot pada setiap *term* yang dicari pada setiap dokumen sehingga ketersediaan dan kemiripan dari suatu *term* di dalam dokumen dapat diketahui [27]. Pada penelitian ini, metode yang akan diterapkan yaitu metode TF-RF (*Term Frequency – Relevance Frequency*), metode ini diciptakan sebagai usaha dalam memperbaiki beberapa metode yang sudah ada.

Term Frequency (TF) adalah faktor yang menentukan bobot istilah dalam sebuah teks yang tergantung pada seberapa sering teks (*term*) tersebut muncul. Saat mengekspresikan suatu kata(*term*) maka frekuensi dari *term* tersebut akan dinilai. Bobot *term* pada dokumen atau nilai kesesuaian akan meningkat seiring dengan banyaknya kemunculan *term* tersebut pada dokumen. Persamaan dari metode ini adalah [19].

$$TF(d, t) = f(d, t) \quad (2.1)$$

Yaitu $f(d, t)$ merupakan frekuensi kemunculan *term* t pada dokumen d .

Pada *Relevance Frequency* (RF) yang merupakan metode yang diusulkan oleh Man Lan, frekuensi terhadap kemunculan *term* di kategori yang berkaitan dilihat sebagai pertimbangan relevansi dokumen [28]. Jadi pada TF-RF, bobot dari suatu *term* dihitung dengan menggunakan persamaan [28].

$$tf_{td}rf = tf_{td} * \log \left(2 + \frac{b}{\max(1, c)} \right) \quad (2.2)$$

Keterangan:

$tf_{td}rf$ = Pembobotan dokumen ke dalam model ruang vector

tf_{td} = Jumlah kemunculan kata t dalam dokumen

b = Jumlah dokumen yang mengandung kata t

c = Jumlah dokumen yang tidak mengandung kata t

2.2.7 XGBoost

eXtreme gradient boosting, disebut sebagai *XGBoost*, adalah algoritma berbasis *tree* yang termasuk ke dalam golongan algoritma *tree* yang sama dengan *decision tree* dan *random forest* [29]. Dengan bantuan prinsip *ensemble*, algoritma *supervised tree XGBoost* mengubah sejumlah set pembelajar yang lemah (pohon) menjadi model yang kuat sehingga dapat membuat prediksi yang akurat [30]. Dikarenakan fakta bahwa *XGBoost* dapat bekerja 10 kali lebih cepat dibanding dengan implementasi dari *gradient boosting* lainnya, banyak akademisi ataupun peneliti yang menerapkan algoritma ini untuk melakukan

klasifikasi dan regresi dalam berbagai situasi, termasuk prediksi penjualan, prediksi perilaku pelanggan, prediksi iklan, dan prediksi teks web [31].

Metode menambahkan model baru ke pendekatan *ensemble* disebut *boosting*, hal ini dilakukan untuk mengoreksi kesalahan dari model sebelumnya. Model akan ditambahkan satu per satu sampai tidak ada lagi peningkatan yang mungkin dilakukan. Teknik *ensemble* menggunakan model pohon klasifikasi dan regresi yang disebut *tree ensemble models*. Strategi yang dikenal dengan teknik *ensemble* menggabungkan prediksi dari berbagai *tree* menjadi satu [31]. Ini berusaha untuk secara berurutan memodelkan setiap *predictor* menggunakan kesalahan residual dari model sebelumnya. Ketika *dataset* dimasukkan, langkah pertama adalah menggunakan dataset yang dipilih untuk membangun model awal. Persamaan 2.3 dan 2.5 kemudian digunakan untuk menentukan nilai prediksi awal dan kesalahan residual dari model asli. Model pertama dibuat menggunakan persamaan nomor 2.3, sedangkan model berikutnya dibuat menggunakan persamaan 2.4.

$$h_0(x) = \text{mean}(Y) \quad (2.3)$$

$$\hat{Y} = Y - h_0(x) \quad (2.4)$$

Dimana Y merepresentasikan nilai residual *error* model awal dan $h_0(x)$ merepresentasikan nilai prediksi awal dari model pertama. Model kedua kemudian akan dibuat menggunakan residual *error* dari model pertama untuk menentukan nilai prediksinya. Kesalahan residual dari model pertama dan kedua kemudian akan digunakan untuk membuat model ketiga untuk menentukan nilai prediksinya. Sebanyak $n_{\text{estimator}}$ ditetapkan, maka proses ini akan terus berulang [31].

XGBoost menghasilkan satu set *decision tree* yang mana setiap model pohon bergantung pada pohon sebelumnya. Nilai prediksi awal untuk model pertama di *XGBoost* akan lemah, tetapi karena lebih banyak model dibangun, bobot diperbarui untuk menghasilkan prediksi yang lebih kuat. Untuk meminimalkan fungsi tujuan, nilai proyeksi dari masing – masing model akan dijumlahkan kemudia dimasukkan ke dalam persamaan nomor 2.5 [31].

$$L^{(t)} = \sum_{i=1}^n l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \quad (2.5)$$

Dimana n adalah jumlah model yang akan digunakan, l adalah fungsi untuk mengukur selisih antara target y_i dan \hat{y}_i yang diprediksi. $f_t(x_i)$ adalah model baru yang dibangun. Sedangkan Ω adalah fungsi untuk membuat model terhindar dari *overfitting*. Persamaan nomor 2.5 digunakan untuk mencari nilai keseluruhan.

2.2.8 Confusion Matrix

Confusion Matrix adalah metode untuk membandingkan nilai nyata dan yang diantisipasi untuk mengevaluasi keefektifan model pembelajaran mesin dalam prediksi label. Sebuah tabel yang disebut *confusion matrix* memiliki empat set terpisah dari kombinasi nilai yang diharapkan dan yang sebenarnya. Empat istilah — *true positive*, *false negative*, *true negative*, dan *false negative* digunakan dalam *confusion matrix* untuk menunjukkan hasil operasi kategorisasi. Selanjutnya penulis akan merancang metode *XGBoost* dengan memanfaatkan skor keempat item tersebut sebagai input analitis untuk menentukan nilai akurasi, presisi, *recall*, dan *f1score*. Menghitung akurasi, *recall*, dan presisi merupakan salah satu metode yang digunakan untuk menilai klasifikasi. Dalam metode ini, *confusion matrix* berfungsi sebagai panduan perhitungan.

Tabel 2.2 Tabel *confusion matrix*

Kelas		Prediksi	
		1	0
Kelas Sebenarnya	1	TP	FN
	0	FP	TN

Pada *matrix* diatas dapat dinyatakan sebagai berikut :

- True Positive* (TP), adalah jumlah dokumen dari kelas 1 yang benar diklasifikasikan sebagai kelas 1.
- False Positive* (FP), adalah jumlah dokumen dari kelas 0 yang salah diklasifikasikan sebagai kelas 1.
- False Negative* (FN), adalah jumlah dokumen dari kelas 1 yang salah diklasifikasikan sebagai kelas 0.

- d. *True Negative* (TN), adalah jumlah dokumen dari kelas 0 yang benar diklasifikasikan sebagai kelas 0.

Penghitungan akurasi dilakukan dengan menghitung jumlah prediksi benar yang kemudian dibagi dengan jumlah prediksi, berikut pada 2.6 adalah persamaan dari penghitungan akurasi :

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} * 100\% \quad (2.6)$$

$$precision = \frac{TP}{TP + FP} * 100\% \quad (2.7)$$

$$recall = \frac{TP}{TP + FN} * 100\% \quad (2.8)$$

Pada persamaan 2.7 dinyatakan sebagai *precision* yang dimana *precision* adalah tingkat keakuratan antara data yang diminta dengan hasil prediksi yang diberikan oleh model. Sedangkan pada persamaan 2.8 ditunjukkan *recall* yang merupakan tingkat keberhasilan model dalam menemukan Kembali sebuah informasi.

BAB III

METODOLOGI PENELITIAN

3.1 Alat dan Bahan

Berisi alat dan bahan yang akan digunakan untuk penelitian.

3.1.1 Alat Penelitian

Dalam penelitian tentang analisis sentimen masyarakat Indonesia di media sosial *twitter* terhadap kebijakan pemerintah dalam penerapan PPKM di Indonesia, digunakan beberapa alat yang terdiri dari perangkat keras dan perangkat lunak. Alat – alat tersebut adalah sebagai berikut.

a. Perangkat Keras

Perangkat keras yang digunakan dalam penelitian adalah satu unit komputer dengan spesifikasi berikut :

1. *Processor Intel® Core™ i5-7400 3,50 GHz*
2. *Memori RAM DDR4 32GB*
3. *Kartu grafis nvidia RTX 2070 8GB VRAM*

b. Perangkat Lunak

Perangkat lunak yang digunakan dalam penelitian ini adalah sebagai berikut.

1. *Sistem operasi Windows 10 Pro*
2. *Jupyter Notebook*
3. *Visual Studio Code*
4. *Bahasa Pemrograman Python versi 3.9*
5. *Microsoft Office*

3.1.2 Bahan Penelitian

Bahan penelitian yang digunakan dalam penelitian tentang analisis sentimen masyarakat Indonesia pada media sosial *twitter* terhadap kebijakan pemerintah dalam melaksanakan penerapan PPKM di Indonesia menggunakan *XGBoost* ini adalah *tweet* masyarakat Indonesia yang berisikan opini terhadap kebijakan pemerintah dalam penerapan PPKM. Pada penelitian ini, *tweet* yang

diperoleh merupakan *tweet* dari rentang waktu bulan April 2020 hingga April 2022 dengan total 20.000 *tweet*, dari total *tweet* - *tweet* tersebut, mengacu pada penelitian sebelumnya yang juga meneliti terkait sentimen masyarakat terhadap kebijakan PPKM pada media sosial twitter dimana penelitian tersebut menggunakan 3000 *tweet* dari total 5000 *tweet* yang diperoleh [12], maka pada penelitian ini penulis mencoba dengan menggunakan *tweet* sejumlah 10.000 *tweet* dengan asumsi bahwa dari total 20.000 *tweet* yang diperoleh, tidak semua *tweet* – *tweet* tersebut memiliki kualitas yang baik untuk dijadikan sebagai *dataset*, jadi disini penulis menarik kesimpulan untuk menggunakan *tweet* sebanyak 10.000 yang didasarkan pada penelitian sebelumnya dan asumsi dari penulis terkait dengan *tweet* – *tweet* dengan kualitas buruk. *Tweet* nantinya diberi label positif dan negatif secara manual oleh 3 orang Warga Negara Indonesia (WNI) yang memiliki rentang umur dari 21 hingga 24 tahun dimana mereka dengan sukarela membantu proses pelabelan *tweet* ini untuk menghindari bias apabila pelabelan dilakukan oleh penulis itu sendiri. *Tweet* – *tweet* tersebut nantinya digunakan sebagai *training* dan *testing* dataset. Berikut merupakan beberapa contoh dari *tweet* yang telah diverifikasi dan disetujui kategori sentimennya oleh 3 orang WNI yang bersangkutan pada saat proses pelabelan :

a. *Tweet Positive*

1. Jaga selalu imunitas dengan mengonsumsi makanan bergizi seimbang, rutin berolahraga, tidur cukup, menghindari stres, dan #vaksindulu bagi yang belum. #revolusimental #ppkm #covid-19 #coronavirus.
2. Presiden berdoa agar masyarakat Indonesia selalu diberikan kesehatan dan kekuatan dalam menghadapi #PPKM yang sedang diberlakukan #ppkm #covid-19 #coronavirus #indonesia #news.
3. PPKM memberikan manfaat yang baik, ini terbukti dengan berkurangnya angka penyebaran dan kematian yang

sebelumnya cukup tinggi, semoga rakyat Indonesia tetap diberikan kesehatan #sehat #ppkm #indonesiamaju.

b. *Tweet Negative*

1. Negara apa sih ini...ppkm mulu otaknya pejabat udah modar apa ya...minyak naik...kebutuhan naik usaha di pampet lagi...Ga lo ppkm aja kami kerepotan tambah lagi #ppkm
2. Aturan PPKM itu tidak jelas kalo kita tidak melawan makin tidak jelas kebijakan-kebijakan pemerintah #ppkm
3. Pemerintah enggak jelas, seenaknya saja memberlakukan aturan kami rakyat kecil kesusahan, dasar rezim dzalim. #ppkm #revolusigagal

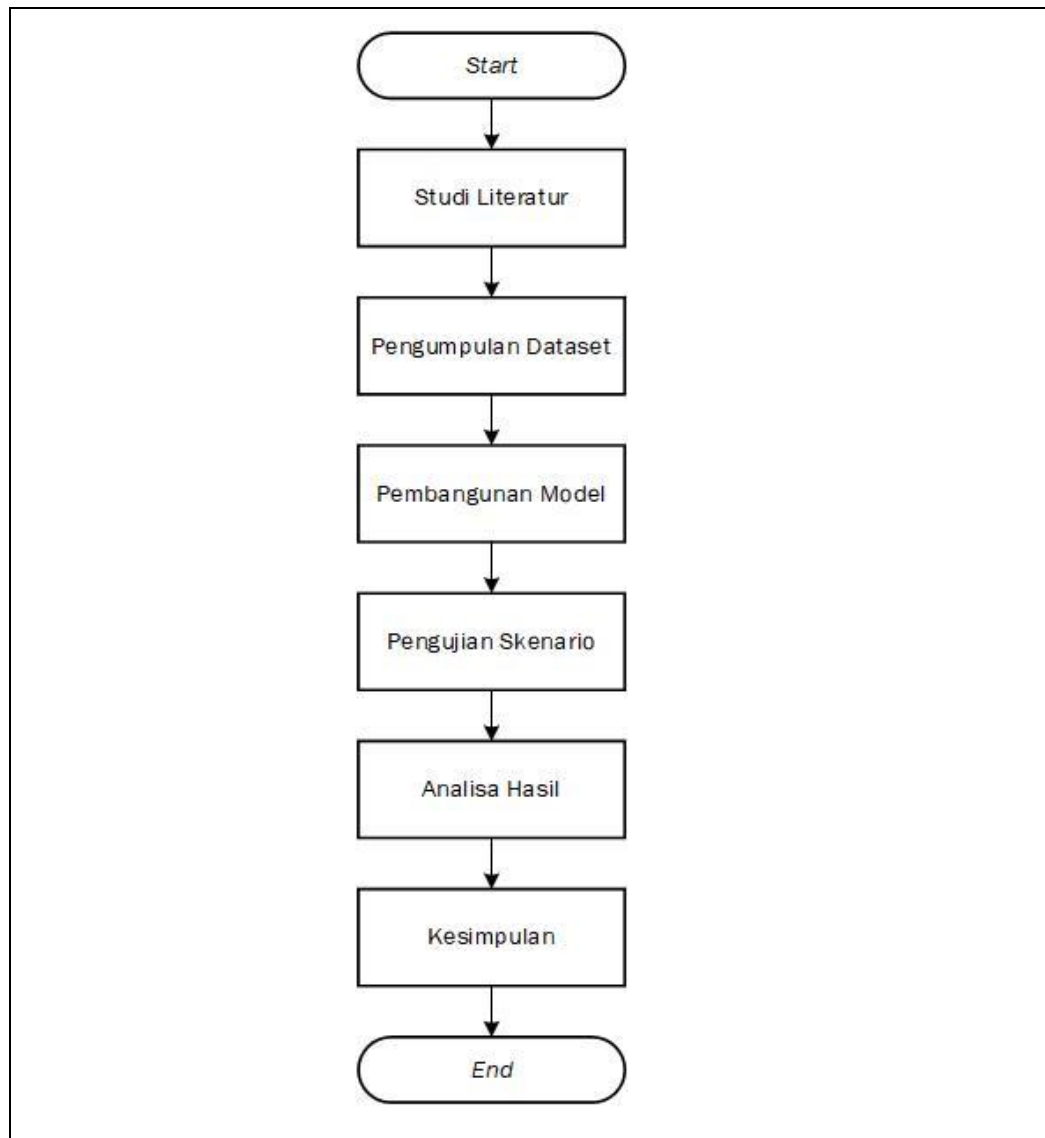
3.2 Studi Literatur

Studi literatur yang dilakukan dengan tujuan untuk mendukung penelitian yaitu mempelajari buku elektronik, jurnal – jurnal penelitian, serta berbagai sumber lainnya yang berkaitan dengan topik penelitian, yaitu analisis sentimen dan *web crawling*. Lebih spesifik, materi yang dipelajari adalah *text preprocessing*, *natural language processing*, analisis sentimen serta pemanfaatan metode *XGBoost* dalam melakukan analisis sentimen. Jurnal – jurnal yang dipelajari membahas berbagai studi kasus tentang analisis sentimen dengan metode *XGBoost*.

3.3 Alur Penelitian

Analisis sentimen masyarakat Indonesia pada media sosial *twitter* terhadap kebijakan pemerintah dalam melaksanakan penerapan PPKM di Indonesia menggunakan *XGBoost* dilakukan dalam beberapa tahapan. Tahap pertama yang dilakukan yakni melakukan studi literatur untuk mendapatkan pengetahuan serta gambaran akan penelitian yang dilakukan. Literatur yang dipelajari berupa jurnal penelitian serta buku yang membahas tentang *web crawling* dan analisis sentimen menggunakan metode *XGBoost*. Kemudian dilakukan pengumpulan

dataset, dataset yang dikumpulkan merupakan sejumlah *tweets* dengan tagar PPKM sebanyak 20.000 *tweet*, dari 20.000 *tweet* tersebut, diambil 10.000 *tweet* sebagai dataset, setelah dilakukan pengumpulan, maka *tweet – tweet* tersebut diberi label, yaitu positif dan negatif. Tahapan selanjutnya yaitu dilakukan pembangunan model yang sesuai dengan literatur yang telah dipelajari, lalu pengujian terhadap model dilakukan untuk mengetahui apakah model yang dibangun telah mendapatkan hasil yang sesuai. Setelah dilakukan beberapa skenario, dilakukan analisa terhadap hasil yang diperoleh dari berbagai macam skenario yang telah direncanakan. Penelitian dilanjutkan dengan pengambilan kesimpulan dari penelitian yang telah dilakukan dari awal hingga hasil akhir yang didapatkan. Diagram alir penelitian dapat dilihat pada Gambar 3.1.



Gambar 3.1 Alur penelitian

3.4 Kebutuhan Sistem

Dalam penelitian tentang analisis sentimen masyarakat terhadap kebijakan pemerintah dalam penerapan PPKM di Indonesia pada media sosial *Twitter* menggunakan metode *XGBoost*, analisis kebutuhan sistem dibagi menjadi 3 jenis yaitu analisis pengguna, analisis perangkat keras dan analisis perangkat lunak yang digunakan dalam penelitian.

3.4.1 Analisis Pengguna

Pengguna dari sistem ini adalah orang – orang atau peneliti yang akan melakukan penelitian terkait dengan analisis sentimen di masa yang akan datang, khususnya mereka yang mengangkat topik serupa dengan penelitian ini ataupun mereka yang menggunakan metode serupa sehingga pada penelitian ini, orang – orang tersebut dapat menggunakan penelitian ini sebagai landasan teori pada penelitian mereka selanjutnya, maupun sebagai sumber referensi pustaka. Selain itu juga, hasil dari penelitian ini dapat digunakan oleh orang – orang seperti *developer* suatu aplikasi, apabila mereka membutuhkan suatu model klasifikasi sentimen untuk membangun aplikasi yang mereka buat.

3.4.2 Analisis Perangkat Keras

Perangkat keras yang digunakan dalam pembangunan sistem, pelatihan data, serta pengujian sistem merupakan elemen penting dalam penelitian ini. Perangkat keras yang mumpuni dapat membantu mempercepat proses-proses yang dilakukan seperti pelatihan data yang membutuhkan sumber daya cukup tinggi. Perangkat keras yang digunakan dalam penelitian ini memiliki spesifikasi seperti yang terdapat pada Tabel 3.1.

Tabel 3.1 Kebutuhan perangkat keras

No.	Nama Perangkat	Spesifikasi
1.	<i>Processor</i>	Intel® Core™ i5-7400 3,50 GHz
2.	Memori	Memori RAM 32GB DDR4
3.	GPU	NVIDIA RTX 2070
4.	<i>Storage</i>	256GB SSD, 1TB HDD, 500GB M.2

3.4.3 Analisis perangkat lunak

Selain perangkat keras, perangkat lunak juga memiliki peranan penting dalam proses pengembangan sistem. Penggunaan perangkat lunak yang tepat dapat membantu mempercepat proses penelitian. Perangkat lunak yang digunakan dalam penelitian ini dapat dilihat pada Tabel 3.2.

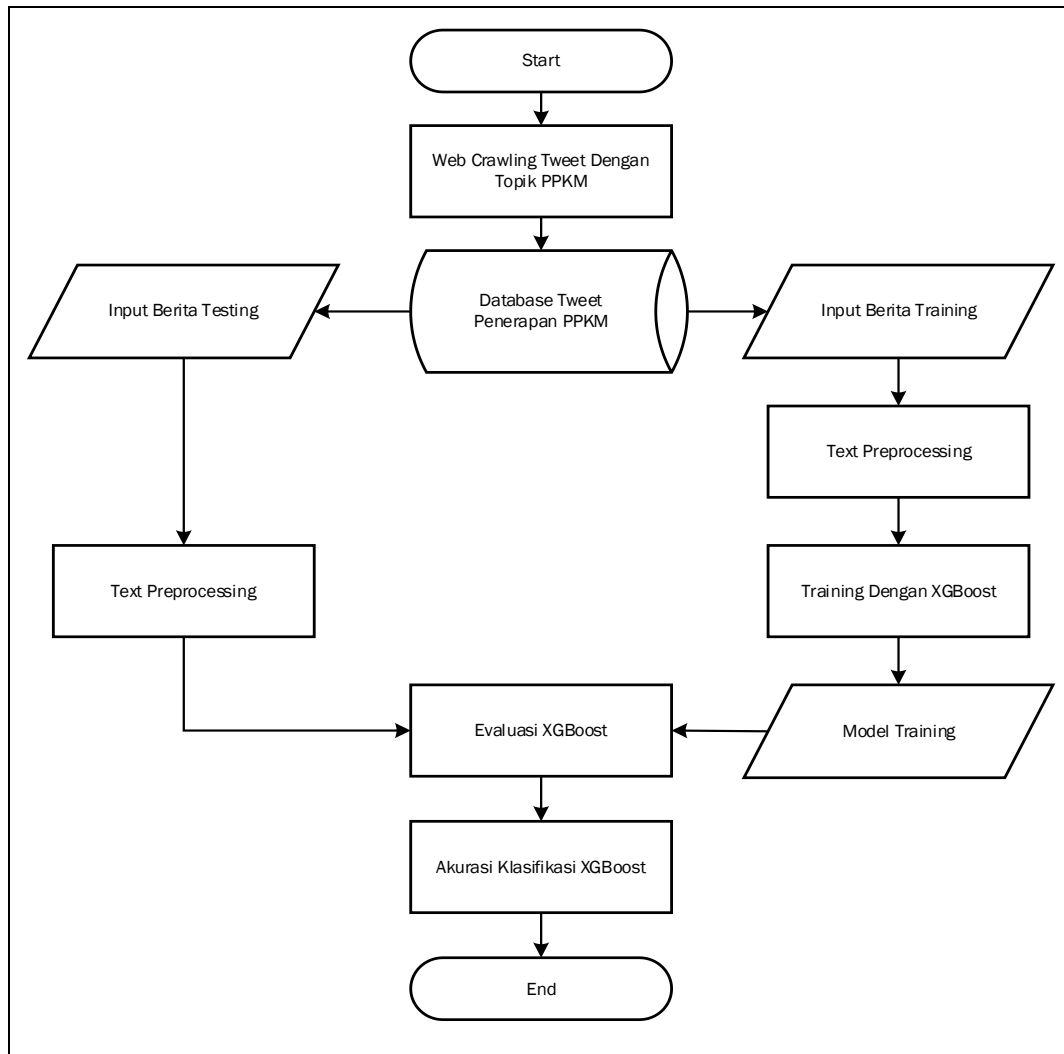
Tabel 3.2 Kebutuhan perangkat lunak

No.	Nama Perangkat	Spesifikasi
1.	Sistem Operasi	Windows 10
2.	<i>Text Editor</i>	<i>Jupyter Notebook</i>
3.	<i>Microsoft Office</i>	<i>Ms.Office Professional Plus 2019</i>
4.	Bahasa pemrograman <i>python</i>	<i>Python 3.9.6</i>
5.	<i>Library NLTK</i>	<i>Python nltk 1.1.2</i>
6.	<i>Library scikit learn</i>	<i>Python scikit learn 0.23.2</i>
7.	<i>Library Sastrawi</i>	<i>Python Sastrawi 1.0.1</i>
8.	<i>Library Scraping</i>	<i>Python snsrape 0.4.3.20220106</i>
9.	<i>Web Browser</i>	<i>Google Chrome</i>
10.	<i>Library Pandas</i>	<i>Python pandas 1.4.4</i>
11.	<i>Library Regex</i>	<i>Python regex 2022.8.17</i>
12.	<i>Library xgboost</i>	<i>Python xgboost 1.7.5</i>
13.	<i>Library nlpaug</i>	<i>Python nlpaug 1.1.11</i>
14.	<i>Library transformers</i>	<i>Python transformers 4.29.2</i>
15.	<i>Library wordcloud</i>	<i>Python wordcloud 1.8.2.2</i>

3.5 Perancangan Sistem

Rancangan dari sistem analisis sentimen komentar masyarakat Indonesia pada media sosial *twitter* terhadap kebijakan pemerintah Indonesia dalam penerapan PPKM dengan menggunakan algoritma *eXtreme Gradient Boost (XGBoost)* yang terdiri dari beberapa tahapan, yang dapat dilihat pada Gambar 3.2. Alur dari perancangan sistem tersebut dimulai dari tahapan *crawling tweet – tweet* yang memiliki tagar #ppkm dimana nantinya *tweet – tweet* tersebut digunakan sebagai dataset pada sistem, *tweet – tweet* yang berisi opini *netizen* Indonesia terkait dengan penerapan kebijakan PPKM yang dilakukan oleh pemerintah diambil dari *database twitter* dan *tweet – tweet* tersebut memiliki beberapa kriteria yakni *tweet* memuat tagar #ppkm, kemudian *tweet* haruslah menggunakan bahasa Indonesia, dan *tweet – tweet* tersebut adalah *tweet* yang dibuat dalam rentang waktu yang dimulai dari tanggal 1 April 2020 hingga 1 April 2022, kemudian *tweet* dibagi menjadi 2 jenis dataset, yaitu *tweet* yang digunakan sebagai data *training* dan *tweet* yang digunakan

sebagai data *testing*, selanjutnya pada *tweet – tweet* tersebut dilakukan *preprocessing* agar *tweet – tweet* nantinya menjadi lebih relevan pada saat memasuki proses *training* oleh model, setelah dilakukan *training* model, maka dilanjutkan dengan mengevaluasi model tersebut dengan *testing* data untuk memperoleh tingkat akurasi dari model yang telah dilatih.



Gambar 3.2 Perancangan sistem

3.5.1 Web Crawling Twitter

Pada tahap ini, *tweet* dikumpulkan melalui jejaring media sosial *Twitter*. *Tweet* yang dikumpulkan adalah *tweet* yang menggunakan *hashtag* “#ppkm”. “#ppkm” kemudian dimasukkan pada *query* pencarian *tweets* yang digunakan oleh *library snsrape* untuk melakukan *crawling* data pada *tweet – tweet* berbahasa Indonesia yang memuat tagar “ppkm” di dalamnya. Pada hasil pencarian yang telah

dilakukan, didapati sebanyak 20000 *tweet* berbahasa Indonesia, yang membahas terkait kebijakan pemerintah Indonesia dalam penerapan PPKM. *Tweet – tweet* tersebut berisi berbagai macam jenis sentimen yang terkandung di dalamnya, dari sentimen positif, netral, dan negatif, namun pada penelitian ini kategori sentimen yang diambil hanya berupa sentimen positif dan negatifnya saja. *Tweet – tweet* yang telah dikumpulkan tersebut nantinya digunakan sebagai data latih dan data uji pada program untuk memperoleh akurasi terhadap bagaimana sentimen sentimen masyarakat Indonesia secara keseluruhan terkait dengan kebijakan pemerintah Indonesia dalam melakukan penerapan PPKM melalui pengujian yang dilakukan dengan menerapkan algoritma *XGBoost*.

3.5.2 Input Dataset Tweet Training dan Testing

Pada tahap ini, *tweet* yang telah diperoleh dari laman media sosial *twitter* yang telah dimuat sebagai dataset akan dibagi menjadi 2 kategori yaitu *tweet – tweet* yang digunakan sebagai data *training* dan data *testing*. *Tweet* yang digolongkan sebagai data *training* digunakan untuk membuat model klasifikasi sedangkan *tweet* yang digunakan sebagai data *testing* digunakan untuk menguji model yang telah dibuat.

a. Input Tweet Training

Tweets training yang sebelumnya telah diperoleh melalui proses *scraping* dari media sosial *Twitter* yang kemudian dimuat ke dalam dataset dengan ekstensi *.csv* akan dimasukkan ke dalam sistem untuk diproses. *Tweet* yang diperoleh merupakan *tweet – tweet* berbahasa Indonesia, dimana *tweet* tersebut memuat opini masyarakat Indonesia tentang kebijakan penerapan PPKM yang dilakukan pemerintah, *tweet – tweet* tersebut ditandai sedemikian dikarenakan memuat tagar “#ppkm” pada penulisannya. Selanjutnya dilakukan *preprocessing* pada seluruh *tweet – tweet* yang telah dimuat sebagai data *training*, yang kemudian di-*training* menggunakan algoritma *XGBoost*.

Contoh *tweet* yang digunakan sebagai *tweet training* pada sistem dapat dilihat pada Tabel 3.3.

Tabel 3.3 *Tweet training*

<i>Date</i>	<i>Username</i>	<i>Tweets</i>
2021-10-25 07:34:41+00:00	nuchillinaris	"Sebab, program penanggulangan #Covid19 dirasakan oleh masyarakat bawah. Pun dg program pemulihan ekonomi sangat membumi & dirasakan benar oleh masyarakat yg perekonomiannya sangat terdampak akibat kebijakan #PPKM. #7ThJokowiLuarBiasa Jokowi diakui dunia! https://t.co/ATrYbGU7px "
2022-03-22 17:25:29+00:00	ViantAntony	Ruwet Ruwet Ruwet inilah Negeri RuwetNesia. Hebatnya Virus itu adalah dia tau Ramadhan akn datang me-raka akn meperbanyak bhkn #PPKM kemungkinan di perpanjang. Yakan pak @KemenkesRI ??? https://t.co/m0wm0fkHUW
2022-03-22 15:23:09+00:00	LaNyallaMM1	"Saya berharap pelonggaran aktivitas bukan hanya untuk menggenjot perekonomian. Tetapi juga dimanfaatkan sektor pendidikan untuk meningkatkan Sumber Daya Manusia yang sedikit mundur karena pandemi.

		@JatimPemprov #LaNyalla #ketuadpdri #dpdri #daridaerahuntukindonesia #ppkm"
--	--	---

3.5.3 Text Preprocessing Tweets Dataset

Text preprocessing yang dilakukan pada penelitian ini dibagi menjadi 3 tahap, yaitu tahap *casefolding*, *tokenization*, *stemming*, dan *stop-word removal*.

a. Casefolding

Casefolding merupakan tahapan pertama yang dilakukan pada *preprocessing text*, pada tahap ini *dataset* yang ada akan disamaratakan penggunaan huruf kapitalnya, yang dimana pada *dataset* ini, seluruh *tweet* akan diubah hurufnya menjadi huruf kecil, ini bertujuan agar *tweets* menjadi konsisten pada penggunaan hurufnya, dan mencegah sistem mengalami kebingungan dikarenakan kata yang sama apabila penulisan hurufnya berbeda, maka kata tersebut akan dianggap sebagai kata yang berbeda oleh sistem.

Tabel 3.4 *Tweet casefolding*

<i>Date</i>	<i>Username</i>	<i>Tweets</i>
2021-10-25 07:34:41+00:00	nuchillinaris	"sebab, program penanggulangan #covid19 dirasakan oleh masyarakat bawah. pun dg program pemulihan ekonomi sangat membumi & dirasakan benar oleh masyarakat yg perekonomiannya sangat terdampak akibat kebijakan #ppkm. #7thjokowiluarbiasa

		jokowi diakui dunia! https://t.co/atrybgu7px
2022-03-22 17:25:29+00:00	ViantAntony	ruwet ruwet ruwet inilah negeri ruwetnesia. hebatnya virus itu adalah dia tau ramadhan akn datang meraka akn meperbanyak bhkn #ppkm kemungkinan di perpanjang. yakan pak @kemenkesri ??? https://t.co/m0wm0fkhuw
2022-03-22 15:23:09+00:00	LaNyallaMM1	"saya berharap pelanggaran aktivitas bukan hanya untuk menggenjot perekonomian. tetapi juga dimanfaatkan sektor pendidikan untuk meningkatkan sumber daya manusia yang sedikit mundur karena pandemi. @jatimpemprov #lanyalla #ketuadpdri #dpdri #dari-daerahuntukindonesia #ppkm"

b. Tokenization

Tokenization merupakan untuk mentransformasikan *tweets* menjadi kumpulan kata yang disebut *terms*. Pada *tokenization* juga dilakukan penghilangan tanda baca. Hal ini dilakukan karena tanda baca tidak dapat digunakan sebagai *terms* karena terdapat pada hampir seluruh dokumen. Sebelum proses *tokenization*, terlebih dahulu dilakukan proses *case folding* atau mengubah setiap kata menjadi huruf kecil. Karakter selain huruf dihilangkan dan dianggap *delimiter*.

Tujuannya adalah agar tidak terjadi kesalahan interpretasi oleh komputer ketika ada dua kata yang sama tapi dianggap berbeda karena perbedaan huruf besar dan huruf kecil. Contoh *tweets* yang telah melewati proses *tokenization* dapat dilihat pada Tabel 3.5.

Tabel 3.5 *Tweet tokenization*

<i>Date</i>	<i>Username</i>	<i>Tweet_Tokens</i>
2021-10-25 07:34:41+00:00	nuchillinaris	'sebab', 'program', 'penanggulangan', 'dirasakan', 'oleh', 'masyarakat', 'bawah', 'pun', 'dg', 'program', 'pemulihan', 'ekonomi', 'sangat', 'membumi', 'amp', 'dirasakan', 'benar', 'oleh', 'masyarakat', 'yg', 'perekonomiannya', 'sangat', 'terdampak', 'akibat', 'kebijakan', 'jokowi', 'diakui', 'dunia'
2022-03-22 17:25:29+00:00	ViantAntony	'ruwet', 'ruwet', 'ruwet', 'inilah', 'negeri', 'ruwetnesia', 'hebatnya', 'virus', 'itu', 'adalah', 'dia', 'tau', 'ramadhan', 'akn', 'datang', 'meraka', 'akn', 'meperbanyak', 'bhkn', 'kemungkinan', 'di', 'perpanjang', 'yakan', 'pak'
2022-03-22 15:23:09+00:00	LaNyallaMM1	'saya', 'berharap', 'pelonggaran', 'aktivitas', 'bukan', 'hanya', 'untuk', 'menggenjot', 'perekonomian', 'tetapi', 'juga', 'dimanfaatkan', 'sektor', 'pendidikan', 'untuk', 'meningkatkan', 'sumber', 'daya', 'manusia', 'yang', 'sedikit', 'mundur', 'karena', 'pandemi'

c. *Stopword Removal*

Pada tahap ini *stopwords* pada *tweet* akan dihapus guna meningkatkan keefektifan proses *training* di kemudian hari, *stopwords* adalah kata – kata pada bidang NLP (*Natural Language Processing*) yang dinyatakan memiliki sedikit makna, bahkan hampir tidak bermakna, kata – kata tersebut seperti ‘yang’, ‘yaitu’, ‘di’, ‘tempat’, ‘terus’, ‘walau’, dan masih banyak lainnya. Pada *machine learning* maupun *deep learning*, *stopword* biasanya dihapus terlebih dahulu sebelum proses pelatihan dilakukan dikarenakan *stopword* cenderung muncul dalam jumlah banyak, dimana hal tersebut berdampak pada tidak adanya informasi unik yang diberikan oleh *stopword* – *stopword* untuk dapat digunakan pada proses klasifikasi atau *clustering*. Pada penelitian ini *sample stopwords* diperoleh dari yang sudah disediakan pada *library* NLTK, *stopwords* – *stopwords* yang ada pada *library* tersebut kemudian penulis coba gabungkan dengan beberapa *stopwords* yang penulis cenderung temukan pada *tweet* – *tweet* yang digunakan. Beberapa contoh dari *stopwords* tersebut tersaji di tabel ini.

<p>['ada', 'adalah', 'adanya', 'adapun', 'agak', 'agaknya', 'agar', 'akan', 'akankah', 'akhir', 'akhiri', 'akhirnya', 'aku', 'akulah', 'amat', 'amatlah', 'anda', 'andalah', 'antar', 'antara', 'antaranya', 'apa', 'apaan', 'apabila', 'apakah', 'apalagi', 'apatah', 'artinya', 'asal', 'asalkan', 'atas', 'atau', 'ataukah', 'ataupun', 'awal', 'awalnya',...-n]</p>

Sedangkan *tweet* – *tweet* yang sudah dihilangkan *stopwordnya* dapat dilihat pada tabel berikut,, dapat dibandingkan pada tahap sebelumnya, kata seperti ‘sebab’, ‘pun’, ‘benar’ dihilangkan pada *tweet* karena kata – kata tersebut tergolong ke dalam *stopwords*.

Tabel 3.6 *Tweet stopwords removal*

<i>Date</i>	<i>Username</i>	<i>Tweet_WSW</i>
2021-10-25 07:34:41+00:00	nuchillinaris	'program', 'penanggulangan', 'dirasakan', 'masyarakat', 'dg', 'program', 'pemulihan', 'ekonomi', 'membumi', 'amp', 'dirasakan', 'masyarakat', 'yg', 'perekonomiannya', 'terdampak', 'akibat', 'kebijakan', 'jokowi', 'diakui', 'dunia'
2022-03-22 17:25:29+00:00	ViantAntony	ruwet', 'ruwet', 'ruwet', 'negeri', 'ruwet-nesia', 'hebatnya', 'virus', 'tau', 'ramadhan', 'akn', 'meraka', 'akn', 'meperbanyak', 'bhkn', 'perpanjang', 'yakan'
2022-03-22 15:23:09+00:00	LaNyallaMM1	'berharap', 'pelonggaran', 'aktivitas', 'menggenjot', 'perekonomian', 'dimanfaatkan', 'sektor', 'pendidikan', 'meningkatkan', 'sumber', 'daya', 'manusia', 'mundur', 'pandemi'

d. *Stemming*

Teknik dalam memperoleh kata dasar atau dalam artian lain *stem* dari suatu kata pada suatu kalimat disebut dengan nama *stemming*. Pada proses tersebut, dilakukan pemotongan pada imbuhan (*affix*) kata, baik itu *prefix* maupun *suffix* [32]. Proses *stemming* dilakukan dengan menggunakan algoritma Nazief dan Adriani karena *tweet* yang digunakan pada penelitian merupakan *tweet* berbahasa Indonesia, selain itu juga algoritma ini memiliki tingkat presisi yang lebih baik dibandingkan algoritma lainnya seperti algoritma Porter. Algoritma *stemming* antara suatu bahasa dengan bahasa yang lain memiliki perbedaan, ini dikarenakan morfologi yang berbeda antara suatu bahasa, seperti morfologi bahasa Indonesia apabila disandingkan dengan bahasa Inggris, contoh kasusnya

adalah pada teks berbahasa Inggris, hanya diperlukan menghilangkan *sufiks* pada suatu teks untuk memperoleh *root word* dari sebuah kata. Sementara pada bahasa Indonesia, proses yang terjadi lebih kompleks, ini dikarenakan terdapat beberapa variasi imbuhan yang harus dibuang agar memperoleh *root word* dari sebuah kata. [32]. Algoritma Nazief dan Adriani melakukan *stemming* dengan menghilangkan *inflection suffixes* (“-lah”, “-kah”, “-ku”, “-mu”, atau “-nya”), *possessive pronouns* (“-ku”, “-mu”, atau “-nya”), *derivation suffixes* (“-i”, “-an” atau “-kan”) dan *derivation prefixes* (“di-”, “ke-”, “se-”, “te-”, “be-”, “me-”, atau “pe-”), kemudian mencocokkan kata dengan kata yang ada di kamus. Proses *stemming* dilakukan untuk menyelaraskan suatu kata yang memiliki imbuhan berbeda agar kata tersebut dapat diartikan atau dimaknai sebagai kata yang sama. Contoh *tweets* yang telah melewati proses *stemming* dapat dilihat pada Tabel 3.7.

Tabel 3.7 *Tweet stemming*

<i>Date</i>	<i>Username</i>	<i>Tweet_Stemming</i>
2021-10-25 07:34:41+00:00	nuchillinaris	'program', 'tanggulang', 'rasa', 'masyara- kat', 'bawah', 'dg', 'program', 'pulih', 'ekonomi', 'sangat', 'bumi', 'amp', 'rasa', 'benar', 'masyarakat', 'yg', 'ekonomi', 'san- gat', 'dampak', 'akibat', 'bijak', 'jokowi', 'aku', 'dunia'
2022-03-22 17:25:29+00:00	ViantAntony	'ruwet', 'ruwet', 'ruwet', 'ini', 'negeri', 'ru- wetnesia', 'hebat', 'virus', 'tau', 'ramadhan', 'akn', 'datang', 'raka', 'akn', 'meperbanyak', 'bhkn', 'mungkin', 'panjang', 'yakan', 'pak'
2022-03-22 15:23:09+00:00	LaNyallaMM1	'harap', 'longgar', 'aktivitas', 'bukan', 'gen- jot', 'ekonomi', 'manfaat', 'sektor', 'didik', 'tingkat', 'sumber', 'daya', 'manusia', 'sedi- kit', 'mundur', 'pandemi'

3.5.4 Feature Weighting

Pada tahap ini, dilakukan pembobotan fitur yang dianggap relevan dalam mewakili suatu kelas, dalam penelitian ini fitur tersebut adalah kata. Kata yang digunakan adalah *unigram* dari hasil *preprocessing*. Dalam penelitian ini metode *feature weighting* yang digunakan adalah metode TF-RF yang dimana akan dibagi ke dalam dua tahap terlebih dahulu pada pemrosesannya:

3.5.4.1 Term Frequency – Relevance Frequency (TF-RF)

a. Term Frequency (TF)

Pada proses berikut ini, semua kata yang ada pada *tweets* akan dijadikan sebagai *feature* pada masing – masing *tweets* untuk proses *training* dan *test*. Melalui proses ini, akan terbentuk *vector* berdasarkan *term* atau kata yang ada pada seluruh teks. Berdasarkan pada jumlah kata yang muncul pada *tweets* tersebut sesuai dengan kata acuannya maka *tweet training* dan *test* dapat diberikan nilai numerik pada *vector*-nya yang sesuai dengan jumlah kemunculan kata acuan dibagi jumlah kata pada kalimat dimana kata acuan tersebut berada. Bobot *term frequency* bersumber dari hasil perhitungan nilai – nilai numerik tersebut.

Setelah tahapan *preprocessing* dilakukan pada *tweet – tweet* yang ada, maka selanjutnya *tweet – tweet* tersebut akan dikonversikan atau diubah ke dalam bentuk angka sehingga *tweet – tweet* tersebut memiliki bobot agar dapat diproses oleh sistem. Penerapan dari kalkulasi *term frequency* dapat dilihat pada 4 contoh dokumen *tweet* yang telah mengalami *preprocessing* sebagai berikut ini:

Negative Tweet 1 : **program tanggulang rasa masyarakat bawah dg program pulih ekonomi sangat bumi amp rasa benar masyarakat yg ekonomi sangat dampak akibat bijak jokowi aku dunia.**

Negative Tweet 2 : **ruwet ruwet ruwet ini negeri ruwetnesia hebat virus tau ramadhan akn datang raka akn meperbanyak bhkn mungkin panjang yakan pak.**

Positive Tweet 3 : **harap longgar aktivitas genjot ekonomi manfaat sektor didik tingkat sumber daya manusia maju pandemi.**

Positive Tweet 4 : **presiden doa masyarakat hidup sehat umur tingkat ekonomi.**

Masing – masing *term* atau kata akan dihitung frekuensi kemunculan nya dalam sebuah dokumen seperti yang terdapat pada Tabel 3.8, dengan persamaan yang digunakan yaitu Persamaan (2.1).

Tabel 3.8 Nilai TF

Term	Term Frequency (TF)			
	D1	D2	D3	D4
program	$\frac{2}{24}$	0	0	0
tanggulang	$\frac{1}{24}$	0	0	0
rasa	$\frac{2}{24}$	0	0	0
masyarakat	$\frac{2}{24}$	0	0	$\frac{1}{8}$
ruwet	0	$\frac{3}{20}$	0	0
virus	0	$\frac{1}{20}$	0	0
pulih	$\frac{1}{24}$	0	0	0
ekonomi	$\frac{2}{24}$	0	$\frac{1}{16}$	$\frac{1}{8}$
sehat	0	0	0	$\frac{1}{8}$
hidup	0	0	0	$\frac{1}{8}$
...

b. *Relevance Frequency* (RF)

Apabila jumlah nilai dari *term frequency* (TF) telah ditemukan pada setiap dokumen, maka prosedur selanjutnya adalah mencari nilai *relevance frequency* (RF) pada setiap kata. Mengacu pada frekuensi kemunculan *term* di kategori yang berkaitan, maka dipertimbangkan relevansi dokumen pada nilai *relevance*

frequency. Nilai RF dari suatu *term* atau kata menjadi tinggi ketika *term frequency* dari kata tersebut memiliki nilai yang tinggi pada suatu dokumen yang mencakup kata tersebut dan pada kelas dokumen lainnya [19]. Berikut disajikan perhitungan nilai RF pada beberapa kata yang dapat dilihat di Tabel 3.9 berikut :

Tabel 3.9 Nilai RF

Term	Relevance Frequency (RF) (Negative Tweet) $\log\left(2 + \frac{b}{\max(1, c)}\right)$	Relevance Frequency (RF) (Positive Tweet) $\log\left(2 + \frac{b}{\max(1, c)}\right)$
program	$\log\left(2 + \frac{1}{\max(1,1)}\right) = 1.09861$	$\log\left(2 + \frac{0}{\max(1,0)}\right) = 0.69314$
tanggulang	$\log\left(2 + \frac{1}{\max(1,1)}\right) = 1.09861$	$\log\left(2 + \frac{0}{\max(1,0)}\right) = 0.69314$
rasa	$\log\left(2 + \frac{1}{\max(1,1)}\right) = 1.09861$	$\log\left(2 + \frac{0}{\max(1,0)}\right) = 0.69314$
masyarakat	$\log\left(2 + \frac{1}{\max(1,1)}\right) = 1.09861$	$\log\left(2 + \frac{1}{\max(1,1)}\right) = 1.09861$
ruwet	$\log\left(2 + \frac{1}{\max(1,1)}\right) = 1.09861$	$\log\left(2 + \frac{0}{\max(1,0)}\right) = 0.69314$
virus	$\log\left(2 + \frac{1}{\max(1,1)}\right) = 1.09861$	$\log\left(2 + \frac{0}{\max(1,0)}\right) = 0.69314$
pulih	$\log\left(2 + \frac{1}{\max(1,1)}\right) = 1.09861$	$\log\left(2 + \frac{0}{\max(1,0)}\right) = 0.69314$
ekonomi	$\log\left(2 + \frac{1}{\max(1,1)}\right) = 1.09861$	$\log\left(2 + \frac{2}{\max(1,0)}\right) = 1.38629$
sehat	$\log\left(2 + \frac{0}{\max(1,0)}\right) = 0.69314$	$\log\left(2 + \frac{1}{\max(1,1)}\right) = 1.09861$
hidup	$\log\left(2 + \frac{0}{\max(1,0)}\right) = 0.69314$	$\log\left(2 + \frac{1}{\max(1,1)}\right) = 1.09861$
...

c. *Term Frequency-Relevance Frequency (TF-RF)*

Pada proses ini, pemberian bobot pada kata dilakukan dengan cara mengalikan nilai *term frequency* dengan nilai *relevance frequency*, hal ini dapat dilihat pada Persamaan (2.2). Pada Tabel 3.10 dan Tabel 3.11 ditampilkan beberapa kata yang telah dihitung nilai akhirnya [19].

Tabel 3.10 Nilai TF-RF Kategori Tweet Positif

Term	TF.RF = TF(d,t) * RF(t)			
	D1	D2	D3	D4
program	0.0577616	0	0	0
tanggulang	0.0288808	0	0	0
rasa	0.0577616	0	0	0
masyarakat	0.0915508	0	0	0.1373262
ruwet	0	0.103971	0	0
virus	0	0.034657	0	0
pulih	0.0288808	0	0	0
ekonomi	0.1155241	0	0.086643	0.1732862
sehat	0	0	0	0.1373262
hidup	0	0	0	0.1373262
...	

Tabel 3.11 Nilai TF-RF Kategori Tweet Negatif

Term	TF.RF = TF(d,t) * RF(t)			
	D1	D2	D3	D4
program	0.0915508	0	0	0
tanggulang	0.0457754	0	0	0
rasa	0.0915508	0	0	0
masyarakat	0.0915508	0	0	0.1373262
ruwet	0	0.164791	0	0
virus	0	0.0549305	0	0
pulih	0.0457754	0	0	0
ekonomi	0.0915508	0	0.0686631	0.1373262

sehat	0	0	0	0.0866425
hidup	0	0	0	0.0866425
...	

3.5.5 Klasifikasi Dengan XGBoost

Setelah *term – term* dari *tweets* pada *dataset* diperoleh bobotnya yang telah diubah menjadi bentuk vektor melalui perhitungan pada metode pembobotan TF-RF, maka tahapan selanjutnya adalah melakukan klasifikasi XGBoost. *Dataset* [X Y] pertama kali akan dicari rata – rata nilai target (Y), ini dilakukan untuk memperoleh nilai prediksi awal (h_0) dan nilai residual error (\hat{Y}) awal, dimana proses ini dinyatakan pada Persamaan (2.3) dan (2.4). Untuk memperoleh model pertama (M1) yang merupakan sebuah *decision tree* yang dilatih dengan variabel independen dan residual error [$X\hat{Y}$] sebagai data untuk mendapatkan prediksi dari model M1, maka dilakukanlah *training* data pada model pertama tersebut terlebih dahulu. Ketika proses *training*, model dapat dimaksimalkan dengan menggunakan beberapa *hyperparameter* yang tersedia pada model yang digunakan.

Pada algoritma *XGBoost* tingkatan seberapa berpengaruh *hyperparameter* pada kinerja model dapat bervariasi, faktor yang memengaruhi kinerja model selain dari penggunaan *hyperparameter* yaitu *dataset*, dan masalah yang ingin diselesaikan. Pada penelitian ini penulis mencoba menerapkan beberapa *hyperparameter* yang biasanya paling krusial dalam menentukan kinerja model yang dimana diperlihatkan sebagai berikut pada Tabel 3.12 di bawah ini.

Tabel 3.12 *hyperparameter XGBoost*

Hyperparameter	Score
<i>gamma</i>	1
<i>learning_rate</i>	1
<i>n_estimators</i>	100
<i>max_depth</i>	1
<i>subsample</i>	1

Penulis memperoleh nilai *score* pada Tabel 3.11 yang mengacu pada penelitian sebelumnya yang juga menggunakan algoritma *XGBoost* [33].

'*n_estimators*' merupakan *hyperparameter* yang menentukan jumlah *decision tree* yang dibangun pada model, '*learning_rate*' merupakan *hyperparameter* yang mengontrol tingkatan cepat lambatnya model belajar, '*max_depth*' mengontrol seberapa maximum kedalaman masing – masing *decision tree* pada model, '*subsample*' menentukan persentase jumlah sampel yang diambil secara *random* dari *training data* untuk membangun setiap *decision tree* pada model, '*gamma*' menentukan ambang batas pada saat model akan berhenti membuat pemisahan *node*. *Score* pada *hyperparameter* dapat berubah dikarenakan penulis melakukan eksperimen untuk menemukan *set* variasi *hyperparameter* yang optimal dengan tujuan untuk meningkatkan kinerja model. Penulis nantinya mencoba menggunakan masing – masing beberapa nilai yang bervariasi pada 5 *hyperparameter* yang telah disajikan pada Tabel 3.11, ini digunakan dengan tujuan untuk memperoleh nilai akurasi yang lebih akurat pada kinerja model.

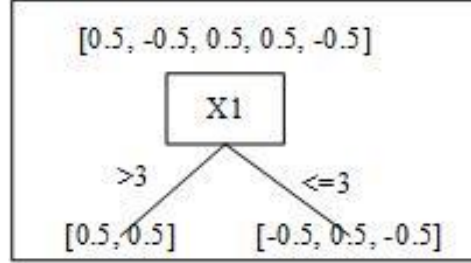
Berikut disajikan contoh perhitungan manual dari algoritma *XGBoost* dengan menggunakan nilai dari hasil perhitungan TF-RF. Fitur yang digunakan hanya dua dengan tujuan agar proses perhitungan dapat mudah dipahami dan juga dikarenakan oleh ini hanya sebagai contoh perhitungan manualnya saja.

Tabel 3.13 Contoh nilai fitur dengan 2 Kelas

	X1	X2	Kelas	Residual
<i>D1</i>	5	1	1	0.5
<i>D2</i>	3	3	0	-0.5
<i>D3</i>	1	0	1	0.5
<i>D4</i>	4	2	1	0.5
<i>D5</i>	2	3	0	-0.5

Untuk membangun model awal, digunakan persamaan yang telah dijabarkan sebelumnya, persamaan 2.3 dan persamaan 2.4 digunakan untuk menentukan nilai prediksi awal dan kesalahan residual dari model asli. Pada persamaan 2.3, $h_0(x)$ yang bernilai 0.5 merupakan representasi nilai prediksi awal dari model pertama dicari nilainya, nilai ini diperoleh dengan merata-ratakan nilai dari target atau kelas ($mean(Y)$). Kemudian pada persamaan 2.4, dijabarkan bahwa nilai yang dicari selanjutnya yaitu \hat{Y} yang merepresentasikan nilai residual *error* model awal,

nilai ini dicari dengan cara mengurangi nilai kelas dengan nilai prediksi awal dari model pertama ($h_0(x)$).



Gambar 3.3 Contoh *decision tree*

Setelah seluruh *pseudo-residual* diperoleh dari setiap sampel yang ada, maka proses dilanjutkan dengan membangun *decision tree* berdasarkan dari *pseudo-residual* yang telah diperoleh, berikut pada Gambar 3.3 disajikan gambar *decision tree awal*, kemudian nilai yang dicari selanjutnya yaitu *similarity weight* pada setiap *node* yang ada dengan persamaan (3.1). *Similarity weight* ini nantinya digunakan untuk memperoleh *gain* pada *tree* guna menentukan *attributes* yang nantinya dijadikan sebagai *attributes* pemisah (*split*) pada *tree*.

$$similarity\ weight = \frac{(\sum Residuals)^2}{\sum(Prob(1 - Prob) + \lambda)} \quad (3.1)$$

$$gain = sim(left\ node) + sim(right\ node) - sim(root\ node) \quad (3.2)$$

Berikut ini disajikan contoh kalkulasi dalam menghitung *similarity weight* untuk *left node*, *right node*, dan *root node* yang ditampilkan pada Gambar 3.3, kalkulasi ini akan disajikan pada persamaan (3.3).

$$similarity\ weight_{Lnode} = \frac{(0.5 + 0.5)^2}{(0.5(1 - 0.5) + 0.5(1 - 0.5))} = 2 \quad (3.3)$$

$$similarity\ weight_{Rnode}$$

$$= \frac{(-0.5 + 0.5 + (-0.5))^2}{(0.5(1 - 0.5) + 0.5(1 - 0.5) + 0.5(1 - 0.5))} = 0.33$$

$$\begin{aligned}
& \text{similarity weight}_{\text{Rootnode}} \\
&= \frac{(0.5 + (-0.5) + 0.5 + 0.5 + (-0.5))^2}{(0.5(1 - 0.5) + 0.5(1 - 0.5) + 0.5(1 - 0.5) + 0.5(1 - 0.5) + 0.5(1 - 0.5))} \\
&= 0.2
\end{aligned}$$

Kemudian setelah nilai tersebut diketahui maka selanjutnya dicari nilai dari "gain" dengan kalkulasi sebagai berikut.

$$\text{gain} = 2 + 0.33 + 0.2 = 2.13 \quad (3.4)$$

Gain tersebut nantinya akan dibandingkan dengan gain yang diperoleh dari *attributes* lainnya selain dari *attributes output* atau target, pada contoh ini yaitu X2, *attributes* dengan nilai gain tertinggi akan digunakan sebagai pemisah (*split*) *node* paling dasar pada *tree*. Lambda pada *denominator* merupakan parameter regularisasi yang digunakan untuk membatasi kemampuan model untuk belajar, ini dikarenakan algoritma *boosting* cenderung menciptakan *overfitting*, dengan ditambakkannya lambda maka *similarity weight* dan *gain* akan menjadi lebih rendah sehingga ini akan mengarahkan ke langkah selanjutnya yaitu *tree pruning* (pemangkasan pohon). Pemangkasan pohon secara sederhana dimaksudkan untuk menentukan apakah *splitting* sebaiknya dilakukan atau tidak.

Sebagai contoh, asumsikan telah dibuat satu buah *decision tree* dari kalkulasi sebelumnya dimana *tree* beserta *leaves* ini dibangun dengan menggunakan nilai *residuals* dari probabilitas yang kita tentukan dengan menggunakan persamaan (2.3) dan persamaan (2.4). Kemudian asumsikan diperoleh sebuah data baru dengan *similarity weight* senilai 1 dimana nilai ini didapat melalui alur pada *node decision tree* yang pertama dibuat, lalu kalkulasi dilanjutkan lagi dengan cara mencari nilai dari *base model* untuk memperoleh nilai probabilitas selanjutnya, *base model* adalah nilai probabilitas pertama, dikarenakan iterasi yang dilakukan baru sekali saja, lalu untuk melanjutkan proses kalkulasi untuk memperoleh *base model*, digunakan persamaan (3.5) kemudian dilanjutkan lagi dengan persamaan (3.6) untuk mencari nilai *sigmoid activation function*.

$$\log(odds) = \log\left(\frac{P}{1-P}\right) = \log\left(\frac{0.5}{1-0.5}\right) = 0 \quad (3.5)$$

$$\sigma(0 + 0.1(1)) = 0.1 \quad (3.6)$$

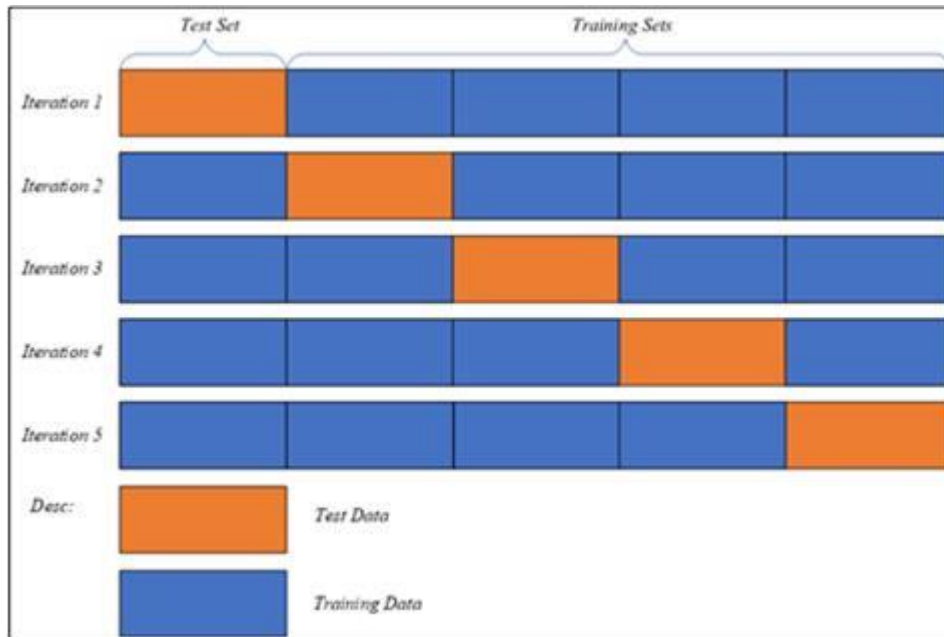
$$probability = \frac{1}{1 + e^{0.1}} \quad (3.7)$$

Pada persamaan (3.6) ditunjukkan bahwa nilai yang diperoleh sama dengan 0.1, lalu pada persamaan (3.7) yang berfungsi untuk mencari nilai probabilitas selanjutnya, asumsikan jika nilai probabilitas yang diperoleh sama dengan 0.6, maka nilai tersebut digunakan untuk mencari nilai residual yang baru dengan cara mengurangi nilai kelas dengan nilai tersebut, yang nantinya nilai residual itu digunakan untuk membuat *decision tree* selanjutnya. Iterasi akan terus dilakukan oleh model dengan membuat *decision tree* baru yang diarahkan untuk memperbaiki prediksi model sebelumnya dengan meminimalkan residual yang masih ada.

Secara matematis, algoritma ini memodelkan hubungan antara fitur – fitur *input* dan label *output*, dengan menggunakan sekumpulan *decision tree*, sambil mengoptimalkan *loss function* melalui pendekatan *gradient* turunan. Melalui hal tersebutlah yang membuat *XGBoost* dapat menghasilkan model yang akurat untuk tugas klasifikasi.

3.6 Pengujian

Pengujian yang dilakukan dalam penelitian terkait analisis sentimen masyarakat pada media sosial *twitter* menggunakan metode *XGBoost* adalah dengan menggunakan teknik *k-fold cross validation*, dan *Dataset* yang terkait pada penelitian ini diperoleh melalui *crawling tweets* pada *twitter*. Teknik *k-fold cross validation* merupakan teknik yang digunakan untuk mengevaluasi hasil klasifikasi. Dalam skenario pengujian pada penelitian ini, *dataset* dibagi menjadi beberapa bagian yang disebut *fold*. Pada setiap iterasi yang dilakukan, salah satu *fold* digunakan sebagai *testing data*, dan sisa *fold* digunakan sebagai *training data*. Proses ini dilakukan sebanyak nilai K yang ditetapkan hingga seluruh *fold* digunakan sebagai *testing data* [19].



Gambar 3.4 Ilustrasi *cross validation 5 fold*

Berdasarkan pada Gambar 3.4 yang merupakan ilustrasi dari *cross validation*, maka prosesnya dapat dijabarkan sebagai berikut ini [19]:

1. Jumlah *instance* dibagi sebanyak K bagian atau disebut *fold*.
2. Pada iterasi ke-1 adalah saat bagian ke-1 dijadikan sebagai data uji dan empat bagian sisanya dijadikan sebagai data latih. Kemudian dilakukan penghitungan akurasi atau kesamaan atau kedekatan pada hasil pengukuran dengan menggunakan angka atau data yang sebenarnya berdasarkan porsi dari data tersebut. Persamaan yang digunakan pada perhitungan akurasi adalah sebagai berikut:

$$Akurasi = \frac{\sum data\ uji\ benar\ klasifikasi}{\sum jumlah\ data\ uji} \times 100 \quad (3.2)$$

3. Pada iterasi ke-2 yaitu saat bagian ke-2 dijadikan sebagai data uji dan bagian lain sisanya dijadikan sebagai data latih. Kemudian, dilakukan penghitungan akurasi berdasarkan porsi data tersebut.
4. Lalu seterusnya penghitungan dilakukan hingga mencapai iterasi atau *fold* ke-K. Kemudian rata – rata akurasi yang diperoleh dari K buah akurasi akan dijadikan sebagai akurasi final.

Dua kelas atau kategori yang berbeda akan digunakan pada penelitian ini untuk membedakan sentimen dari *tweets* yaitu, kategori *tweets* positif dan *tweets* negatif. Terdapat beberapa parameter yang diuji dalam penelitian ini antara lain :

1. Pengaruh himpunan variasi *hyperparameter* pada model *eXtreme Gradient Boosting* terhadap akurasi kinerja model. *XGBoost* memiliki banyak *hyperparameter* yang sangat berperan penting dalam memengaruhi kinerja model.
2. Pengaruh augmentasi data pada akurasi

Augmentasi pada data digunakan untuk mengatasi apabila jumlah *dataset* yang diperoleh masih cukup kurang dari jumlah *dataset* yang direncanakan untuk digunakan pada proses klasifikasi dengan cara meningkatkan jumlah data tersebut melalui proses augmentasi itu sendiri [34], selain itu apabila data yang ada di masing – masing kategori sentimen ternyata memiliki jumlah yang tidak seimbang, maka proses augmentasi data dapat membantu untuk menyeimbangkan jumlah *dataset* tersebut dengan melakukan augmentasi pada data yang memiliki jumlah yang lebih sedikit.

3. Pengaruh *stemming* pada akurasi

Menurut penelitian yang disebutkan dalam referensi [35], penggunaan *stemming* sebenarnya dapat mengurangi tingkat akurasi karena dalam prosesnya mengubah kata-kata yang sebenarnya merupakan ciri penting dari dokumen yang relevan.

4. Pengaruh nilai K pada metode *K-fold cross validation* yang diterapkan terhadap nilai akurasi model yang diperoleh.

Pada setiap percobaan evaluasi dilakukan dengan menghitung *accuracy* *recall* dan *precision* dari model.

Untuk menghitung nilai-nilai tersebut diperlukan *confussion matrix* untuk menyajikan hasil klasifikasi dalam bentuk tabel. Tabel 3.13 *confussion matrix* dalam penelitian ini dapat dilihat pada.

Tabel 3.14 *Confusion matrix* yang digunakan pada penelitian

Kebenaran	Hasil Klasifikasi		Total
	Positif	Negatif	
Positif	<i>True Positive</i>	<i>False Negative</i>	Total Kelas Positif
Negatif	<i>False Positive</i>	<i>True Negative</i>	Total Kelas Negatif
	Prediksi Kelas Positif	Prediksi Kelas Negatif	

Pada Tabel 3.13 *confusion matrix*, Ketika hasil klasifikasi dari kelas memberikan hasil positif sementara pada kebenarannya juga berkategori positif, maka hal tersebut dapat digolongkan sebagai *true positive*, sementara apabila hasil klasifikasi menunjukkan kelas negatif dan kebenarannya juga menunjukkan negatif maka hal tersebut dinamakan dengan *true negative*. Namun apabila hasil klasifikasi dan kebenarannya berlawanan, seperti didapati hasil klasifikasinya berkategori positif sementara kebenarannya negatif maka hal tersebut dinamakan *false positive*, sementara untuk sebaliknya disebut dengan *false negative*.

Recall dan *precision* untuk tiap sentimen dihitung dengan menggunakan Persamaan (2.8) dan Persamaan (2.7). Sedangkan untuk *accuracy* dihitung menggunakan Persamaan (2.6). Nilai *recall* dan *precision* untuk tiap percobaan didapatkan dengan mencari nilai rata-rata dari *recall* dan *precision* per sentimen. Performa model secara keseluruhan didapatkan dengan menghitung nilai *accuracy* serta nilai rata-rata *recall* dan *precision* dari seluruh percobaan.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Pengumpulan Data

Pada penelitian ini, pengumpulan *dataset* dilakukan dengan cara melakukan *crawling* pada laman *twitter* dikarenakan *dataset* yang dikumpulkan merupakan *dataset* berjenis *tweets*, *tweets – tweets* yang diambil merupakan *tweets* yang memuat tagar “#ppkm” dengan periode pengambilan *tweets* dari tanggal 1 april 2020 hingga 1 april 2022. *Crawling* dilakukan dengan memanfaatkan *library* “*snsrape*”, dari hasil *crawling* diperoleh sekitar 20.000 *tweets* berbahasa Indonesia, yang berisikan berbagai macam jenis sentimen masyarakat Indonesia terhadap kebijakan PPKM yang diterapkan oleh pemerintah Indonesia dimulai dari *tweets* dengan sentimen positif, negatif, dan netral.

Pada penelitian ini, kategori sentimen yang digunakan hanya sentimen dengan kategori positif atau negatif, setelah dilakukan *crawling*, *tweets – tweets* tersebut disimpan ke dalam file dengan ekstensi “*csv*”, dari 20.000 *tweets* yang diperoleh sebelumnya, diambil 10.000 *tweets* saja untuk digunakan pada proses klasifikasi jenis *tweets* pada penelitian ini. Pertimbangan dalam menggunakan 10.000 *tweets* ini adalah karena untuk memangkas waktu yang dibutuhkan pada saat proses pelabelan dilakukan dan juga dari segi jumlah tersebut sudah sangat melampaui jumlah *dataset* yang digunakan pada penelitian sebelumnya [12]. Setelah dilakukan proses pelabelan, dari 10.000 *tweets* tersebut, ternyata diperoleh 1958 *tweets* berkategori positif, dan 3980 *tweets* dengan kategori negatif, dimana sisa dari *tweets – tweets* tersebut merupakan *tweets* dengan kategori netral. *Tweets* netral tersebut diabaikan dan hanya diambil *tweets* dengan kategori positif ataupun negatif.

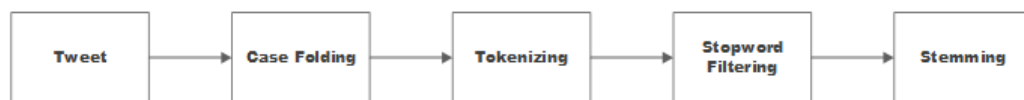
4.2 Augmentasi Data

Berdasarkan dari hasil pelabelan yang diperoleh dan telah dipaparkan pada subbab sebelumnya, apabila mengacu pada referensi laman resmi google terkait dengan *imbalanced dataset* [36], maka *dataset* ini dapat dikatakan sebagai *imbalanced*, ini dapat dilihat pada perbandingan jumlah *tweets* dengan kategori sentimen positif yang berjumlah 1958 dengan *tweets* pada kategori negatif dengan jumlah 3980 *tweets*, dimana jumlah *tweets* pada kategori positif hanya senilai 32.97% dari total data yang ada dan dikategorikan sebagai tingkat ketidakseimbangan dengan kategori “*mild*” apabila mengacu pada referensi yang ada [36]. Dilatar belakangi oleh hal tersebut maka penulis mencoba

melakukan augmentasi pada *dataset* yang ada, augmentasi dilakukan pada *tweets* dengan kategori positif dikarenakan jumlahnya yang hanya setengah dari *tweets* pada kategori negatif, dimana hal tersebutlah yang menyebabkan terjadinya ketidakseimbangan data dan tentunya hal ini dapat memengaruhi hasil klasifikasi dari model nantinya.

4.3 Text Preprocessing

Tahapan selanjutnya pada penelitian ini yaitu dilakukan *preprocessing* pada seluruh *dataset*, dimana pada tahapan ini, *preprocessing* terdiri menjadi beberapa bagian yang tersaji pada Gambar 4.1 berikut ini.



Gambar 4.1 Bagan proses *preprocessing*

Dimulai dari *tweet* yang telah dilabeli, kemudian dilanjutkan dengan melewati beberapa tahapan secara berurut meliputi *case folding*, *tokenizing*, *stopword removal*, dan *stemming*.

4.3.1 Case Folding

Tahapan awal yang dilakukan dalam *preprocessing* adalah melakukan *case folding* pada seluruh teks pada *tweets* menjadi huruf kecil dengan tujuan untuk menghindari *tweets – tweets* tersebut memiliki makna ganda dikarenakan kata yang sama apabila ada perbedaan terhadap penggunaan huruf besar maupun huruf kecil, maka kata tersebut akan diartikan sebagai kata yang berbeda, maka dari itu *case folding* dilakukan untuk mencegah ambiguitas pada *term* atau kata seperti pada Gambar 4.2 di bawah ini.

	tweet	sentiment
0	Karena stimulasi untuk #anakberkebutuhankhusus tidak optimal jika dilak...	0
1	Ye ampun mbh anda seorang Wapres loh.&Gelar anda seorang kiyai @Kiy...	0
2	Mulaik kumat lagi. 😊 #ppkm https://t.co/MNDSPHJU69	0
3	Welcome PPKM bau bau naik level nih, gimana nih mang\n#ppkm	0
4	Sesungguhnya Allah tidak menciptakan kita, hambaNya ini, sebagai makhluk...	0

↓

	tweet	sentiment
0	karena stimulasi untuk #anakberkebutuhankhusus tidak optimal jika dilak...	0
1	ye ampun mbh anda seorang wapres loh.&gelar anda seorang kiyai @kiy...	0
2	mulaik kumat lagi. 😊 #ppkm https://t.co/mndsphju69	0
3	welcome ppkm bau bau naik level nih, gimana nih mang\n#ppkm	0
4	sesungguhnya allah tidak menciptakan kita, hambanya ini, sebagai makhluk...	0

Gambar 4.2 Tahap *Case Folding*

4.3.2 Tokenizing

Setelah *case folding* dilakukan, maka tahapan selanjutnya yaitu melakukan *tokenizing*. *Tokenizing* dilakukan dengan tujuan untuk memisahkan kata dari kalimat, dimana kata atau *term* yang ada pada *tweets* dipisahkan sehingga menjadi satuan kata atau dalam artian lain dapat disebut dengan *token*. Pada Gambar 4.3 berikut disajikan bagaimana perubahan pada *tweet* setelah dilakukannya *tokenizing*.

	tweet	sentiment
0	karena stimulasi untuk #anakberkebutuhankhusus tidak optimal jika dilak...	0
1	ye ampun mbh anda seorang wapres loh.&gelar anda seorang kiyai @kiy...	0
2	mulaik kumat lagi. 😞 #ppkm https://t.co/mndsphju69	0
3	welcome ppkm bau bau naik level nih, gimana nih mang\n#ppkm	0
4	sesungguhnya allah tidak menciptakan kita, hambanya ini, sebagai makhlu...	0

	tweet	sentiment
0	[karena, stimulasi, untuk, tidak, optimal, jika, dilakukan, secara, onl...	0
1	[ye, ampun, mbh, anda, seorang, wapres, lohampgelar, anda, seorang, kiy...	0
2	[mulaik, kumat, lagi]	0
3	[welcome, ppkm, bau, bau, naik, level, nih, gimana, nih, mang]	0
4	[sesungguhnya, allah, tidak, menciptakan, kita, hambanya, ini, sebagai,...	0

Gambar 4.3 Tahap *Tokenizing*

4.3.3 Stopwords Filtering

Kemudian setiap *term* yang telah diperoleh dari proses *tokenizing* akan melewati tahapan *stopword filtering*, pada proses ini kata atau *term* yang tidak relevan dan dianggap tidak memiliki makna akan dibuang, dan *term* yang dianggap penting akan tetap disimpan. Pada Gambar 4.4 disajikan perubahan dari *tweet* yang telah melewati proses *stopwords filtering*


	tweet	sentiment
0	[karena, stimulasi, untuk, tidak, optimal, jika, dilakukan, secara, onl...	0
1	[ye, ampun, mbh, anda, seorang, wapres, lohampgelar, anda, seorang, kiy...	0
2	[mulaik, kumat, lagi]	0
3	[welcome, ppkm, bau, bau, naik, level, nih, gimana, nih, mang]	0
4	[sesungguhnya, allah, tidak, menciptakan, kita, hambanya, ini, sebagai,...	0

	tweet	sentiment
0	[stimulasi, optimal, dilakukan, online, memenuhi, kebutuhan, tersebut, ...	0
1	[ye, ampun, mbh, seorang, wapres, lohampgelar, seorang, kiyai, marufami...	0
2	[mulaik, kumat]	0
3	[welcome, ppkm, bau, bau, naik, level, nih, gimana, nih, mang]	0
4	[sesungguhnya, allah, menciptakan, hambanya, makhluk, lemah, mudah, ber...	0

Gambar 4.4 Tahap *Stopwords Filtering*

4.3.4 Stemming

Proses terakhir merupakan *stemming*, *stemming* yaitu tahapan yang bertujuan untuk mengubah kata yang memiliki imbuhan menjadi kata dasarnya saja, pada proses *stemming* ini, algoritma yang digunakan merupakan algoritma nazief dan adriani [32].



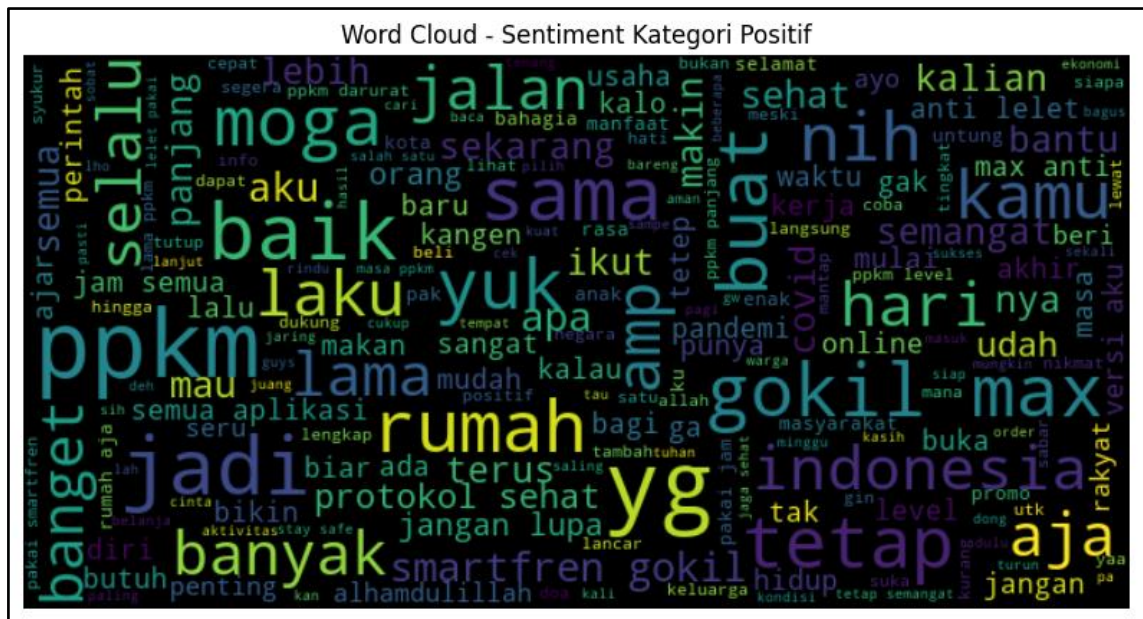
	tweet	sentiment
0	[stimulasi, optimal, dilakukan, online, memenuhi, kebutuhan, tersebut, ...	0
1	[ye, ampun, mbh, seorang, wapres, lohampgelar, seorang, kiyai, marufami...	0
2	[mulaik, kumat]	0
3	[welcome, ppkm, bau, bau, naik, level, nih, gimana, nih, mang]	0
4	[sesungguhnya, allah, menciptakan, hambanya, makhluk, lemah, mudah, ber...	0

	tweet	sentiment
0	[stimulasi, optimal, laku, online, penuh, butuh, sebut, tetap, operasi,...	0
1	[ye, ampun, mbh, orang, wapres, lohampgelar, orang, kiyai, marufamin, k...	0
2	[mulaik, kumat]	0
3	[welcome, ppkm, bau, bau, naik, level, nih, gimana, nih, mang]	0
4	[sungguh, allah, cipta, hamba, makhluk, lemah, mudah, putus, asa, manus...	0

Gambar 4.5 Tahap *Stemming*

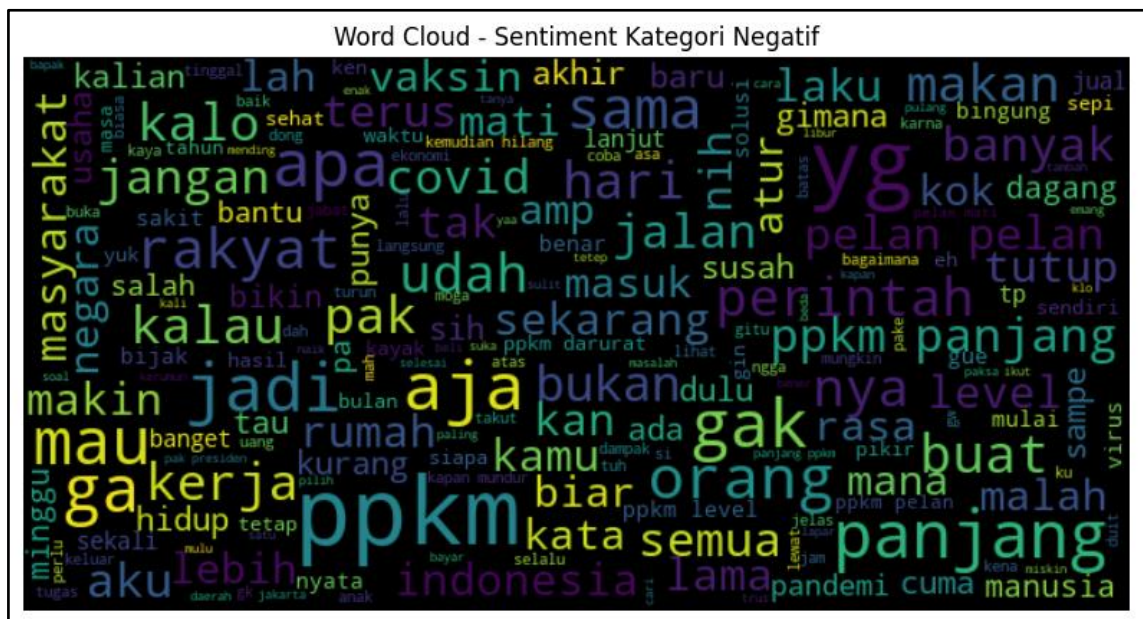
4.4 Wordcloud

Berikut ini penulis juga sajikan tampilan dari *wordcloud* pada kedua jenis *dataset*, baik yang belum diaugmentasi maupun yang telah dilakukan augmentasi, *wordcloud* yang ditampilkan merupakan *wordcloud* dari masing – masing kategori sentimen dengan *tweets* yang telah melewati tahapan *preprocessing* (termasuk *stemming*), ini dikarenakan pada *tweets* yang telah melewati proses *stemming* telah diperoleh *root word* atau kata dasarnya, sehingga *tweets* – *tweets* yang memiliki *sufiks* atau *prefiks* yang berbeda namun memiliki *root word* atau kata dasar yang sama dapat digolongkan sebagai 1 kata apabila memiliki kata dasar yang sama, secara langsung ini dapat meningkatkan jumlah frekuensi dari *term* atau kata yang dimaksud. Tujuan dari visualisasi *wordcloud* ini adalah untuk menampilkan informasi berupa kata – kata yang paling sering muncul dalam *tweets* yang ada, frekuensi kemunculan kata ditunjukkan dengan ukuran kata pada *wordcloud*, semakin besar frekuensinya maka semakin besar ukuran kata tersebut.



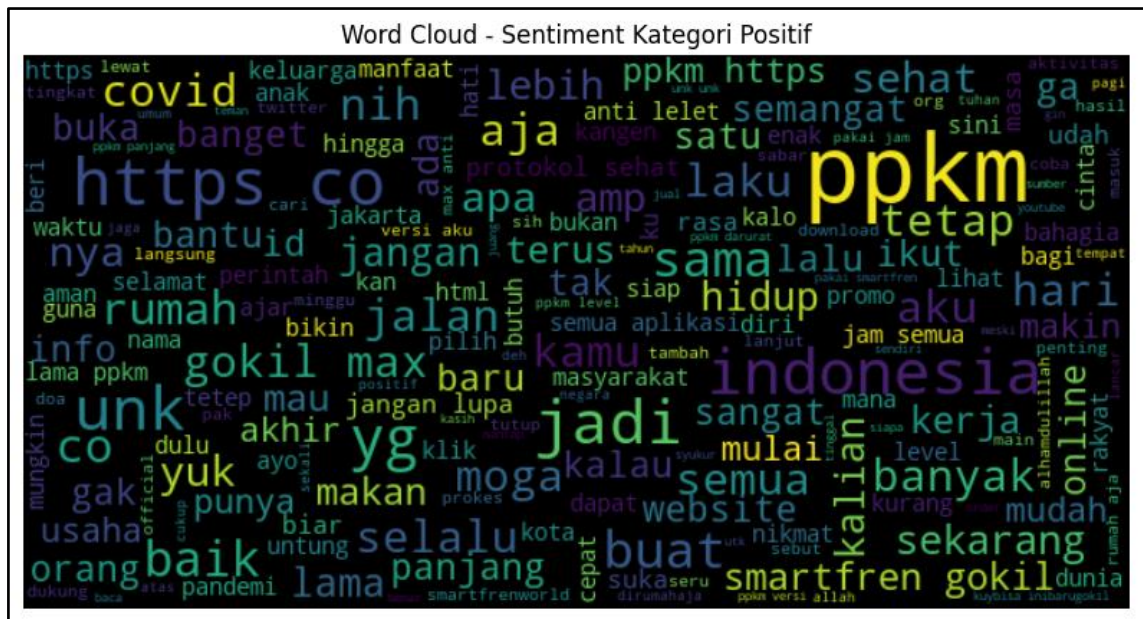
Gambar 4.6 Wordcloud sentimen kategori positif pada dataset tanpa augmentasi

Pada Gambar 4.10 ditampilkan visualiasi *wordcloud* dari *tweets – tweets* dengan kategori positif yang berada pada dataset yang belum dilakukan augmentasi.



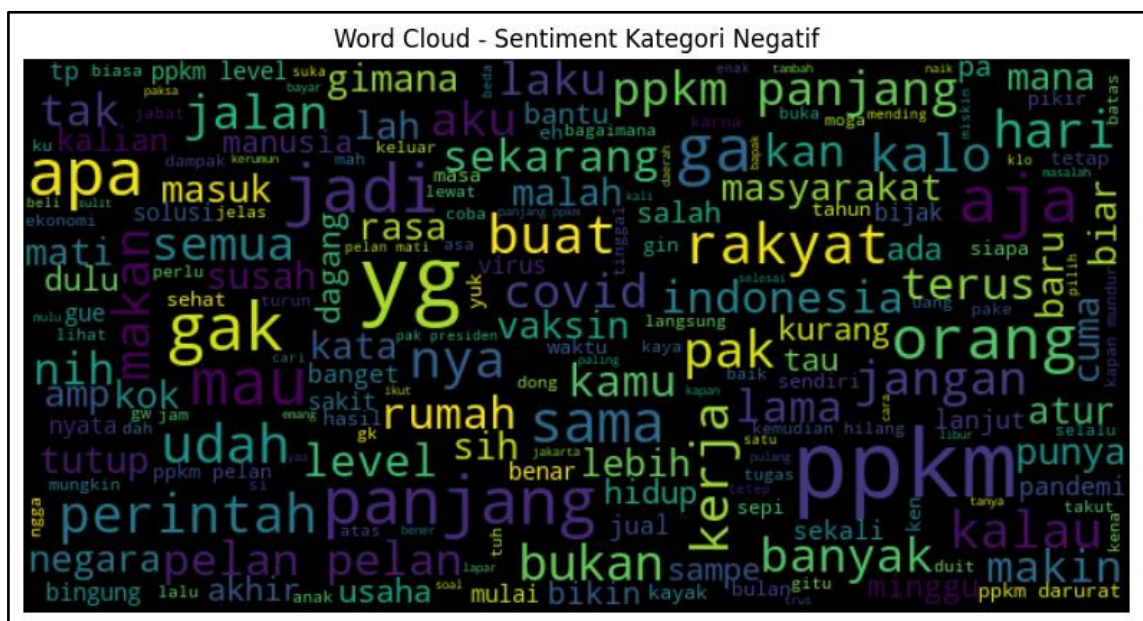
Gambar 4.7 Wordcloud sentimen kategori negatif pada dataset tanpa augmentasi

Pada Gambar 4.11 ditampilkan visualiasi *wordcloud* dari *tweets – tweets* dengan kategori negatif yang berada pada dataset yang belum dilakukan augmentasi.



Gambar 4.8 Wordcloud sentimen kategori positif pada dataset augmentasi

Pada Gambar 4.12 ditampilkan visualiasi *wordcloud* dari *tweets – tweets* dengan kategori positif yang berada pada dataset yang telah dilakukan augmentasi.



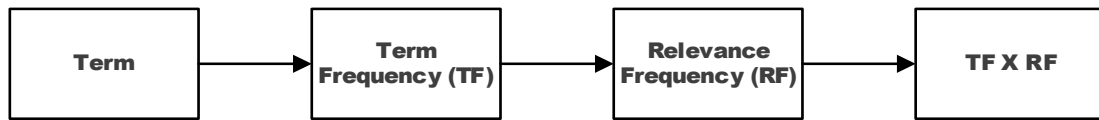
Gambar 4.9 Wordcloud sentimen kategori negatif pada dataset augmentasi

Pada Gambar 4.13 ditampilkan visualiasi *wordcloud* dari *tweets – tweets* dengan kategori negatif yang berada pada dataset yang telah dilakukan augmentasi.

4.5 Term Weighting

Tahapan selanjutnya yang dilakukan adalah melakukan pembobotan pada *term* yang ada pada tiap – tiap *tweet* yang telah melewati tahapan *preprocessing* sebelumnya. Tahapan pembobotan *term* ini bertujuan untuk menentukan ciri atau fitur pada masing –

masing kategori sentimen. Setelah ciri atau fitur *term – term* tersebut diperoleh, maka dapat tahapan selanjutnya yang dapat dilakukan adalah melakukan *training dataset* pada model. Pada Gambar 4.6 disajikan tahapan – tahapan yang dilakukan pada pembobotan *term* berikut ini.



Gambar 4.10 Tahapan pembobotan *term*

Pada tahapan ini, *term – term* yang telah diperoleh dari hasil *preprocessing* sebelumnya, diolah kembali untuk diubah ke dalam bentuk *vector numeric*. Kemudian, setiap *term* pada vektor tersebut dikumpulkan dan disimpan bersama dengan frekuensi kemunculan term tersebut di setiap dokumen dalam setiap kelas atau kategori. Nilai ini kemudian mewakili frekuensi term yang disebut sebagai *term frequency* (TF). Selanjutnya, dilakukan pencarian nilai keunikan untuk setiap term dalam setiap kategori atau kelas dengan memeriksa nilai term tersebut di dokumen lain. Selain itu, dicari juga term-term yang memiliki frekuensi tinggi dalam suatu dokumen dan dokumen di kelas lain yang menunjukkan relevansi frekuensi (*relevance frequency*). Kemudian, dilakukan perkalian antara setiap nilai *relevance frequency* pada setiap kata dengan nilai *term frequency* pada masing-masing kata. Hasil perkalian tersebut akan menghasilkan bobot dari *Term Frequency Relevance Frequency* (TF-RF). Bobot TF-RF ini kemudian digunakan sebagai fitur atau bobot pada vektor token atau *term* yang telah terbentuk, selanjutnya, vektor tersebut akan disimpan sebagai model *training*. Model *training* yang telah disimpan dalam proses sebelumnya akan digunakan untuk melakukan klasifikasi terhadap data *testing* yang digunakan.

4.6 Pengujian

Dalam penelitian ini, dilakukan beberapa skenario pengujian yang meliputi beberapa pengujian berikut :

1. Pengaruh himpunan variasi dari *hyperparameter* pada akurasi model *XGBoost*.

Hyperparameter pada *XGBoost* merupakan salah satu faktor yang sangat memengaruhi kinerja model, seperti pada hal yang telah dijabarkan oleh penulis pada bab sebelumnya. Pada skenario ini penulis mencoba melakukan variasi pada *score* atau nilai *hyperparameter* yang digunakan untuk mencari model dengan nilai *score* yang optimal. *Score hyperparameter* yang penulis variasikan yaitu "*n_estimators*",

"*learning_rate*", "*max_depth*", "*subsample*", dan "*gamma*", karena berdasarkan dari penelitian sebelumnya, *hyperparameter* di atas merupakan *hyperparameter* yang paling krusial dalam menentukan akurasi model [33]. Beberapa *score* yang penulis gunakan merupakan *score* yang digunakan pada penelitian – penelitian sebelumnya [33][37][38], dan juga ada beberapa *score* yang penulis tentukan sendiri nilainya. Skenario *score* yang digunakan pada tiap – tiap *hyperparameter* meliputi sebagai berikut.

- a. *n_estimators* : [100, 1000]
- b. *learning_rate* : [0.01, 0.1]
- c. *max_depth* : [3, 6]
- d. *subsample* : [1]
- e. *gamma* : [0]

2. Pengaruh augmentasi data pada akurasi model *XGBoost*

Setelah dilakukan pelabelan dan diketahui bahwa jumlah *tweets* dengan kategori positif yaitu sebanyak 1958, sedangkan *tweets* dengan kategori negatif yaitu sebanyak 3980. Penulis mencoba untuk melakukan augmentasi pada *tweets* yang berada pada kategori positif, dikarenakan jumlah *tweets* yang berada pada kategori tersebut dapat dikatakan hanya setengah dari *tweets* yang berada pada kategori negatif, dimana apabila mengacu pada referensi berikut[36], maka *dataset* tersebut dapat dikatakan sebagai *imbalanced dataset*. Augmentasi data penulis lakukan dengan asumsi bahwa augmentasi ini dapat menambah hasil akurasi yang diperoleh pada saat *training* model dilakukan.

3. Pengaruh *stemming* pada akurasi model *XGBoost*

Menurut beberapa penelitian yang disebutkan dalam referensi [35], penggunaan *stemming* sebenarnya dapat mengurangi tingkat akurasi karena dalam prosesnya mengubah kata-kata yang sebenarnya merupakan ciri penting dari dokumen yang relevan. Pada skenario ini penulis mencoba membandingkan akurasi model pada *dataset* yang telah melewati proses *stemming* dan *dataset* yang tanpa melalui proses *stemming*.

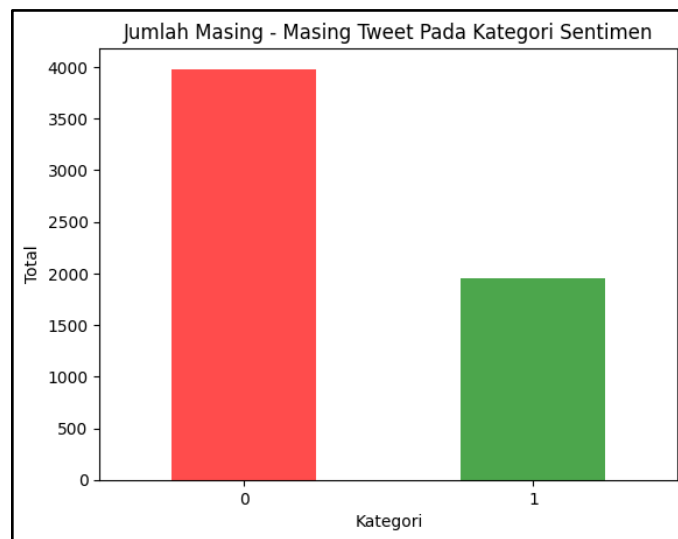
4. Pengaruh nilai K pada metode *K-fold cross validation* pada akurasi model *XGBoost*.

Sama seperti pada apa yang telah dijelaskan sebelumnya, pada penelitian ini nilai K yang digunakan pada metode *k-fold cross validation* divariasikan menjadi dua jenis, yaitu dengan menggunakan nilai K = 5, dan K = 10. Setelah itu dilakukan

perbandingan antara nilai K mana yang memiliki hasil performa lebih baik terhadap model diantara kedua nilai tersebut.

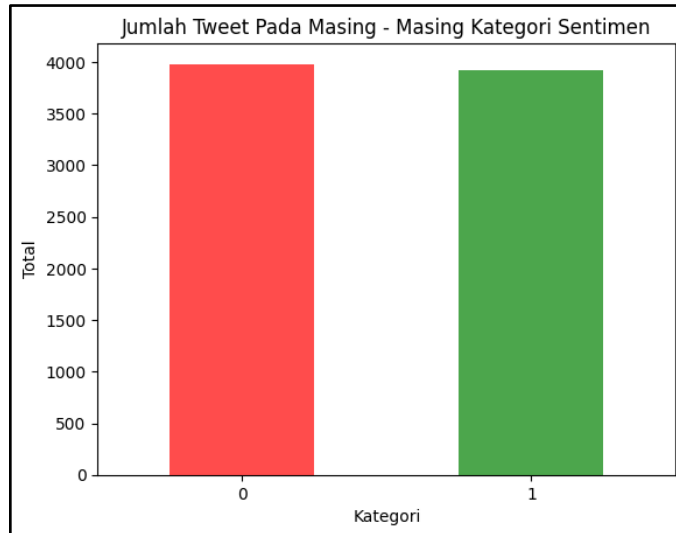
4.7 Hasil Pengujian

Berlandaskan pada beberapa skenario yang telah dijabarkan pada bagian sebelumnya, proses pengujian diawali dengan menggolongkan *dataset* menjadi dua bagian yaitu, *dataset* yang belum mengalami augmentasi dan *dataset* yang telah dilakukan augmentasi pada data tersebut.



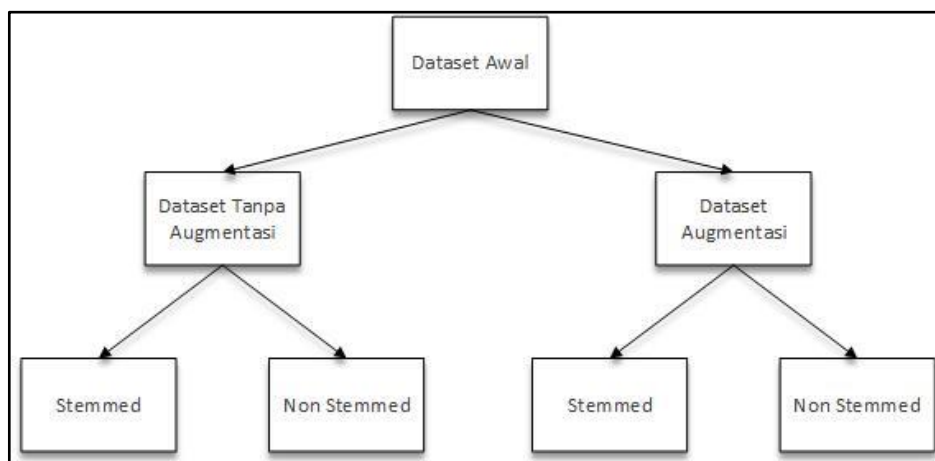
Gambar 4.11 Perbandingan jumlah data sebelum diaugmentasi

Proses augmentasi dilakukan dengan menggunakan *library* “nlpaug”, *tweets* – *tweets* pada kategori positif diambil, kemudian masing – masing *tweets* tersebut diaugmentasi dengan menggantikan kata – kata yang ada pada kalimat *tweets* tersebut dengan sinonim atau kata – kata yang memiliki arti serupa, augmentasi berupa substitusi ini dilakukan untuk memperluas variasi kata yang ada pada *tweets* tanpa mengubah makna dari *tweets* tersebut. Proses augmentasi ini kemudian menggandakan jumlah *tweets* pada kategori positif yang sebelumnya berjumlah 1958 *tweets* menjadi 3916 *tweets*, perubahan tersebut dapat dilihat pada Gambar 4.8 dimana jumlah *tweets* pada masing – masing kategori hampir setara. Tentunya dengan dilakukan augmentasi ini maka jumlah *dataset* yang dimiliki juga bertambah. Dengan total sebelumnya berjumlah 5938 *tweets* menjadi 7896 *tweets*. Pengujian ini nantinya dipadu padankan dengan beberapa pengujian lainnya.



Gambar 4.12 Perbandingan jumlah data setelah diaugmentasi

Pengujian selanjutnya dilakukan untuk mengetahui bagaimana hasil akurasi dari proses klasifikasi yang dilakukan terhadap data yang telah melewati proses *stemming* dengan data yang digunakan tanpa melewati proses *stemming*. Pada pengujian sebelumnya, *dataset* dibagi menjadi 2 bagian yakni *dataset* yang belum diaugmentasi serta *dataset* yang telah diaugmentasi, pada skenario pengujian kedua ini, pada masing – masing *dataset* tersebut ditambahkan lagi dengan skenario *stemming* dan *non-stemming*, sehingga kombinasi skenario uji coba yang dilakukan menjadi 4 kombinasi yakni *dataset* tanpa augmentasi yang telah dilakukan *stemming*, *dataset* tanpa augmentasi serta tidak dilakukan *stemming*, *dataset* augmentasi dengan *stemming*, dan *dataset* augmentasi yang tidak dilakukan *stemming*. Gambar 4.9 menampilkan kombinasi skenario yang dilakukan dari pengujian pertama hingga kedua.



Gambar 4.13 Skenario pengujian pertama hingga kedua

Pada skenario pengujian selanjutnya, uji coba yang dilakukan adalah untuk melihat bagaimana kombinasi dari kelima *hyperparameter* yang telah ditentukan

terhadap hasil akurasi dari klasifikasi model yang diperoleh nantinya, apakah kombinasi optimal dari *hyperparameter* yang ada berbeda pada setiap skenario yang ada ataukah sama pada setiap skenario yang diujikan, pada uji coba ini kombinasi *hyperparameter* terbaik dari setiap skenario dicari nilai *score*-nya. Kombinasi *hyperparameter* yang ada berjumlah sebanyak 8 kombinasi yang diperoleh melalui hasil perkalian dari 2 variasi *score hyperparameter* "n_estimator", 2 variasi *score hyperparameter* "learning_rate", 2 variasi *score hyperparameter* "max_depth", 1 variasi *score hyperparameter* "subsample", dan 1 variasi *score hyperparameter* "gamma". Pencarian kombinasi ini menggunakan metode "gridsearchCV" yang tersedia pada library "sklearn".

Selanjutnya, uji coba dilakukan menggunakan *dataset* yang telah dikumpulkan pada penelitian ini, *dataset* diuji dengan skenario – skenario yang telah dijabarkan pada paragraf - paragraf sebelumnya, proses pengujian ini dilakukan dengan menggunakan model klasifikasi "eXtreme Gradient Boosting (XGBoost)" dan menerapkan metode *cross validation* dengan masing - masing nilai K yang berbeda yaitu 5 dan 10. Kemudian, berdasarkan dari pengujian yang telah dilakukan, maka diperoleh hasil yaitu *accuracy* (akurasi), *precision* (presisi), dan *recall* yang dimana hasil – hasil tersebut ditampilkan pada tabel dengan masing – masing jenis skenario yang dilakukan dan dijabarkan juga nilai rata - ratanya.

1. Pengujian pada *dataset* yang belum diaugmentasi

Tabel 4.1 Pengujian pada *dataset* yang tidak diaugmentasi dan telah distemming dengan k-fold = 5

Fold	Dataset Tanpa Augmentasi Dengan Preprocess Stemming		
	Akurasi	Presisi	Recall
1	82.74%	81.11%	78.90%
2	83.00%	83.77%	76.82%
3	83.59%	82.51%	79.34%
4	82.31%	81.27%	77.37%
5	84.08%	84.33%	78.69%
	Rata – Rata (%)		
	83.14%	82.60%	78.23%

Pada Tabel 4.1 ditampilkan hasil pengujian pada *dataset* yang belum diaugmentasi, namun pada *tweets – tweets* nya telah melewati *preprocessing* termasuk dengan

stemming. Nilai *fold* yang digunakan yaitu $K = 5$, setelah dilakukan pengujian diperoleh rata – rata nilai akurasi sama dengan 83.14%, presisi 82.60%, dan *recall* 78.23%.

Tabel 4.2 Pengujian pada dataset yang tidak diaugmentasi dan telah distemming dengan k-fold = 10

Fold	Dataset Tanpa Augmentasi Dengan Preprocess Stemming		
	Akurasi	Presisi	Recall
1	84.01%	82.06%	81.46%
2	85.02%	84.11%	81.18%
3	83.00%	83.23%	77.21%
4	83.33%	83.37%	77.85%
5	81.48%	79.94%	78.69%
6	85.69%	85.31%	81.55%
7	86.87%	86.59%	83.08%
8	79.97%	78.49%	74.43%
9	83.31%	84.20%	77.10%
10	85.50%	85.69%	80.70%
Rata – Rata (%)			
	83.82%	83.30%	79.14%

Pada Tabel 4.2 ditampilkan hasil pengujian pada *dataset* yang belum diaugmentasi, namun pada *tweets – tweets* nya telah melewati *preprocessing* termasuk dengan *stemming*. Nilai *fold* yang digunakan yaitu $K = 10$, setelah dilakukan pengujian diperoleh rata – rata nilai akurasi sama dengan 83.82%, presisi 83.30%, dan *recall* 79.14%.

Tabel 4.3 Pengujian pada dataset yang tidak diaugmentasi dan tidak distemming dengan k-fold = 5

Fold	Dataset Tanpa Augmentasi Tanpa Preprocess Stemming		
	Akurasi	Presisi	Recall
1	82.32%	80.58%	78.46%
2	83.16%	84.31%	76.82%
3	83.33%	82.27%	78.95%
4	81.55%	81.12%	75.64%
5	82.14%	81.89%	76.34%
Rata – Rata (%)			
	82.50%	82.03%	77.24%

Pada Tabel 4.3 ditampilkan hasil pengujian pada *dataset* yang belum diaugmentasi serta pada *tweets – tweets* nya belum melewati tahapan *stemming*. Nilai *fold* yang digunakan yaitu $K = 10$, setelah dilakukan pengujian diperoleh rata – rata nilai akurasi sama dengan 82.50%, presisi 82.03%, dan *recall* 77.24%

Tabel 4.4 Pengujian pada dataset yang tidak diaugmentasi dan tidak distemming dengan k-fold = 10

Fold	Dataset Tanpa Augmentasi Tanpa Preprocess Stemming		
	Akurasi	Presisi	Recall
1	81.14%	78.94%	77.64%
2	85.02%	84.58%	80.66%
3	82.66%	84.10%	75.93%
4	81.65%	81.69%	75.43%
5	79.12%	77.34%	73.55%
6	85.86%	85.09%	82.20%
7	85.86%	86.02%	81.29%
8	80.64%	79.98%	74.55%
9	82.97%	83.53%	76.85%
10	84.32%	83.79%	79.68%
	Rata – Rata (%)		
	82.92%	82.51%	77.78%

Pada Tabel 4.4 ditampilkan hasil pengujian pada *dataset* yang belum diaugmentasi serta pada *tweets – tweets* nya belum melewati tahapan *stemming*. Nilai *fold* yang digunakan yaitu $K = 10$, setelah dilakukan pengujian diperoleh bahwa rata – rata nilai akurasi sama dengan 82.92%, presisi 82.03%, dan *recall* 77.78%.

Berdasarkan pada tabel – tabel di atas yang telah dipaparkan pada skenario dengan jenis *dataset* yang belum dilakukan augmentasi, diperoleh hasil bahwa nilai akurasi, presisi dan *recall* yang tertinggi berada pada pengujian dengan *dataset* yang telah melewati proses *stemming* dan menggunakan nilai *K-Fold* sama dengan 10 dengan rata - rata hasil akurasi senilai 83.82%, presisi 83.30%, dan *recall* 79.14%. Apabila dilihat dari keempat tabel yang telah dipaparkan pada *dataset* yang belum diaugmentasi ini, rata – rata dari nilai akurasi, presisi, dan *recall* yang diperoleh pada masing – masing skenario dengan menggunakan *dataset* ini yaitu senilai 83.09% pada akurasi dengan nilai standar

deviasi 0.48%, 82.61% pada presisi dengan nilai standar deviasi 0.45%, dan 78.10% pada *recall* dengan nilai standar deviasi 0.70%.

2. Pengujian pada *dataset* yang telah diaugmentasi

Tabel 4.5 Pengujian pada dataset yang diaugmentasi dan telah distemming dengan k-fold = 5

Fold	Dataset Augmentasi Dengan Preprocess Stemming		
	Akurasi	Presisi	Recall
1	77.09%	78.92%	76.99%
2	79.23%	81.67%	79.11%
3	86.19%	86.32%	86.17%
4	88.41%	88.42%	88.42%
5	90.56%	90.57%	90.57%
	Rata – Rata (%)		
	84.30%	85.18%	84.25%

Pada Tabel 4.5 ditampilkan hasil pengujian pada *dataset* yang telah diaugmentasi serta pada *tweets – tweetsnya* telah melewati proses *stemming*. Nilai *fold* yang digunakan yaitu K = 5, setelah dilakukan pengujian diperoleh bahwa rata – rata nilai akurasi sama dengan 84.30%, presisi 85.18%, dan *recall* 84.25%.

Tabel 4.6 Pengujian pada dataset yang diaugmentasi dan telah distemming dengan k-fold = 10

Fold	Dataset Augmentasi Dengan Preprocess Stemming		
	Akurasi	Presisi	Recall
1	77.59%	78.28%	77.53%
2	79.37%	82.02%	79.26%
3	82.15%	84.07%	82.06%
4	80.00%	81.67%	79.91%
5	78.73%	79.57%	78.67%
6	92.91%	92.99%	92.93%
7	90.24%	90.28%	90.23%
8	89.10%	89.12%	89.11%
9	91.76%	91.80%	91.75%
10	90.87%	90.90%	90.89%

	Rata – Rata (%)		
	85.27%	86.07%	85.23%

Pada Tabel 4.6 ditampilkan hasil pengujian pada *dataset* yang telah diaugmentasi serta pada *tweets – tweetsnya* telah melewati proses *stemming*. Nilai *fold* yang digunakan yaitu $K = 10$, setelah dilakukan pengujian diperoleh bahwa rata – rata nilai akurasi sama dengan 85.27%, presisi 86.07%, dan *recall* 85.23%.

Tabel 4.7 Pengujian pada dataset yang diaugmentasi dan tidak distemming dengan k-fold = 5

Fold	Dataset Augmentasi Tanpa Preprocess Stemming		
	Akurasi	Presisi	Recall
1	76.33%	78.39%	76.23%
2	77.14%	79.70%	77.02%
3	85.43%	85.62%	85.40%
4	88.22%	88.23%	88.21%
5	89.87%	89.87%	89.86%
	Rata – Rata (%)		
	83.40%	84.36%	83.34%

Pada Tabel 4.7 ditampilkan hasil pengujian pada *dataset* yang telah diaugmentasi namun pada *tweets – tweetsnya* tidak dilakukan *stemming*. Nilai *fold* yang digunakan yaitu $K = 5$, setelah dilakukan pengujian diperoleh bahwa rata – rata nilai akurasi sama dengan 83.40%, presisi 84.36%, dan *recall* 83.34%.

Tabel 4.8 Pengujian pada dataset yang diaugmentasi dan tidak distemming dengan k-fold = 10

Fold	Dataset Augmentasi Tanpa Preprocess Stemming		
	Akurasi	Presisi	Recall
1	77.97%	78.44%	77.93%
2	78.48%	81.10%	78.37%
3	81.52%	84.00%	81.42%
4	77.59%	79.63%	77.49%
5	76.71%	77.86%	76.63%
6	92.28%	92.32%	92.29%
7	88.97%	89.12%	88.95%
8	88.47%	88.47%	88.46%
9	91.00%	91.18%	90.97%
10	90.11%	90.15%	90.13%
	Rata – Rata (%)		
	84.31%	85.23%	84.26%

Pada Tabel 4.8 ditampilkan hasil pengujian pada *dataset* yang telah diaugmentasi namun pada *tweets – tweetsnya* tidak dilakukan *stemming*. Nilai *fold* yang digunakan yaitu K = 10, setelah dilakukan pengujian diperoleh bahwa rata – rata nilai akurasi sama dengan 84.31%, presisi 85.23%, dan *recall* 84.26%.

Berdasarkan dari apa yang telah dipaparkan pada keempat tabel yang berada pada kategori *dataset* yang telah diaugmentasi dengan skenarionya masing – masing, diperoleh hasil pengujian bahwa nilai akurasi, presisi, dan *recall* yang tertinggi berada pada pengujian dengan *dataset* yang telah melewati proses *stemming* dengan menggunakan nilai K-Fold sama dengan 10, hasil rata – rata dari pengujian tersebut yakni akurasi senilai 85.27%, presisi 86.07%, dan *recall* 85.23%. Sementara apabila dilihat dari keempat tabel yang telah dipaparkan, rata – rata dari nilai akurasi, presisi, dan *recall* yang diperoleh pada masing – masing skenario dengan menggunakan *dataset* yang telah diaugmentasi yaitu senilai 84.32% pada akurasi dengan nilai standar deviasi 0.66%, 85.21% pada presisi dengan nilai standar deviasi 0.60%, dan 84.27% pada *recall* dengan nilai standar deviasi 0.67%.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan dari penelitian yang telah dilakukan, maka diperoleh hasil penelitian yang dapat ditarik kesimpulannya bahwa:

1.
2.
3.

5.2 Saran

Berdasarkan dari penelitian yang telah dilakukan, maka dengan berikut ini dipaparkan beberapa saran untuk perbaikan atau pengembangan yang bisa dilakukan pada penelitian kedepannya :

1.
2.
3.

DAFTAR PUSTAKA

- [1] W. Zendrato, “Gerakan mencegah daripada mengobati terhadap pandemi covid-19,” *Jurnal Education and development*, vol. 8, no. 2, hlm. 242–242, 2020, [Daring]. Tersedia pada: <https://www.sehatq.com/artikel/bahaya-virus->
- [2] S. S. Indriani dan D. Prasanti, “Analisis konvergensi simbolik dalam media sosial youth group terkait kasus COVID-19 di Indonesia,” *Jurnal Kajian Komunikasi*, vol. 8, no. 2, hlm. 179–193, 2020.
- [3] L. Agustino, “Analisis Kebijakan Penanganan Wabah Covid-19: Pengalaman Indonesia,” *Jurnal Borneo Administrator*, vol. 16, no. 2, hlm. 253–270, Agu 2020, doi: 10.24258/jba.v16i2.685.
- [4] M. Mawar, L. Andriyani, A. Gultom, dan K. Ketiera, “Dampak Sosial Ekonomi Kebijakan Pemberlakuan Pembatasan Kegiatan Masyarakat (PPKM) di Indonesia,” Tangerang Selatan, 2021. [Daring]. Tersedia pada: <http://jurnal.umj.ac.id/index.php/semnaslit>
- [5] R. Yanuarti, “Jurnal Sistem dan Teknologi Informasi Analisis Media Sosial Twitter Terhadap Topik Vaksinasi Covid-19,” vol. 6, no. 2, 2021, [Daring]. Tersedia pada: <http://jurnal.unmuhjember.ac.id/index.php/JUSTINDO>
- [6] S. Azeharie, “Analisis penggunaan Twitter sebagai media komunikasi selebritis di Jakarta,” *Jurnal Komunikasi*, vol. 6, no. 2, hlm. 83–98, 2014, [Daring]. Tersedia pada: <https://journal.untar.ac.id/index.php/komunikasi/article/view/35>
- [7] R. S. D. Puspita dan G. Gumelar, “Pengaruh empati terhadap perilaku prososial dalam berbagi ulang informasi atau retweet kegiatan sosial di jejaring sosial twitter,” *Jurnal Penelitian dan Pengukuran Psikologi: JPPP*, vol. 3, no. 1, hlm. 1–7, 2014.
- [8] N. D. Asih dan M. Rosit, “Opini Publik di Media Sosial: Analisis Isi Opini Kandidat Ahok-Djarot dan Anies-Sandi di Twitter,” *CoverAge: Journal of Strategic Communication*, vol. 8, no. 2, hlm. 45–56, Mar 2018.
- [9] E. D. Liddy, “Natural Language Processing,” 2001, [Daring]. Tersedia pada: <https://surface.syr.edu/istpub>
- [10] L. Ardiani, H. Sujaini, dan T. Tursina, “Implementasi Sentiment Analysis Tanggapan Masyarakat Terhadap Pembangunan di Kota Pontianak,” *Jurnal Sistem*

- dan Teknologi Informasi (Justin)*, vol. 8, no. 2, hlm. 183, Apr 2020, doi: 10.26418/justin.v8i2.36776.
- [11] E. M. Sipayung, H. Maharani, dan I. Zefanya, “Perancangan Sistem Analisis Sentimen Komentar Pelanggan Menggunakan Metode Naive Bayes Classifier,” *Jurnal Sistem Informasi (JSI)*, vol. 8, no. 1, hlm. 2355–4614, 2016, [Daring]. Tersedia pada: <http://ejournal.unsri.ac.id/index.php/jsi/index>
 - [12] A. Putra, D. Haeirudin, H. Khairunnisa, dan R. Latifah, “Analisis Sentimen Masyarakat Terhadap Kebijakan PPKM Pada Media Sosial Twitter Menggunakan Algoritma Svm,” *Prosiding Semnastek*, 2021.
 - [13] T. Krisdiyanto, E. Maricha, dan O. Nurharyanto, “Analisis Sentimen Opini Masyarakat Indonesia Terhadap Kebijakan PPKM pada Media Sosial Twitter Menggunakan Naïve Bayes Clasifiers,” *Jurnal CoreIT*, vol. 7, no. 1, 2021.
 - [14] B. Pratama dkk., “Sentiment Analysis of the Indonesian Police Mobile Brigade Corps Based on Twitter Posts Using the SVM and NB Methods,” dalam *Journal of Physics: Conference Series*, Institute of Physics Publishing, Mei 2019. doi: 10.1088/1742-6596/1201/1/012038.
 - [15] F. F. Rachman dan S. Pramana, “Analisis Sentimen Pro dan Kontra Masyarakat Indonesia tentang Vaksin COVID-19 pada Media Sosial Twitter,” *Health Information Management Journal ISSN*, vol. 8, no. 2, hlm. 2655–9129, 2020.
 - [16] A. P. Giovani, A. Ardiansyah, T. Haryanti, L. Kurniawati, dan W. Gata, “Analisis Sentimen Aplikasi Ruang Guru Di Twitter Menggunakan Algoritma Klasifikasi,” *Jurnal Teknoinfo*, vol. 14, no. 2, hlm. 115–123, Jul 2020, doi: 10.33365/JTI.V14I2.679.
 - [17] S. Ainin, A. Feizollah, N. B. Anuar, dan N. A. Abdullah, “Sentiment analyses of multilingual tweets on halal tourism,” *Tour Manag Perspect*, vol. 34, hlm. 100658, Apr 2020, doi: 10.1016/J.TMP.2020.100658.
 - [18] E. Podasca, “Predicting the Movement Direction of OMXS30 Stock Index Using XGBoost and Sentiment Analysis,” 2021, [Daring]. Tersedia pada: www.bth.se
 - [19] R. DwiYansaputra, G. Satya Nugraha, F. Bimantoro, dan A. Aranta, “Deteksi SMS Spam Berbahasa Indonesia menggunakan TF-IDF dan Stochastic Gradient Descent Classifier,” *Jurnal Teknologi Informasi, Komputer, dan Aplikasinya (JTIKA)*, vol. 3, no. 2, hlm. 200–207, 2021, [Daring]. Tersedia pada: <http://jtika.if.unram.ac.id/index.php/JTIKA/>

- [20] Y. Mejova, V. Shirsat, dan R. S. Jagdale, "Hybrid sentiment analysis framework for a morphologically rich language," *J Intell Inf Syst*, vol. 46, hlm. 599–620, 2016.
- [21] D. Rustiana dan N. Rahayu, "Analisis Sentimen Pasar Otomotif Mobil: Tweet Twitter Menggunakan Naïve Bayes," *Simetris: Jurnal Teknik Mesin, Elektro Dan Ilmu Komputer*, vol. 8, no. 1, hlm. 113–120, 2017.
- [22] v. Smith, *Go Web Scraping Quick Start Guide: Implement the power of Go to scrape and crawl data from the web*. Packt Publishing Ltd, 2019.
- [23] A. H. Tri Jaka, "Preprocessing Text untuk Meminimalisir Kata yang Tidak Berarti dalam Proses Text Mining," Semarang, 2015.
- [24] M. Anjali dan G. Jivani, "A Comparative Study of Stemming Algorithms," *Int. J. Comp. Tech. Appl*, vol. 2, no. 6, hlm. 1930–1398, 2011, [Daring]. Tersedia pada: www.ijcta.com
- [25] M. Astiningrum dan K. S. Batubulan, "Implementasi Analisis Sentimen Twitter Mengenai Opini Masyarakat Terhadap Rkuhp Tahun 2019," *Seminar Informatika Aplikatif Polinema*, 2020.
- [26] G. Gupta dan M. Sumit, "Text Document Tokenization for Word Frequency Count using Rapid Miner (Taking Resume as an Example)," Patiala (Punjab), 2015. [Daring]. Tersedia pada: www.ijcaonline.org
- [27] A. T. Ni'mah dan A. Z. Arifin, "Perbandingan Metode Term Weighting terhadap Hasil Klasifikasi Teks pada Dataset Terjemahan Kitab Hadis," *Rekayasa*, vol. 13, no. 2, hlm. 172–180, Agu 2020, doi: 10.21107/rekayasa.v13i2.6412.
- [28] A. N. Assidyk, Setiawan, E. Budi, dan K. Isman, "Analisis Perbandingan Pembobotan TF-IDF dan TF-RF pada Trending Topic di Twitter dengan Menggunakan Klasifikasi K-Nearest Neighbor," *eProceedings of Engineering*, vol. 7, no. 2, 2020.
- [29] T. Chen dan C. Guestrin, "XGBoost: A scalable tree boosting system," dalam *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery, Agu 2016, hlm. 785–794. doi: 10.1145/2939672.2939785.
- [30] W. F. Mustika, H. Murfi, dan Y. Widyaningsih, "Analysis Accuracy of XGBoost Model for Multiclass Classification - A Case Study of Applicant Level Risk Prediction for Life Insurance," dalam *2019 5th International Conference on Science in Information Technology (ICSITech)*, 2019, hlm. 71–77. doi: 10.1109/ICSITech46713.2019.8987474.

- [31] M. Riza Kurniawanda, F. Adline, dan T. Tobing, “Analysis Sentiment Cyberbullying in Instagram Comments with XGBoost Method,” *International Journal of New Media Technology*, vol. 9, no. 1, hlm. 28, 2022.
- [32] D. Wahyudi, T. Susyanto, dan D. Nugroho, “Implementasi dan analisis algoritma stemming nazief & adriani dan porter pada dokumen berbahasa indonesia,” *Jurnal Ilmiah SINUS*, vol. 15, no. 2, 2017.
- [33] S. T. Simanjuntak, “Analisis Sentimen pada Layanan Gojek Indonesia Menggunakan Xtreme Gradient Boosting,” Universitas Sumatera Utara, 2021.
- [34] R. Z. Fadillah, A. Irawan, M. Susanty, dan I. Artikel, “Data Augmentasi Untuk Mengatasi Keterbatasan Data Pada Model Penerjemah Bahasa Isyarat Indonesia (BISINDO),” *JURNAL INFORMATIKA*, vol. 8, no. 2, 2021, [Daring]. Tersedia pada: <http://ejournal.bsi.ac.id/ejurnal/index.php/ji>
- [35] M. Lan, “A new term weighting method for text categorization,” *PhD Theses, School of Computing, National University of Singapore*, 2007.
- [36] “Imbalanced Data | Machine Learning | Google for Developers.” <https://developers.google.com/machine-learning/data-prep/construct/sampling-splitting/imbalanced-data> (diakses 18 Juni 2023).
- [37] S. Putatunda dan K. Rama, “A comparative analysis of hyperopt as against other approaches for hyper-parameter optimization of XGBoost,” dalam *ACM International Conference Proceeding Series*, Association for Computing Machinery, Nov 2018, hlm. 6–10. doi: 10.1145/3297067.3297080.
- [38] K. Budholiya, S. K. Shrivastava, dan V. Sharma, “An optimized XGBoost based diagnostic system for effective prediction of heart disease,” *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 7, hlm. 4514–4523, Jul 2022, doi: 10.1016/j.jksuci.2020.10.013.