

Terbit online pada laman : <http://teknosi.fti.unand.ac.id/>

Jurnal Nasional Teknologi dan Sistem Informasi

| ISSN (Print) 2460-3465 | ISSN (Online) 2476-8812 |



Artikel Penelitian

Pebandingan Metode *Decision Tree* dan XGBoost untuk Klasifikasi Sentimen Vaksin Covid-19 di *Twitter*

Habib Hakim Sinaga^{a,*}, Surya Agustian^b

^{ab} Program Studi Teknik Informatika, Universitas Islam Negeri Sultan Syarif Kasim Riau, Jl. H.R. Soebrantas No. 155 Km 15, Simpang Baru, Tampan, 28293, Pekanbaru, Riau, Indonesia

INFORMASI ARTIKEL

Sejarah Artikel:

Diterima Redaksi: 19 Oktober 2022

Revisi Akhir: 15 Desember 2022

Diterbitkan Online: 31 Desember 2022

KATA KUNCI

Decision tree,
XGBoost,
Vaksin Covid-19,
Klasifikasi sentimen

KORESPONDENSI

E-mail: 11651100031@students.uin-suska.ac.id

A B S T R A C T

Pemerintah Indonesia melaksanakan vaksinasi dalam upaya pencegahan virus COVID-19. Namun upaya tersebut memicu pro dan kontra dalam masyarakat. Pro dan kontra tersebut dapat dikatakan sebagai sentimen. Sentimen dapat diungkapkan di berbagai media, salah satunya adalah media sosial. Teknik yang digunakan untuk mendeteksi sentimen pada media sosial salah satunya adalah klasifikasi teks dengan *machine learning*. Penelitian ini akan membandingkan *Decision tree* dan XGBoost untuk mengklasifikasikan sentimen di *twitter*. Data diperoleh dengan cara *crawling* menggunakan pemrograman *python* dan *Twitter API*. Data diberi label dengan teknik *crowdsourcing* dan *majority voting*. Data yang digunakan setelah diseimbangkan adalah 6000 data latih, 778 data validasi dan 400 data uji. Hasil pengujian *Decision tree* dan XGBoost mendapatkan hasil terbaik pada model XGBoost dengan nilai akurasi sebesar 66% dan *f1-score* sebesar 57%. Hasil ini juga merupakan yang terbaik dibanding metode yang digunakan pada penelitian sebelumnya dengan *dataset* yang sama.

1. PENDAHULUAN

Corona Virus, Disease 19 (COVID-19) adalah sebuah penyakit yang disebabkan oleh virus *Severe Acute Respiratory Syndrome Coronavirus 2* (SARS-COV2) yang mengakibatkan infeksi pada bagian saluran pernapasan atas dengan tingkat ringan hingga sedang. Virus ini merupakan varian jenis baru hasil yang memiliki tingkat penularan lebih tinggi dibandingkan varian sebelumnya. Pada manusia, virus korona menyebabkan infeksi saluran pernapasan seperti pilek, meskipun beberapa bentuk penyakit seperti MERS dan SARS yang sifatnya lebih mematikan [1].

Virus tersebut diyakini berasal dari kelelawar yang muncul di Wuhan, China pada tahun 2019 dan telah menyebar dengan sangat cepat melalui interaksi antar manusia, menjangkau 188

negara dan 25 negara di seluruh dunia sejak pertama kali diumumkan. Berdasarkan laporan Satgas Covid-19 Indonesia, per 19 Juni 2021 terdapat 130.096 kasus aktif, 54.043 kematian, dan 1.779.127 orang dari jumlah kasus terkonfirmasi sebanyak 1.963.266 orang terhitung sejak diumumkannya kasus pertama pada 2 Maret 2020.

Dampak dari wabah pandemi ini telah menimbulkan banyak gejala di berbagai belahan dunia, termasuk Indonesia. Penutupan sekolah, fasilitas umum, pembatasan transportasi, layanan kota dan banyak lagi. Di Indonesia, pemerintah mengadakan Pembatasan Sosial Berskala Besar (PSBB) [2].

Kegiatan vaksinasi ini mendapat respon dari masyarakat. Masyarakat memberikan respon dan opininya di berbagai media. Salah satu media yang banyak digunakan oleh masyarakat untuk memberikan pendapatnya terhadap sesuatu adalah media sosial.

Twitter merupakan salah satu media sosial yang populer di Indonesia. Indonesia menempati peringkat 5 pengguna Twitter terbesar di dunia. Berdasarkan data dari PT Bakrie Telecom, Indonesia memiliki 19,5 juta pengguna aktif¹. Ada banyak informasi yang ada di Twitter, tidak terkecuali informasi kegiatan vaksinasi. Informasi kegiatan vaksinasi pada media sosial Twitter ini juga menimbulkan banyak respon pro dan kontra.

Respon pro dan kontra di masyarakat berkembang menjadi sebuah sentimen. Sentimen merupakan opini, sikap, dan perasaan seseorang tentang produk, layanan, organisasi, kepribadian, topik atau masalah, dan peristiwa. Klasifikasi Sentimen merupakan bidang studi yang mengklasifikasikan pendapat, sikap, dan perasaan orang terhadap masalah dan peristiwa yang terjadi dalam produk, layanan, organisasi, kepribadian, dan komunitas itu sendiri. ;Klasifikasi sentimen; dapat mengelompokkan teks yang ada didalam suatu kalimat, pendapat, opini maupun dokumen. Klasifikasi sentimen dilakukan untuk melihat apakah suatu pendapat bernilai positif, negatif ataupun netral [3].

Banyaknya jumlah *tweet* dan komentar di *Twitter* menyebabkan sulitnya melakukan klasifikasi sentimen jika dilakukan secara manual. Sehingga, teknik yang digunakan adalah klasifikasi dengan *machine learning*.

Metode *machine learning* yang populer antara lain *Naïve Bayes*, *Support Vector Machine*, serta *Decision Tree*. Klasifikasi sentimen terhadap vaksin COVID-19[dengan metode *Naïve Bayes* yang dilakukan oleh [4] dengan memilih kombinasi teks *preprocessing*, *balancing dataset*, serta *parameter tuning* menghasilkan akurasi terbaik 61% dan *f1-score* 57,15%. Selanjutnya, klasifikasi sentimen COVID-19 menggunakan *Support Vector Machine* dengan *dataset* yang sama [5] mendapatkan model terbaik dengan akurasi 65% dan *f1-score* sebesar 56,81%. Klasifikasi sentimen dengan teknik *deep learning* menggunakan *Long-short term memory* [6] dengan *dataset* yang sama mendapatkan hasil *f1-score* 54% dengan akurasi 66%.

Klasifikasi teks menggunakan *Decision Tree* diusulkan oleh [7] untuk mendeteksi *Hate Speech* dan *Abusive Language*. Proses pemilihan kombinasi teks *preprocessing* serta *tuning parameter* memperoleh akurasi sebesar 71,03%. Penelitian dengan *dataset* yang sama menggunakan *Random Forest* [8] yang merupakan salah satu algoritma pengembangan dari *Decision Tree* mendapatkan hasil akurasi rata-rata lebih tinggi sebesar 75,96%. Metode *machine learning* yang lain seperti *K-Nearest Neighbor* pada *dataset* yang sama [9] juga mendapatkan performa lebih dari *Decision Tree* tinggi dengan akurasi 78,46 %.

Extreme Gradient Boosting (XGBosost) dan *Random Forest* memiliki struktur yang sama yaitu pohon. Selain *Random Forest*, XGBoost juga merupakan salah satu algoritma hasil pengembangan dari *Decision Tree* [10]. Algoritma XGBoost digunakan dalam penelitian tentang perbandingan algoritma SVM, *Random Forest* dan XGBoost [11]. Penelitian ini mendapatkan hasil tertinggi pada model XGBoost dengan nilai *accuracy* sebesar 82%, *recall* 70%, dan *precision* 92%.

Penelitian ini bertujuan untuk mengklasifikasikan sentimen terhadap vaksin COVID-19 dan melakukan *improve* terhadap *Decision Tree* menggunakan XGBoost serta membandingkan performa metode *Decision tree* dan XGBoost. Penelitian ini dilakukan terhadap *dataset* yang digunakan pada penelitian [4], [5], dan [6] untuk mengevaluasi dan membandingkan performa *Decision tree* dan XGBoost terhadap ketiga metode tersebut.

2. METODE

Penelitian ini terdiri dari 6 tahap penelitian secara garis besar, yaitu pengumpulan data, *text preprocessing*, *Term Frequency-Inverse Document Frequency*, *training Decision Tree* dan XGBoost, parameter *tuning*, dan pengujian seperti pada gambar berikut ini.



Gambar 1. Metode Penelitian

2.1. Pengumpulan dan Pengolahan Dataset

Data yang dikumpulkan yaitu data *tweet* bahasa Indonesia dari rentang waktu Maret 2021 hingga April 2021. Data dikumpulkan dengan cara *crawling* menggunakan pemrograman *python* dan *Twitter API*. Kata kunci yang digunakan difilter sedemikian rupa agar *tweet* yang didapatkan relevan dengan vaksin COVID-19 sehingga diperoleh data *tweet* sebanyak 13,115 data. Kata kunci yang digunakan dapat dilihat pada tabel 1.

Data kemudian diberi label positif, negatif, dan netral. Label positif merupakan label yang berisi pujian, dukungan, masukan, saran serta perasaan positif seperti kesenangan, kebahagiaan, dan kegembiraan. Label negatif biasanya mengandung keluhan, kritik, cacian, hinaan serta emosi negatif seperti ketidakpuasan, kekecewaan, frustrasi hingga kebencian. Data yang tidak masuk kriteria positif dan negatif anak dikategorikan sebagai label netral.

¹ <https://www.kominfo.go.id>

Tabel 1. Kata Kunci Pencarian *Tweet*

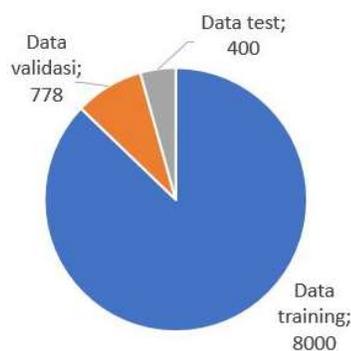
| | |
|--|--|
| Relevan dengan vaksin covid-19 | vaksin corona, vaksin covid gratis, vaksin gotong royong, vaksin covid, vaksin covid jadi sehat, vaksin nusantara, vaksin sinovac, vaksin sinovac aman, vaksin sinovac gagal, vaksin sinovac mantap, vaksin sinovac lemah, positif covid setelah vaksin, vaksin sinovac bagus, |
| Tidak mengandung kata “covid” dan merk vaksin, tapi masih dalam topik vaksinasi covid-19 | vaksin tanpa efek samping, vaksin guru, optimis-vaksin aman, vaksin lemes, sakit habis vaksin, vaksinasi indonesia, vaksin sukarela, vaksin nakes, vaksin murah, vaksin berhasil, vaksin gagal, vaksin bulan;puasa |

Data diberi label dengan cara *crowdsourcing* [12]. Setiap data *tweet* diberi label oleh 3 orang. Proses ini dikerjakan oleh 12 orang yang dibagi menjadi 4 kelompok dengan 3 orang pada masing-masing kelompok. Setiap orang akan memberi label sebanyak 3000 data. Label final yang didapat merupakan hasil *majority vote* dari tiga annotator. Data yang memiliki label positif, negatif dan netral sekaligus akan dihapus dari dataset. Sehingga didapatkan data sebanyak 9.178 data *tweet*. Tabel 2 merupakan contoh data *tweet* yang dilabel.

Tabel 2. Contoh Pelabelan Data

| Sentimen | <i>Tweet</i> |
|----------|--|
| Positif | Lancar gk ada halangan. Lega akhirnya bokap nyokap selese juga vaksin sampe dosis kedua iniðY~Š |
| Netral | 93.901 Warga Kota Bogor Sudah Disuntik Vaksin Covid-19 https://t.co/uVZEo7fDkE |
| Negatif | @Singa_Nala @JanissaryD_Last Kalau tetap jaga prokes buat apa di vaksin ..onceee...Mending gak usah paksin...prokes juga |

Data yang sudah valid kemudian dibagi menjadi 3 menjadi data latih atau data *train*, data validasi atau data *development*, dan data uji atau data *test*. Data *train* berjumlah 8000 data, data *development* berjumlah 778 dan data *test* berjumlah 400 data. Data *train* adalah data yang akan digunakan sebagai data latih pada model yang akan dibangun. Data *development* adalah data yang akan digunakan untuk validasi dari model yang akan dibangun. Data *test* adalah jenis data *tweet* yang bertujuan untuk pengujian model. Data *test* juga merupakan data yang belum dikenal oleh model sebelumnya. Gambar 2 merupakan visualisasi hasil pembagian data.



Gambar 2. Hasil Pembagian Data

2.2. Teks Preprocessing

Teks *preprocessing* adalah proses awal untuk menyeleksi teks agar menjadi lebih terstruktur dengan melalui serangkaian tahapan agar bisa digunakan untuk tahap selanjutnya. Penelitian ini akan menggunakan beberapa tahapan seperti *cleaning*, *case folding*, *stopword removal* dan *stemming* [13].

Tahap pertama adalah *cleaning*. *Cleaning* adalah tahap membersihkan dokumen dari simbol-simbol, karakter, link, dan jenis teks lain yang tidak berguna dan tidak memiliki keterkaitan informasi serta tidak mengubah informasi sentimen dalam dokumen tersebut seperti, hashtag (#), karakter (%@\$^&*#()_+”:{}<>.,?!~/[]), angka, link website (http://link.com), dan mention @username.

Berikutnya adalah tahap *case folding*. *Case folding* adalah proses merubah semua huruf yang ada pada data atau dokumen menjadi huruf kecil. Tahap *cleaning* dan *case folding* akan dilakukan bersamaan guna mengurangi biaya komputasi.

Tahap berikutnya adalah *stopword removal*. *Stopword removal* adalah proses menghapus kata-kata *stopword* yang meliputi kata sambung, kata ganti, kata depan seperti kata “dan”, “lalu”, dan “atau”.

Tahap terakhir adalah tahap *stemming*. *Stemming* merupakan tahap menghapus imbuhan lalu mengubah suatu kata tersebut menjadi kata dasar.

2.3. Term Frequency-Inverse Document Frequency

TF-IDF (*Term Frequency-Inverse Document Frequency*) merupakan varian lanjut dari *Bag of Word*. TF-IDF adalah metode yang digunakan untuk menghitung bobot atau frekuensi kemunculan setiap kata yang telah diekstrak lalu mengubah kata tersebut menjadi vektor. Vektor tersebut akan digunakan pada proses klasifikasi [14].

Term frequency (tf) merupakan frekuensi kemunculan kata pada suatu dokumen. Sementara *Wtf* adalah jumlah bobotan dari tf yang telah dihitung menggunakan logaritma. Berikut rumus menghitung *Term Frequency* :

$$Wtf_{t,d} = \begin{cases} 1 + \log_{10} tf_{t,d} & , \text{if } tf_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Document Frequency adalah jumlah dokumen yang mengandung tf. Sedangkan Inverse Document Frequency merupakan bobot kebalikan dari bobot DF. Kata yang jarang muncul di banyak dokumen mempunyai nilai IDF yang tinggi. Berikut merupakan persamaan IDF:

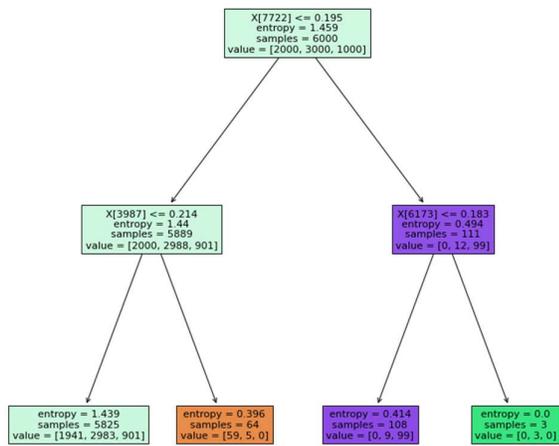
$$idf_t = \log_{10} \left(\frac{N}{df_t} \right) \tag{2}$$

Perkalian bobot TF dan bobot IDF mendapatkan hasil bobot TF-IDF. Berikut persamaan TF-IDF:

$$W_{t,d} = W_{tf,t,d} \times idf_t \tag{3}$$

2.4. Decision Tree

Decision tree merupakan metode machine learning yang berbentuk pohon upside-down yang dimulai dari root (akar) dan setiap cabangnya (node dan leaf) menampilkan berbagai pilihan. Setiap internal node merupakan pengujian terhadap atribut. Sedangkan leaf merupakan label yang didapatkan dari hasil pengujian [15]. Gambar 3 merupakan contoh visualisasi decision tree.



Gambar 3. Contoh Decision Tree Klasifikasi Teks

Dalam membangun sebuah pohon, langkah awal adalah menentukan root terpilih dengan mencari nilai Entropy terkecil. Root merupakan akar yang biasanya merupakan atribut yang memiliki value paling berpengaruh pada suatu kelas tertentu [16]. Rumus mencari nilai Entropy sebagai berikut:

$$Entropy(S) = \sum_{i=1}^n -p_i * \log_2 p_i \tag{4}$$

Kemudian periksa node mempunyai nilai 0. Jika ada, tentukan leaf yang terbentuk. Jika seluruh node bernilai 0, maka proses terhenti. Jika terdapat anggota node yang mempunyai nilai entropy lebih besar dari 0, maka proses akan diulang dari awal dengan node sebagai syarat hingga semua anggota node bernilai 0.

Nilai entropy kemudian digunakan untuk mencari nilai gain terbesar dari atribut-atribut yang ada. Rumus mencari nilai gain sebagai berikut:

$$Gain(A,S) = Entropy(S) - \sum_{i=1}^n \left(\frac{|S_i|}{|S|} \right) * Entropy(S_i) \tag{5}$$

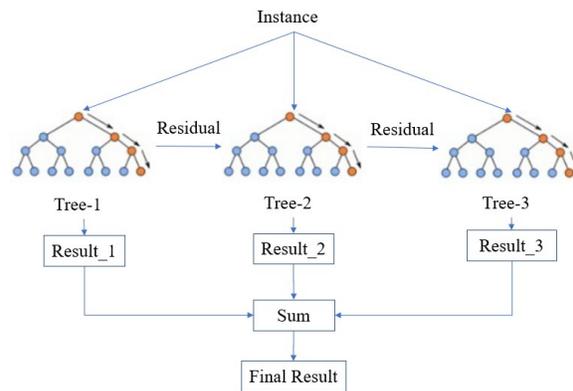
Atribut dengan nilai gain tertinggi akan dipilih menjadi root dalam sebuah pohon.

Decision tree memiliki prinsip kerja yaitu pembentukan pohon dan pembuatan aturan-aturan (rules). Decision tree membentuk pola pohon dengan metode divide and conquer. Selain menentukan root node, metode divide and conquer juga diterapkan untuk menentukan nilai gain untuk node-node pada level berikutnya. Metode tersebut diterapkan secara kontinyu sampai terbentuk daun-daun (leaf).

Prinsip kedua yaitu pembuatan rules. Aturan yang terbentuk dari pohon menghasilkan kondisi if-then. Aturan didapat dengan membaca alur pohon dari root hingga leaf. Setiap cabang akan membentuk kondisi if, sedangkan value pada leaf membentuk hasil atau then.

2.5. XGBoost

Extreme Gradient Boosting (XGBoost) merupakan salah satu machine learning yang digunakan untuk prediksi dan klasifikasi yang memiliki struktur decision tree [11]. XGBoost merupakan salah satu metode boosting yang terdiri dari beberapa decision tree yang mana pohon sebelumnya dan pohon berikutnya akan saling betergantungan [10].



Gambar 4. Konsep Dasar XGBoost (Sumber: researchgate.net)

Pada saat melakukan klasifikasi, XGBoost akan melakukan update bobot pada masing-masing pohon yang dibangun sehingga diperoleh pohon klasifikasi yang kuat [17]. Kemudian seluruh bobot pohon dijumlahkan pada saat melakukan prediksi lalu dimasukkan kedalam fungsi logistik:

$$\hat{y}_i = \sum_k^K f_k(x_i), f_k \in F \tag{6}$$

Pohon yang bobotnya dijumlahkan bertujuan untuk meminimumkan fungsi objektif. Fungsi objektif dapat ditulis sebagai berikut:

$$obj(\theta) = \sum_i^n l(\hat{y}_i, y_i) + \sum_k^K \Omega(f_k) \tag{7}$$

Dimana $\sum_i^n l(\hat{y}_i, y_i)$ merupakan *loss function* yang dapat dibedakan untuk mengukur kecocokan model dan dataset *training*. Sedangkan $\sum_k^K \Omega(f_k)$ adalah kompleksitas model [18].

Loss function dapat diukur dengan menggunakan *Mean Squared Error* (MSE)

$$MSE = \sum_i^n (y_i - \hat{y}_i)^2 \tag{8}$$

Pada saat proses *training*, model dilatih secara aditif, Adapun prosedurnya sebagai berikut:

$$\begin{aligned} \hat{y}_i^0 &= 0 \\ \hat{y}_i^1 &= f_1(x_i) = \hat{y}_i^0 + f_1(x_i) \\ \hat{y}_i^1 &= f_1(x_i) + f_2(x_i) = \hat{y}_i^1 + f_2(x_i) \\ &\dots \\ \hat{y}_i^t &= \sum_{k=1}^t f_k(x_i) = \hat{y}_i^{t-1} + f_t(x_i) \end{aligned} \tag{9}$$

Pohon yang bobotnya dijumlahkan setiap langkah bertujuan untuk mengoptimalkan fungsi objektif. Fungsi objektif dapat ditulis ulang sebagai berikut:

$$\begin{aligned} obj^{(t)} &= \sum_i^n l(\hat{y}_i, y_i) + \sum_k^K \Omega(f_k) \\ &= \sum_{i=1}^n l(y_i, \hat{y}_i^{t-1} + f_t(x_i)) + \Omega(f_t) \end{aligned} \tag{10}$$

2.6. Pengujian

Pada tahap ini akan dilakukan pengujian terhadap model *decision tree* dan XGBoost. Pengujian dilakukan setelah mendapatkan kombinasi teks *preprocessing* dan parameter terbaik masing-masing model. *Confusion matrix* akan digunakan untuk melihat nilai performa masing-masing model.

Confusion matrix merupakan cara untuk melihat apakah objek yang diprediksi itu benar atau salah. *Confusion matrix* biasanya mencari nilai akurasi, *precision*, dan *recall* [19]. Pada penelitian ini performa masing-masing model akan dinilai berdasarkan akurasi dan *f1-score*. Berikut penjelasan *confusion matrix* pada tabel 3.

Tabel 3. Confusion Matrix

| Kelas | Terklasifikasi Positif | Terklasifikasi Negatif |
|---------|------------------------|------------------------|
| Positif | TP (True Positive) | FP (False Positive) |
| Negatif | FN (False Negative) | TN (True Negative) |

$$Akurasi = \frac{(TP+TN)}{TP+TN+FP+FN} \times 100 \tag{11}$$

$$Precision = \frac{TP}{TP+FP} \times 100 \tag{12}$$

$$Recall = \frac{TP}{TP+FN} \times 100 \tag{13}$$

$$F1-score = \frac{2 \times (Recall \times Precision)}{Recall + Preci} \tag{14}$$

3. HASIL

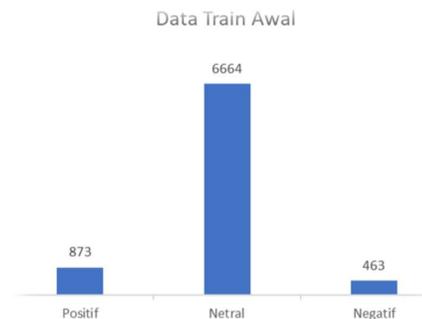
3.1. Balancing Data Train

Dari hasil proses pelabelan data secara *crowdsourcing*, ditemukan data *imbalance* antara kelas positif, negatif, dan netral. Pada data *train* terdapat 6664 kelas netral, 463 kelas positif, dan 873 kelas negatif. Pada umumnya, model *machine learning* pada kasus data *imbalance* hanya akan berhasil mendeteksi kelas yang dominan yaitu kelas netral. Oleh karena itu, dilakukan *balancing* terhadap data *train*.

Langkah awal proses ini adalah menyeleksi data kelas netral dengan memilih data yang mempunyai jumlah kata lebih dari 8. Didapat sebanyak 5914 data netral.

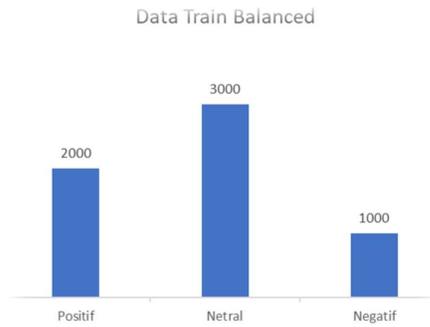
Selanjutnya kelas netral dipangkas menjadi 3000 data. Hal ini berguna untuk persiapan *oversampling* kelas positif dan kelas negatif. Data kelas positif dan negatif akan menggunakan *library RandomOverSampler* pada tahap *oversampling* ini. *RandomOverSampler* akan menambah sampel data minoritas secara acak sebanyak jumlah data mayoritas yaitu data berlabel netral yang berjumlah 3000, sehingga masing-masing kelas berjumlah 3000 data.

Data yang sudah melaluji tahap *oversampling* mempunyai rasio 1:1:1, yang mana data tersebut berpotensi *overfitting* karena rasio awal data adalah 1:14:2. Data label positif dan negatif akan dipangkas agar membentuk rasio data 1:3:2.



Gambar 5. Data Train Awal

Gambar 4 dan 5 merupakan visualisasi antara data *train* awal dan data *train* yang sudah melalui proses *balancing*.



Gambar 6. Data Train yang Seimbang

Selanjutnya, seluruh proses penelitian ini akan menggunakan data *train* yang seimbang untuk membangun model.

3.2. Pencarian Kombinasi Teks Preprocessing

Pada tahap ini akan dilakukan pencarian kombinasi teks *preprocessing* terbaik. Selain untuk menyeleksi data, variasi teks *preprocessing* dilakukan untuk mencari kombinasi terbaik karena teks *preprocessing* tergantung bagaimana kondisi *dataset*, bisa jadi menyebabkan akurasi tinggi atau bahkan menurunkan nilai akurasi.

Pada tahap ini, proses *cleaning* merupakan proses yang wajib. Jadi, akan dilakukan eksperimen terhadap *stopword removal* dan *stemming*. Eksperimen yang akan dilakukan bisa dilihat pada tabel 4.

Tabel 4. Eksperimen Teks Preprocessing

| Stopword Removal | Stemming |
|------------------|----------|
| Ya | Ya |
| Ya | Tidak |
| Tidak | Ya |
| Tidak | Tidak |

Eksperimen dilakukan dengan *decision tree* dan XGBoost terhadap data validasi atau data *dev* (*development*). Tabel 5 dan 6 merupakan hasil komposisi teks *preprocessing* terbaik masing-masing model.

Tabel 5. Kombinasi Terbaik Decision Tree Terhadap Data Dev

| Stopword Removal | Stemming | Akurasi | F1-Score |
|------------------|-----------|------------|--------------|
| Ya | Ya | 76,9% | 49,1% |
| Ya | Tidak | 75% | 46,7% |
| Tidak | Ya | 79% | 49,6% |
| Tidak | Tidak | 77,6% | 47,6% |

Tabel 6. Kombinasi Terbaik XGBoost Terhadap Data Dev

| Stopword Removal | Stemming | Akurasi | F1-Score |
|------------------|-----------|--------------|--------------|
| Ya | Ya | 78,6% | 56,7% |
| Ya | Tidak | 74,5% | 51,4% |
| Tidak | Ya | 78,9% | 56,9% |
| Tidak | Tidak | 77,2% | 55,6% |

Berdasarkan eksperimen yang telah dilakukan terhadap data validasi, *decision tree* dan XGBoost sama-sama mendapatkan kombinasi teks *preprocessing* terbaik yaitu dengan menggunakan *Stemming* dan tanpa menggunakan *Stopword removal*.

3.3. Parameter Tuning

Tahap ini merupakan tahap mencari parameter terbaik dari masing-masing model *decision tree* dan XGBoost. Parameter *tuning* bertujuan untuk mencari setelan parameter terbaik yang menghasilkan *f1-score* terbaik.

Proses parameter *tuning* dilakukan dengan teknik *GridSearch*. Adapun parameter *decision tree* yang akan menjadi perbandingan yaitu *criterion* {*entropy*, *gini*}, *splitter* {*best*, *random*}, dan *max_depth* {1, 2, 3, ...20}. Setelah melakukan *tuning*, diperoleh parameter terbaik *decision tree* seperti pada tabel 7.

Tabel 7. Parameter Terbaik Decision Tree

| Parameter | Value |
|------------------|----------------|
| <i>criteion</i> | <i>entropy</i> |
| <i>splitter</i> | <i>random</i> |
| <i>max_depth</i> | 13 |

Pada XGBoost, parameter yang akan menjadi perbandingan dapat dilihat pada tabel 8.

Tabel 8. Parameter XGBoost

| Parameter | Rentang Nilai |
|-------------------------|---------------------|
| <i>subsample</i> | {0.5, 0.75, 1} |
| <i>colsample_bytree</i> | {0.3, 0.5, 0.75, 1} |
| <i>max_depth</i> | {4,5,6,7,8} |
| <i>min_child_weight</i> | {1, 3, 5} |
| <i>learning_rate</i> | {0.3, 0.1, 0.03} |

Setelah melakukan *tuning* dengan *GridSearch*, diperoleh parameter terbaik XGBoost sebagai tabel 9.

Tabel 9. Parameter Terbaik XGBoost

| Parameter | Value |
|-------------------------|-------|
| <i>subsample</i> | 1 |
| <i>colsample_bytree</i> | 0,3 |
| <i>max_depth</i> | 5 |
| <i>min_child_weight</i> | 1 |
| <i>learning_rate</i> | 0,3 |

Parameter terbaik kedua model tersebut akan digunakan pada tahap pengujian akhir terhadap data *test*.

4. PEMBAHASAN

Pada tahap ini, model final yang telah diperoleh akan dilakukan pengujian terhadap data *test*. Data *test* merupakan data yang belum dikenal oleh model pada saat melakukan training. Hasil pengujian model *decision tree* dan XGBoost setelah mendapatkan kombinasi teks *preprocessing* dan parameter terbaik dapat dilihat pada tabel berikut:

Tabel 10. Hasil Pengujian

| Metode | Akurasi | Precision | Recall | F1-score |
|-----------------|-------------|-----------|--------|-------------|
| Naïve Bayes [4] | 0.61 | 0.58 | 0.60 | 0.57 |
| SVM [5] | 0.65 | 0.61 | 0.54 | 0.56 |
| LSTM [6] | 0.66 | 0.75 | 0.53 | 0.54 |
| Decision Tree | 0.65 | 0.74 | 0.48 | 0.51 |
| XGBoost | 0.66 | 0.69 | 0.54 | 0.57 |

Decision tree memiliki akurasi 65%, lebih rendah 1% dibanding XGBoost yang mendapatkan hasil 66%. Namun perbedaan *f1-score* diantara keduanya terpaut 6% dimana XGBoost mendapatkan nilai *f1-score* lebih tinggi sebesar 57%. Decision tree juga memperoleh *f1-score* paling rendah dibanding metode machine learning yang lain.

Disisi lain XGBoost mendapatkan hasil terbaik dari keseluruhan metode yang dibandingkan. XGBoost mendapatkan nilai akurasi tertinggi 66%, yang mana nilai tersebut sama dengan akurasi tertinggi yang didapatkan dengan metode LSTM. Sedangkan nilai *f1-score* metode XGBoost mendapatkan nilai tertinggi 57%. Nilai tersebut sama dengan nilai *f1-score* tertinggi yang diperoleh dengan metode Naïve Bayes.

5. KESIMPULAN

Berdasarkan hasil pengujian yang telah dilakukan, dapat ditarik kesimpulan bahwa terlihat keunggulan metode XGBoost dibanding decision tree dalam melakukan klasifikasi sentimen pada dataset yang tidak seimbang. Pada kasus ini, XGBoost mendapatkan hasil terbaik dibanding decision tree maupun semua metode yang digunakan dengan hasil akurasi 66% dan *f1-score* 57%. Hasil penelitian ini adalah sebuah model XGBoost yang dapat mengklasifikasi sentimen terhadap vaksin COVID-19. Model ini juga bisa dijadikan perbandingan dengan metode lain dan juga bisa dikembangkan dengan membuat sistem berbasis aplikasi agar dapat memprediksi sentimen melalui input kalimat manual.

Saran yang dapat diberikan untuk peneliti selanjutnya adalah dengan menambah porsi data label positif dan negatif agar data lebih seimbang sehingga diharapkan akan meningkatkan kualitas model dan meningkatkan hasil klasifikasi, baik itu menggunakan decision tree maupun XGBoost. Selain itu, disarankan juga untuk menggunakan word embedding sebagai fitur klasifikasi teks pada penelitian berikutnya.

DAFTAR PUSTAKA

[1] N. R. Yunus dan A. Rezki, “Kebijakan Pemberlakuan Lock Down Sebagai Antisipasi Penyebaran Corona Virus Covid-19,” *SALAM: Jurnal Sosial dan Budaya Syar-i*, vol. 7, no. 3, hlm. 227–238, 2020, doi: [10.15408/sjsbs.v7i3.15083](https://doi.org/10.15408/sjsbs.v7i3.15083).

[2] A. L. Fairuz, R. D. Ramadhani, dan N. A. Tanjung, “Analisis Sentimen Masyarakat Terhadap COVID-19 Pada Media Sosial,” *Jurnal DINDA*, vol. 1, no. 1, hlm. 10–12, 2021, [Daring]. Available:

<http://journal.itelkom-pwt.ac.id/index.php/dinda/article/view/180>

[3] M. A. Fauzi dan S. Adinugroho, “Analisis Sentimen Pariwisata di Kota Malang Menggunakan Metode Naive Bayes dan Seleksi Fitur Query Expansion Ranking Image Processing View project Smart Wheelchair View project,” *Researchgate.Net*, vol. 2, no. 8, hlm. 2766–2770, 2018.

[4] P. Yohana, “Analisis Sentimen Vaksin Covid19 Menggunakan Naive Bayes,” Universitas Islam Negeri Sultan Syarif Kasim Riau, Pekanbaru, 2022.

[5] M. Rizky, “Analisis Sentimen Masyarakat Terhadap Vaksin Covid-19 Menggunakan Metode Support Vector Machine Pada Media Sosial Twitter,” Universitas Islam Negeri Sultan Syarif Kasim Riau, Pekanbaru, 2022.

[6] M. Ihsan, B. S. Negara, dan S. Agustian, “Metode LSTM (Long short term memory) untuk Klasifikasi Sentimen Vaksin Covid-19 pada Twitter,” *Digital Zone: Jurnal Teknologi Informasi Dan Komunikasi*, vol. 13(1), hlm. 1–13, 2022.

[7] F. Ihsan, “Implementasi Algoritma Decision Tree untuk Mendeteksi Multi-Label Hate Speech dan Abusive Language pada Twitter Bahasa Indonesia,” Universitas Islam Negeri Sultan Syarif Kasim Riau, 2021.

[8] A. Amri, “Implementasi Algoritma Random Forest Untuk Mendeteksi Hate Speech Dan Abusive Language Pada Twitter Bahasa Indonesia,” 2020.

[9] A. Fadilah, “Penerapan Algoritma K-Nearest Neighbor untuk Mendeteksi Ujaran Kebencian dan Bahasa Kasar pada Twitter Bahasa Indonesia,” 2021.

[10] M. Syukron, R. Santoso, dan T. Widiharah, “Perbandingan Metode Smote Random Forest Dan Smote Xgboost Untuk Klasifikasi Tingkat Penyakit Hepatitis C Pada Imbalance Class Data,” *Jurnal Gaussian*, vol. 9, no. 3, hlm. 227–236, 2020, doi: [10.14710/j.gauss.v9i3.28915](https://doi.org/10.14710/j.gauss.v9i3.28915).

[11] M. R. Givari, M. R. Sulaeman, dan Y. Umaidah, “Perbandingan Algoritma SVM, Random Forest Dan XGBoost Untuk Penentuan Persetujuan Pengajuan Kredit,” *Nuansa Informatika*, vol. 16, no. 1, hlm. 141–149, 2022, doi: [10.25134/nuansa.v16i1.5406](https://doi.org/10.25134/nuansa.v16i1.5406).

[12] R. Wardoyo, A. Musdholifah, G. Angga Pradipta, dan I. N. Hariyasa Sanjaya, “Weighted Majority Voting by Statistical Performance Analysis on Ensemble Multiclassifier,” dalam *2020 Fifth International Conference on Informatics and Computing (ICIC)*, Nov 2020, hlm. 1–8. doi: [10.1109/ICIC50835.2020.9288552](https://doi.org/10.1109/ICIC50835.2020.9288552).

[13] S. Khomsah dan A. S. Aribowo, “Model Text-Preprocessing Komentar Youtube Dalam Bahasa Indonesia,” *JURNAL RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 1, no. 10, hlm. 648–654, 2021.

[14] U. Rofiqoh, R. S. Perdana, dan M. A. Fauzi, “Analisis Sentimen Tingkat Kepuasan Pengguna Penyedia Layanan Telekomunikasi Seluler Indonesia Pada Twitter Dengan Metode Support Vector Machine dan Lexion Based Feature,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (J-PTIHK) Universitas Brawijaya*, vol. 1, no. 12, hlm. 1725–1732, 2017, [Daring]. Available: <http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/628>

- [15] I. Sutoyo, "Implementasi Algoritma Decision Tree Untuk Klasifikasi Data Peserta Didik," *Jurnal Pilar Nusa Mandiri*, vol. 14, no. 2, hlm. 217, 2018, doi: [10.33480/pilar.v14i2.926](https://doi.org/10.33480/pilar.v14i2.926).
- [16] K. Kusriani, S. Hartati, R. Wardoyo, dan A. Harjoko, "Klasifikasi Citra Dengan Pohon Keputusan," *JUTI: Jurnal Ilmiah Teknologi Informasi*, vol. 7, no. 2, hlm. 55, 2008, doi: [10.12962/j24068535.v7i2.a173](https://doi.org/10.12962/j24068535.v7i2.a173).
- [17] Z. Salam Patrous, "Evaluating XGBoost for User Classification by using Behavioral Features Extracted from Smartphone Sensors," hlm. 67, 2018, [Daring]. Available: <https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1240595&dswid=-6444>
- [18] I. Muslim dan K. Karo, "Implementasi Metode XGBoost dan Feature Importance untuk Klasifikasi pada Kebakaran Hutan dan Lahan," *Journal of Software Engineering, Information and Communication Technology*, vol. 1, no. 1, hlm. 10–16, 2020.
- [19] D. I. Baihaqi, A. N. Handayani, dan U. Pujiyanto, "Perbandingan Metode Naïve Bayes Dan C4.5 Untuk Memprediksi Mortalitas Pada Peternakan Ayam Broiler," *Simetris: Jurnal Teknik Mesin, Elektro dan Ilmu Komputer*, vol. 10, no. 1, hlm. 383–390, Apr 2019, doi: [10.24176/simet.v10i1.2846](https://doi.org/10.24176/simet.v10i1.2846).

BIODATA PENULIS



Habib Hakim Sinaga
 Pria kelahiran Sungai Pakning 22 Februari 1999. Sekarang berstatus mahasiswa jurusan Teknik Informatika Fakultas Sains dan Teknologi UIN Sultan Syarif Kasim Riau