ANALISIS SENTIMEN PADA ULASAN APLIKASI PEDULI LINDUNGI DI *GOOGLE PLAY STORE* DENGAN METODE *ADASYN-MULTINOMIAL NAIVE BAYES*

oleh

IMAM SUYUTI

M0119043

SKRIPSI

ditulis dan diajukan untuk memenuhi sebagian persyaratan

memeroleh gelar Sarjana Matematika

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS SEBELAS MARET

SURAKARTA

2023

PERNYATAAN

Dengan ini saya menyatakan bahwa skripsi saya yang berjudul "Analisis Sentimen pada Ulasan Aplikasi Peduli Lindungi di *Google Play Store* dengan Metode *ADASYN-Multinomial Naïve Bayes*" belum pernah diajukan untuk memeroleh gelar kesarjanaan pada suatu perguruan tinggi, dan sepanjang pengetahuan saya juga belum pernah ditulis atau dipublikasikan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar rujukan.

Surakarta, Januari 2023

Imam Suyuti

RINGKASAN

Imam Suyuti, 2023. ANALISIS SENTIMEN PADA ULASAN APLIKASI PEDULI LINDUNGI DI *GOOGLE PLAY STORE* DENGAN METODE *ADASYN-MULTINOMIAL NAIVE BAYES*. Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Sebelas Maret.

Setiap aplikasi terdapat *rating* dan ulasan penggguna mengenai pelayanan dan fitur-fitur yang diberikan. Ulasan yang diberikan dapat berupa saran, kritik, maupun keluhan. Ulasan dapat dikelompokkan ke dalam sentimen positif dan sentimen negatif. Pembagian kelas tersebut dapat diolah dengan klasifikasi data.

Pada penelitian ini bertujuan untuk menganalisis sentimen pada ulasan tentang aplikasi Peduli Lindungi dengan menggunakan metode *ADASYN-multinomial naïve Bayes*. *Multinomial naïve Bayes* merupakan metode *supervised learning* yang menggunakan probabilitas dan lebih difokuskan untuk klasifikasi teks. Data yang digunakan pada penelitian ini merupakan data ulasan aplikasi Peduli Lindungi dari Desember 2021 sampai April 2022. Data tersebut bersumber dari *google play store*[17]. Data yang diperoleh adalah 9257 data. Data yang diperoleh setelah *data prepocessing* adalah 9172 data. Selanjutnya data diberi label yaitu sentimen positif atau negatif dengan menggunakan *vader lexicon*.Data ulasan dilakukan pembobotan dengan menggunakan *TF-IDF*. Data *training* yang digunakan adalah 80% dari total data yaitu 7337 data, sedangkan data *testing* yang digunakan adalah 20% atau 1835 data. Data yang tidak seimbang pada data *training* dapat diatasi dengan metode *ADASYN*.

Berdasarkan hasil penelitian dan pembahasan diperoleh kesimpulan untuk data *testing* dengan 1835 data ulasan diklasifikasikan dengan benar ada 1577 data yaitu 581 data untuk kelas sentimen positif dan 996 data untuk kelas sentimen negatif. Data yang diklasifikasikan salah yaitu sentimen positif yang diklasifikasikan sebagai sentimen negatif ada 137 data dan sentimen negatif yang yang diklasifikasikan sebagai sentimen positif ada 121 data. Klasifikasi data ulasan aplikasi Peduli Lindungi di *google play store* menggunakan metode *ADASYN-multinomial naïve Bayes* menunjukkan akurasi 85,94% dan termasuk *good classification* dengan nilai *AUC* 0,851.

SUMMARY

Imam Suyuti, 2023. SENTIMENT ANALYSIS ON PEDULI LINDUNGI APP REVIEWS IN GOOGLE PLAY STORE WITH ADASYN-MULTINOMIAL NAIVE BAYES METHOD. Faculty of Mathematics and Natural Sciences, Universitas Sebelas Maret.

Each application has user ratings and reviews regarding service and the features provided. The comments given can be in the form of suggestions, criticisms, as well as complaints. Reviews can be grouped into positive sentiments and negative sentiment. The class division can be processed with data classification.

This study aims to analyze the sentiments in the reviews about the Peduli Lindungi application using the ADASYN-multinomial naïve Bayes method. Multinomial naïve Bayes is a supervised learning method that uses probability and is more focused on text classification. The data used in this study is review data the Peduli Lindungi application from December 2021 to April 2022. The data sourced from the google play store [17]. The data obtained is 9257 data. The data obtained after the data preprocessing is 9172 data. Then, data is labeled i.e. positive or negative sentiment by using vader lexicon. The review data was weighted using TF-IDF. Data the training used is 80% of the total data, namely 7337 data, meanwhile the testing data used is 20% or 1835 data. Unbalanced data in the training data can be overcome by the ADASYN method.

Based on the results of research and discussion, it was concluded that for data testing with 1835 data review classified correctly there are 1577 data namely 581 data for the positive sentiment class and 996 data for the sentiment class negative. Data that is classified incorrectly, namely positive sentiment which is classified as negative sentiment, there are 137 data and negative sentiment which is classified as negative sentiment classified as positive sentiment there are 121 data. Classification of review data the Peduli Lindungi application on the Google Play Store uses the ADASYN-multinomial naıve Bayes method which shows an accuracy of 85.94% and is included in good classification with an AUC value of 0.851.

MOTO

"Trust the process and DOA"

PERSEMBAHAN

Karya ini saya persembahkan untuk
Ibu saya Nasri, Bapak saya Rukani,
serta teman-teman yang telah memberi bantuan serta dukungan kepada saya.

PRAKATA

Bismillahirrahmanirrahim,

Puji syukur kepada Allah *Subhanahu Wa Ta'ala* atas segala rahmat dan hidayah-Nya, skripsi ini dapat terselesaikan. Sholawat serta salam selalu dihaturkan kepada Nabi Muhammad *Shallallahu'Alaihi Wa sallam*. Skripsi ini tidak akan berhasil dengan baik tanpa bantuan dari berbagai pihak. Ucapan terima kasih disampaikan kepada semua pihak yang telah membantu dalam penyusunan skripsi ini, terutama kepada

1. Dr. Dewi Retno Sari S, S.Si., M.Kom. sebagai Pembimbing I yang telah memberikan bimbingan mengenai materi dan motivasi sehingga skripsi ini dapat terselesaikan,

2. Dra. Purnami Widyaningsih, M.App.Sc. sebagai Pembimbing II yang telah memberikan bimbingan mengenai penulisan dan motivasi sehingga skripsi ini dapat terselesaikan, dan

3. keluarga serta teman-teman yang telah membantu dan senantiasa memberikan semangat dalam menyelesaikan skripsi ini.

Semoga skripsi ini bermanfaat bagi semua pembaca.

Surakarta, Juni 2023

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Penyebaran virus COVID-19 (Coronavirus disease 2019) mendorong pemerintah Indonesia untuk mengeluarkan kebijakan pengendalian penyebaran virus tersebut dengan memberlakukan pembatasan berskala besar yang disebut dengan istilah Pembatasan Sosial Berskala Besar (PSBB). PSBB ini bertujuan untuk memutus mata rantai penyebaran virus COVID-19. Beberapa fasilitas umum ditutup selama PSBB meskipun terdapat sektor vital seperti fasilitas pemerintahan, kesehatan, dan pasar atau minimarket masih tetap buka dengan menjalankan protokol kesehatan. Kebijakan tersebut didasari pada Undang-Undang No. 6 Tahun 2018 tentang Kekarantinaan Kesehatan.

Dalam rangka menangani penyebaran wabah COVID-19 di Indonesia, PT Telekomunikasi Indonesia Tbk (Telkom) dan Kementerian Komunikasi dan Informatika (Kemkominfo) bekerjasama membuat aplikasi Peduli Lindungi. Aplikasi ini digunakan untuk melindungi masyarakat yang tengah mengakses fasilitas publik agar kegiatan yang sedang dilakukan aman dan dapat terhindar dari adanya penyebaran COVID-19 (Herdiana [10]). Aplikasi ini juga digunakan untuk pelaksanaan *surveilans* kesehatan dalam menangani penyebaran COVID-19, dengan melakukan *tracing* yaitu kegiatan pelacakan terhadap orang-orang yang berkontak dengan orang-orang yang diduga terinfeksi COVID-19. Selain itu, dengan melakukan *tracking* yaitu melacak penyebaran virus corona dengan melihat siapa saja yang telah bertemu dengan orang yang positif terinfeksi virus COVID-19 dan menyelenggarakan *warning and fencing* yaitu adanya peringatan dan pengawasan dengan membatasi pergerakan seseorang yang sedang dalam karantina

atau isolasi (Nurhidayati dkk. [15]). Partisipasi masyarakat sangat dibutuhkan dengan membagikan data lokasinya saat melakukan perjalanan agar dapat dilakukan penelusuran riwayat kontak dengan orang yang positif terinfeksi virus COVID-19.

Pada setiap aplikasi, terdapat rating dan ulasan pengguna mengenai pelayanan dan fitur-fitur yang diberikan. Ulasan yang diberikan dapat berupa saran, kritik, maupun keluhan. Hal tersebut sangat bermanfaat bagi pengguna lain yang akan menggunakan aplikasi tersebut. Pengumpulan dan penyortiran data ulasan yang dilakukan tanpa algoritme atau aplikasi yang dapat mengumpulkan data tidaklah hal yang mudah karena ulasan yang tersedia pada fitur komentar di situs *Google Play* biasanya sangat banyak. Menurut Moraes *et al.* [14], metode yang cocok untuk mengumpulkan data informasi tersebut adalah metode *web scraping*. Metode *web scraping* merupakan metode yang digunakan untuk mengumpulkan informasi atau data semi-terstruktur dari *website*.

Analisis sentimen berguna untuk mengelompokkan atau menyaring antara sentimen positif dan negatif pada suatu ulasan. Dalam hal ini, digunakan metode *vader lexicon* untuk melakukan proses pelabelan secara otomatis kelas sentimen pada data ulasan berbahasa Inggris. Permasalahan yang sering muncul pada sebagian besar penelitian analisis sentimen, yaitu kebanyakan data ulasan cenderung tidak seimbang (*imbalanced dataset*) dari segi jumlah kelas tiap individu, misalnya cenderung ke arah positif atau sebaliknya. Secara umum, algoritme *machine learning* akan menghasilkan suatu model dengan tingkat sensitivitas yang rendah terhadap kelas minoritas ketika menerima *dataset* yang tidak seimbang karena hal tersebut akan menyebabkan performa klasifikasi sentimen yang dilakukan menjadi buruk.

Pada penelitian ini, akan dianalisis sentimen pada ulasan tentang aplikasi Peduli Lindungi dengan menggunakan metode *ADASYN* dan *multinomial naïve Bayes*. Metode *ADASYN* digunakan untuk menangani kasus data yang tidak seimbang. Sedangkan metode *multinomial naïve Bayes* digunakan untuk mengklasifikasi data ulasan ke dalam sentimen positif dan negatif.

1.2 Perumusan Masalah

Berdasarkan latar belakang masalah dirumuskan masalah yaitu bagaimana menganalisis sentimen pada ulasan tentang aplikasi Peduli Lindungi dengan menggunakan metode *ADASYN* dan *multinomial naïve Bayes*.

1.3 Tujuan Penelitian

Berdasarkan perumusan masalah, tujuan penelitian ini adalah menganalisis sentimen pada ulasan tentang aplikasi Peduli Lindungi dengan menggunakan metode *ADASYN* dan *multinomial naïve Bayes*.

1.4 Manfaat Penelitian

Penelitian ini diharapkan untuk menambah wawasan ilmu pengetahuan tentang implementasi metode *ADASYN* dan *multinomial naïve Bayes* dalam klasifikasi sentimen pada ulasan aplikasi Peduli Lindungi.

BAB II

LANDASAN TEORI

Bab ini terdiri dari tiga bagian yaitu tinjauan pustaka, teori penunjang, dan kerangka pemikiran. Tinjauan pustaka memuat uraian hasil-hasil penelitian yang telah dilakukan oleh peneliti terdahulu yang ada hubungannya dengan penelitian ini. Teori penunjang berisi definisi dan teori yang menjadi dasar dalam penelitian. Kemudian kerangka pemikiran berisi alur pemikiran sebagai tuntunan pemecahan masalah penelitian.

2.1 Tinjauan Pustaka

Pada tahun 2008, He *et al.* [9] meneliti tentang *ADASYN* untuk pembelajaran *dataset* yang tidak seimbang. Hasil penelitian menunjukkan bahwa hasil simulasi pada lima set data berdasarkan berbagai matrik evaluasi menunjukkan efektivitas metode *ADASYN*.

Pada tahun 2017, Song *et al.* [21] meneliti tentang pendekatan klasifikasi novel berdasarkan *naïve Bayes* untuk analisis sentimen Twitter. Hasil penelitian menunjukkan bahwa metode *naïve Bayes* menghasilkan akurasi tertinggi dibanding *maximum entropy* dan *support vector machine*. Penelitian yang dilakukan oleh Pintoko dan Muslim [18] pada tahun 2018 menyimpulkan bahwa metode *naïve Bayes* dapat menganalisis sentimen pada data ulasan jasa transportasi *online* pada Twitter. Namun penelitian ini masih terdapat beberapa kekurangan, yaitu kurangnya fitur yang digunakan pada data latih dan adanya perbedaan hasil dari pelabelan sentimen data yang dilakukan secara manual untuk menguji model dengan hasil prediksi sentimen dari hasil klasifikasi model.

Pada tahun 2019, Abbas *et al.* [1] meneliti model klasifikasi *multinomial naïve Bayes* untuk analisis sentimen. Hasil penelitian menunjukkan bahwa model

multinomial naïve Bayes dengan *term frequency inverse document frequency* (*TF-IDF*) menghasilkan akurasi yang lebih baik dibandingkan model *multinomial naïve Bayes* tanpa *TF-IDF* dalam kinerja kategorisasi teks. Pada tahun 2022, Agustina dkk. [2] mengimplementasikan algoritma *naïve Bayes* untuk analisis sentimen ulasan Shopee pada *Google Play Store*. Hasil penelitian menunjukkan bahwa metode *multinomial naïve Bayes* dengan pembagian data *hold-out* (pembagian data *training* dan *testing*) menghasilkan akurasi yang lebih baik dibandingkan pembagian data *k-fold cross validation* untuk kasus klasifikasi ulasan Shopee.

2.2 Teori Penunjang

Penelitian ini bertujuan untuk menganalisis sentimen pada ulasan tentang aplikasi Peduli Lindungi dengan menggunakan metode *ADASYN* dan *multinomial naïve Bayes*. Oleh karena itu, beberapa definisi yang mendasari penelitian perlu diuraikan. Beberapa definisi tersebut meliputi analisis sentimen, *web scraping*, *data preprocessing*, pembobotan *TF-IDF*, *VADER*, *ADASYN*, *multinomial naïve Bayes*, serta evaluasi kinerja klasifikasi.

2.2.1 Analisis Sentimen

Analisis sentimen merupakan salah satu bidang pada *text mining* yang menganalisa sebuah pendapat, opini, evaluasi, sentimen, sikap atau penilaian seseorang terhadap individu, kelompok, produk, organisasi, masalah, peristiwa atau topik (Sabily dkk. [20]). Analisis sentimen juga bisa diartikan sebagai riset komputasional dari sebuah opini dan emosi yang diekspresikan secara tekstual. Analisis sentimen biasanya digunakan untuk menganalisa produk atau organisasi dalam rangka peningkatan kualitas dari produk atau organisasi nantinya (Gunawan dkk. [7]).

Analisis sentimen dibagi menjadi dua kategori yaitu *coarse-grained sentiment analysis* dan *fined-grained sentiment analysis* (Sabily dkk. [20]).

1. *Coarse-Grained* adalah proses menganalisis sentimen sebuah dokumen secara keseluruhan. Sentimen ini ada tiga jenis yaitu positif, netral, dan negatif. *Coarse-Grained* biasanya digunakan untuk ulasan-ulasan yang berupa dokumen seperti ulasan hotel, film, dan buku.

2. *Fine-Grained* adalah proses menganalisis yang orientasinya lebih spesifik, yaitu pada kalimat di sebuah dokumen. Sentimen ini ada dua jenis yaitu positif dan negatif. *Fine-Grained* biasanya digunakan untuk ulasan-ulasan yang berupa kalimat seperti ulasan produk, aplikasi, dan jasa. Contoh dari *fined-grained* adalah "Saya benci orang itu, dia suka pamer di depan guru" (Negatif), atau bisa juga "Jalanan hari ini terasa nyaman karena tidak ada kemacetan" (Positif).

2.2.2 *Web Scraping*

Web scraping adalah metode yang digunakan untuk mengumpulkan informasi atau data semi-terstruktur dari *website* (biasanya dalam bentuk halaman *web* dalam bahasa markup, seperti HTML atau XHTML) untuk dianalisis (Turland [22]).

Proses *web scraping* dibagi menjadi tiga tahap yaitu *download content* dari halaman *web*, mengumpulkan data, dan menyimpan data dalam format *csv* atau *json*. Salah-satu bahasa yang digunakan untuk proses *web scraping* adalah *python* karena memiliki *libraries* yang memproses data dengan baik.

2.2.3 *Data Preprocessing*

Data preprocessing merupakan proses membersihkan data yang dilakukan setelah *dataset* terkumpul, agar proses pada *machine learning* menjadi lebih cepat dan akurat (Nurrohmat & Azhari [16]). Tujuan dari *data preprocessing* yaitu mengubah data teks yang awalnya tidak terstruktur menjadi data yang terstruktur. Secara umum proses tahapan *data preprocessing* dapat dilakukan sebagai berikut.

1. *Case folding* adalah proses standarisasi bentuk huruf agar tidak ada perbedaan makna. Huruf kapital akan diubah ke huruf kecil sedangkan tanda baca dan angka dihapus.

2. *Tokenizing* adalah proses pemisahan kata per kata yang tidak saling mempengaruhi dari teks dokumen.

3. *Filtering* adalah proses penyaringan atau pemilihan kata dalam dokumen. Kata yang kurang relevan disaring menggunakan *stopword*. *Stopword* merupakan kata yang muncul dalam jumlah besar dan tidak relevan (Agustina dkk. [2]). Contoh *stopword* dalam bahasa Inggris yaitu "a", "and", "in", dll.

4. *Stemming* adalah pengubahan kata berimbuhan menjadi kata dasar. Misalnya kata "program", "programs", "programer", "programing", dan "programers" akan ditransformasikan menjadi kata "program".

2.2.4 Pembobotan *Term Frequency-Inverse Document Frequency* (*TF-IDF*)

TF-IDF merupakan metode untuk menghitung bobot setiap kata yang paling umum digunakan pada *information retrieval*. Metode ini juga terkenal efisien, mudah dan memiliki hasil yang akurat (Maarif [13]). Metode ini akan menghitung nilai *Term Frequency* (*TF*) dan *Inverse Document Frequency* (*IDF*) pada setiap kata di setiap dokumen dalam korpus. Secara sederhana, metode *TF-IDF* digunakan untuk mengetahui berapa sering suatu kata muncul di dalam dokumen. Rumus untuk menghitung bobot setiap kata t di dokumen d ditulis sebagai

$$W_{dt} = (tf)_{dt} \times log(\frac{N}{(df)_t})$$

dengan

W_{dt} : bobot dokumen ke-d terhadap kata ke-t

$(tf)_{dt}$: banyaknya kata ke-t terhadap dokumen ke-d

N : total dokumen

$(df)_t$: banyak dokumen yang mengandung kata ke-t.

2.2.5 *Valence Aware Dictionary for Social Reasoning* (*VADER*)

VADER merupakan model yang digunakan dalam menganalisis sentimen dan mampu menentukan keragaman data melalui intensitas kekuatan emosional yang ada sesuai dengan kamus data *lexicon* yang tersedia (Elbagir & Yang [5]). Metode leksikal merupakan metode yang tidak memerlukan *data training* atau data yang telah dilabeli namun sudah tersedia dalam kamus lengkap dengan kepolaritasan sentimennya. Salah-satu contoh dari metode analisis sentimen secara leksikal yaitu *vader lexicon polarity detection* (Hutto & Gilbert [11]).

2.2.6 *Adaptive Synthetic Sampling Approach* (*ADASYN*)

ADASYN merupakan algoritme yang digunakan untuk menangani *dataset* yang tidak seimbang dalam klasifikasi data (He *et al.* [9]). *Dataset* yang tidak seimbang berdasarkan jumlah data tiap kelas, misalnya cenderung ke arah positif atau sebaliknya. *ADASYN* dapat menghasilkan sampel secara adaptif terhadap kelas minoritas yang dibentuk oleh distribusi data untuk mengurangi bias yang disebabkan oleh distribusi data yang tidak merata. Berikut algoritme *ADASYN*

Input

(1) *Training dataset* D_{tr} dengan m sampel $\{x_i, y_i\}$, $i = 1, \ldots, m$, dimana x_i merupakan sampel dalam ruang matriks X dimensi n dan $y_i \in Y = \{1, -1\}$ merupakan label identitas kelas yang berkaitan dengan x_i. Mendefinisikan m_s dan m_l sebagai jumlah sampel kelas minoritas dan jumlah sampel kelas mayoritas. Oleh karena itu, $m_s \leq m_l$ dan $m_s + m_l = m$.

Prosedur

(1) Menghitung tingkat ketidakseimbangan kelas

$$d = m_s/m_l \tag{2.1}$$

dimana $d \in (0, 1]$.

(2) Jika $d < d_{th}$ dimana (d_{th} merupakan penetapan *threshold* untuk derajat tolerasi maksimum dari rasio ketidakseimbangan kelas):

(a) Menghitung jumlah sampel data sintetis yang perlu dihasilkan untuk kelas minoritas

$$G = (m_l - m_s) \times \beta \tag{2.2}$$

dimana $\beta \in [0, 1]$ merupakan parameter yang digunakan untuk menentukan tingkat keseimbangan yang diinginkan setelah generalisasi data sintetis. $\beta = 1$ berarti *dataset* yang sepenuhnya seimbang dibuat setelah proses generalisasi.

(b) Untuk setiap sampel $x_i \in$ kelas minoritas, Menemukan K tetangga terdekat berdasarkan jarak euclidan pada ruang dimensi n , dan menghitung rasio r_i yang dirumuskan sebagai

$$r_i = \Delta_i/K, \qquad i = 1, \ldots, m_s \tag{2.3}$$

dimana Δ_i merupakan jumlah sampel di K tetangga terdekat dari x_i yang termasuk dalam kelas mayoritas, oleh karena itu $r_i \in [0, 1]$.

(c) Menormalisasi r_i berdasarkan $\hat{r}_i = r_i / \sum_{i=1}^{m_s} r_i$, sehingga $\hat{r}_i$ merupakan distribusi kerapatan ($\sum_i \hat{r}_i = 1$).

(d) Menghitung jumlah sampel data sintetis yang perlu dihasilkan pada setiap sampel minoritas x_i

$$g_i = \hat{r}_i \times G \tag{2.4}$$

dimana G merupakan jumlah sampel data sintetis yang perlu dihasilkan untuk kelas minoritas (2.2).

(e) Untuk setiap sampel data kelas minoritas x_i, menghasilkan sampel data sintetis sebanyak g_i.

2.2.7 *Multinomial Naïve Bayes*

Multinomial naïve Bayes merupakan metode *supervised learning* yang menggunakan probabilitas dan lebih difokuskan untuk klasifikasi teks (Liu & Ozsu [12]). *Multinomial naïve Bayes* juga memiliki fitur unik, yaitu hasil yang diperoleh untuk masing-masing kelas bersifat independen. Hal ini berarti, dari dokumen satu ke dokumen berikutnya tidak ada keterkaitannya sama sekali sehingga hasil yang didapat murni dari dokumen yang diolah itu sendiri. Probabilitas ulasan d yang memiliki kelas c ditulis sebagai

$$P(c|d) \propto P(c) \prod_{i=1}^{n_d} P(t_i|c) \tag{2.5}$$

dengan

$P(c|d)$: probabilitas suatu kelas c pada ulasan d

$P(c)$: probabilitas *prior* c

$P(t_i|c)$: probabilitas t_i pada kelas c

t_i : kata ke-i.

Rumus probabilitas *prior* kelas c ditulis sebagai

$$P(c) = \frac{N_c}{N} \tag{2.6}$$

dengan

N_c : jumlah kelas c pada seluruh dokumen

N : jumlah seluruh dokumen.

Sementara rumus *multinomial* yang digunakan dengan pembobotan kata *TF-IDF* ditulis sebagai

$$P(t_i|c) = \frac{W_{ct} + 1}{(\sum_{W' \in V} W_{ct}) + B'} \tag{2.7}$$

dengan

W_{ct} : bobot *TF-IDF* kata t pada dokumen dengan kelas c

$\sum_{W' \in V} W_{ct}$: jumlah bobot *TF-IDF* seluruh kata pada kelas c

B' : jumlah *IDF* seluruh kata pada seluruh dokumen.

Klasifikasi dilakukan setelah melewati tahap *preprocessing* dan perhitungan bobot pada data yang hasilnya akan digunakan pada proses klasifikasi. Tahapan perhitungan pada proses klasifikasi adalah sebagai berikut.

(1) Menghitung probabilitas *prior* setiap kelas dengan persamaan (2.6).

(2) Menghitung probabilitas kata ke-n dengan persamaan (2.7).

(3) Menghitung probabilitas dokumen menentukan kelas dengan persamaan (2.5).

(4) Menentukan kelas dokumen dengan memilih nilai probabilitas tertinggi.

2.2.8 Evaluasi Kinerja Klasifikasi

Sebuah sistem klasifikasi harus dinilai performanya agar dapat mengukur tingkat akurasi dari prediksi klasifikasi yang dihasilkan. Ada dua metode perhitungan yang digunakan untuk menilai performa klasifikasi yang ditunjukkan sebagai berikut.

2.2.8.1 Matriks *Confusion*

Matriks *confusion* adalah salah-satu metode evaluasi berupa tabel yang menyatakan berapa banyak data uji yang benar atau salah diklasifikasikan (Bramer [4]). Jika data positif dan diprediksi positif maka akan dihitung sebagai *true positive* dan jika data positif diprediksi negatif maka akan dihitung sebagai *false negative*. Pada data negatif jika diprediksi negatif akan dihitung sebagai *true negative* dan jika diprediksi positif maka akan dihitung sebagai *false positive*.

Tabel 2.1. Matriks *confusion*

Aktual	Prediksi	
	Positif	Negatif
Positif	*True Positive (TP)*	*False Negative (FN)*
Negatif	*False Positive (FP)*	*True Negative (TN)*

Sejumlah ukuran kinerja klasifikasi dapat ditentukan berdasarkan Tabel 4.14 Matriks *confusion*. Pada penelitian ini, evaluasi kinerja klasifikasi yang digunakan adalah akurasi dan *area under curve* (AUC). Berikut rumus yang memengaruhi nilai akurasi dan AUC (Gorunescu [6]).

$$\begin{aligned} Sensitivitas/Recall/TP_{rate} &= \frac{TP}{TP+FN} \\ FP_{rate} &= \frac{FP}{FP+TN} \\ akurasi &= \frac{TP+TN}{TP+TN+FP+FN} \end{aligned} \tag{2.8}$$

dimana *recall* merupakan rasio prediksi benar positif dibandingkan dengan seluruh data yang aktual positif, FP_{rate} merupakan rasio prediksi salah positif dibandingkan dengan seluruh data yang aktual negatif, dan akurasi merupakan tingkat kedekatan antara hasil klasifikasi dengan aktual.

2.2.8.2 *Area Under Curve* (*AUC*)

AUC merupakan kriteria evaluasi yang menggunakan sensitivitas atau spesifisitas sebagai dasar pengukuran (He dan Ma [8]). Apabila terjadi kasus ketidakseimbangan data (*imbalance dataset*) maka dalam memilih model mana yang terbaik dapat dilakukan dengan menggunakan nilai AUC karena nilai akurasi hanya mempelajari data mayoritas saja sehingga hasil yang didapatkan mungkin saja terjadi bias atau *overfitting*. Berikut rumus AUC

$$AUC = \frac{1 + TP_{rate} - FP_{rate}}{2} \tag{2.9}$$

Menurut Bekkar *et al.* [3], nilai AUC pada umumnya berada pada $[0,5–1,0]$ dengan 5 kategori dan pada Tabel 2.2 ditunjukkan interval masing-masing kategori.

Tabel 2.2. Kategori nilai *AUC*

Nilai AUC	Deskripsi
0,9 - 1,0	*Excellent Classification*
0,8 - 0,9	*Good Classification*
0,7 - 0,8	*Fair Classification*
0,6 - 0,7	*Poor Clasification*
0,5 - 0,6	*Failure Clasification*

2.3 Kerangka Pemikiran

Berdasarkan tinjauan pustaka dan teori penunjang, dapat disusun kerangka pemikiran berikut. Permasalahan umum yang terjadi pada analisis sentimen yaitu adanya *imbalanced dataset*. *Imbalanced dataset* adalah data yang tidak seimbang dari segi jumlah tiap kelas individu. Metode *multinomial naïve Bayes* merupakan metode *supervised learning* yang menggunakan probabilitas dan lebih difokuskan untuk klasifikasi teks (Liu & Ozsu [12]). Metode *multinomial naïve Bayes* dapat diterapkan pada analisis sentimen pada ulasan aplikasi Peduli Lindungi di *Google Play Store*. Metode yang digunakan untuk mengatasi *imbalanced dataset* adalah metode *ADASYN*.

BAB III

METODE PENELITIAN

Metodologi penelitian dibagi menjadi dua bagian yaitu data penelitian dan langkah penelitian. Penelitian ini merupakan penelitian terapan yakni implementasi metode *ADASYN* dan *multinomial naïve Bayes* dalam klasifikasi sentimen pada ulasan aplikasi Peduli Lindungi.

3.1 Data Penelitian

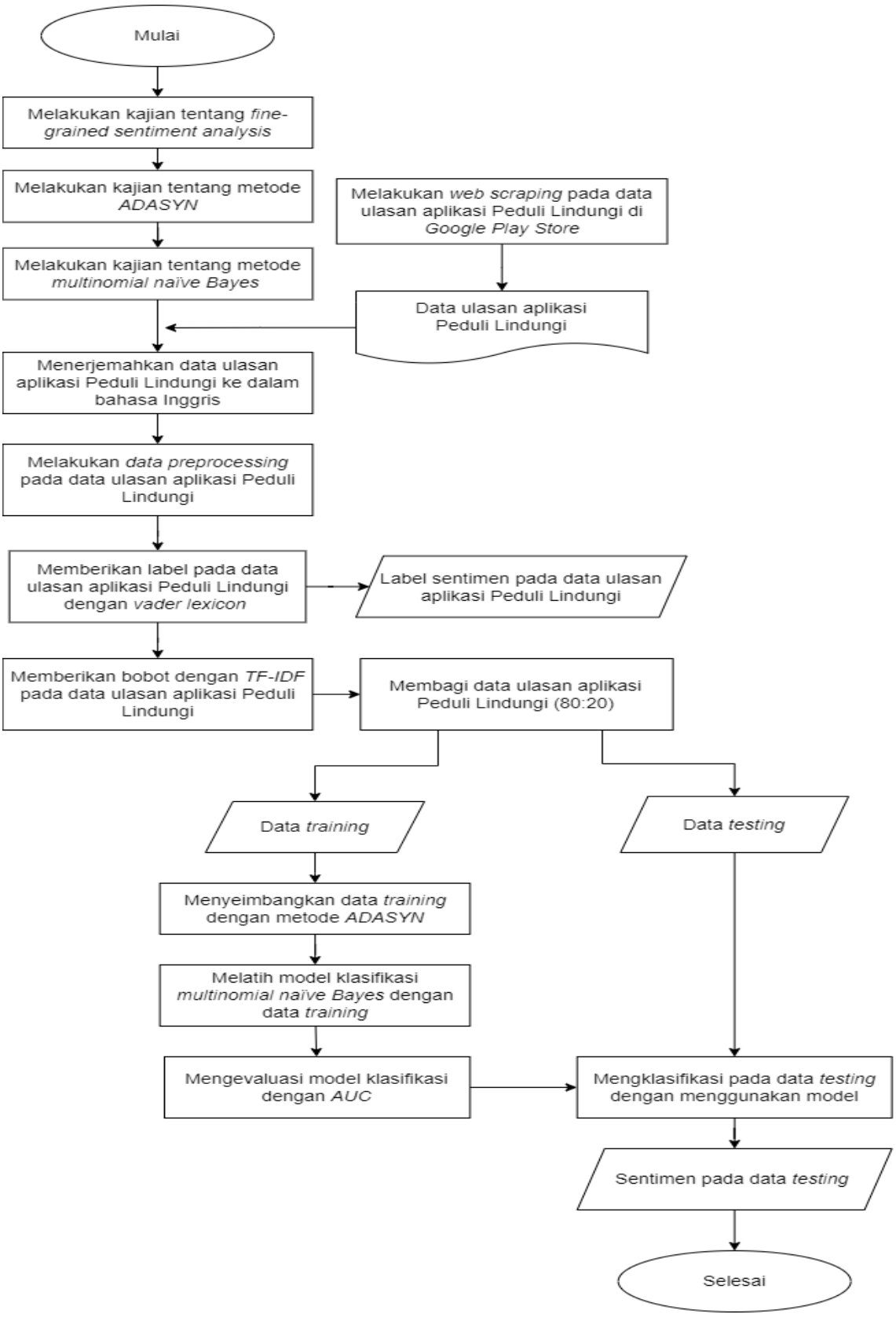
Data yang digunakan adalah data sekunder dari ulasan aplikasi Peduli Lindungi di *Google Play Store* (Peduli Lindungi [17]). Empat variabel yang digunakan dalam penelitian ini ditunjukkan pada Tabel 3.1.

Tabel 3.1. Variabel penelitian

Variabel	Data
Y_1	Nama *reviewer*
Y_2	*Rating*
Y_3	Tanggal ulasan
Y_4	Ulasan

3.2 Langkah Penelitian

Berikut merupakan langkah penelitian yang dilakukan untuk mencapai tujuan penelitian pada Gambar 3.1.



Gambar 3.1. *Flowchart* langkah penelitian

BAB IV

HASIL DAN PEMBAHASAN

Pada bab ini dibahas tentang menganalisis sentimen pada ulasan tentang aplikasi Peduli Lindungi dengan menggunakan metode *ADASYN* dan *multinomial naïve Bayes* meliputi deskripsi data, *data preprocessing*, pelabelan data, Pembobotan data dengan *TF-IDF*, penerapan metode *ADASYN*, penerapan metode *multinomial naïve Bayes*, dan pengujian hasil klasifikasi.

4.1 Deskripsi Data

Pada penelitian ini data yang digunakan merupakan data ulasan aplikasi Peduli Lindungi di *google play store* pada bulan Desember 2021 hingga April 2022. Total data yang diperoleh melalui proses *web scraping* adalah 9257 data. Variabel data yang digunakan adalah ulasan. Berikut salah-satu sampel data ulasan.

Tabel 4.1. Data ulasan aplikasi Peduli Lindungi

Ulasan
It's a bold move to make this app mandatory while it's far from user-friendly. The certificates won't appear although I have entered the right data (and to think about it, why can't they make it easier by only inputting the data once on the profile and it will integrate into the certificates and vaccines data). Not to mention the check-in issues. Please take everyone's review here as a constructive feedback and improve the app.

4.2 *Data Preprocessing*

Data ulasan yang telah diterjemahkan dalam bahasa Inggris, kemudian dilakukan *data preprocessing*. Proses tahapan *data preprocessing* dapat dilakukan sebagai berikut

1. *Case Folding*

Tabel 4.2. Hasil *case folding*

Sebelum	Sesudah
It's a bold move to make this app mandatory while it's far from user-friendly. The certificates won't appear although I have entered the right data (and to think about it, why can't they make it easier by only inputting the data once on the profile and it will integrate into the certificates and vaccines data). Not to mention the check-in issues. Please take everyone's review here as a constructive feedback and improve the app.	its a bold move to make this app mandatory while its far from userfriendly the certificates wont appear although i have entered the right data and to think about it why cant they make it easier by only inputting the data once on the profile and it will integrate into the certificates and vaccines data not to mention the checkin issues please take everyones review here as a constructive feedback and improve the app

2. *Tokenizing*

Tabel 4.3. Hasil *tokenizing*

Sebelum	Sesudah
its a bold move to make this app mandatory while its far from userfriendly the certificates wont appear although i have entered the right data and to think about	['its', 'a', 'bold', 'move', 'to', 'make', 'this', 'app', 'mandatory', 'while', 'its', 'far', 'from', 'userfriendly', 'the', 'certificates', 'wont', 'appear', 'although', 'i', 'have', 'entered', 'the', 'right', 'data', 'and', 'to', 'think', 'about',

Tabel 4.3 Lanjutan

Sebelum	Sesudah
it why cant they make it easier	'it', 'why', 'cant', 'they', 'make', 'it', 'easier',
by only inputting the data once	'by', 'only', 'inputting', 'the', 'data', 'once',
on the profile and it will integrate	'on', 'the', 'profile', 'and', 'it', 'will', 'integrate',
into the certificates and vaccines data	'into', 'the', 'certificates', 'and', 'vaccines', 'data',
not to mention the checkin issues	'not', 'to', 'mention', 'the', 'checkin', 'issues',
please take everyones review	'please', 'take', 'everyones', 'review',
here as a constructive feedback	'here', 'as', 'a', 'constructive', 'feedback',
and improve the app	'and', 'improve', 'the', 'app']

3. *Filtering*

Tabel 4.4. Hasil *filtering*

Sebelum	Sesudah
['its', 'a', 'bold', 'move', 'to', 'make', 'this',	['bold', 'move', 'make',
'app', 'mandatory', 'while', 'its', 'far', 'from',	'app', 'mandatory', 'far',
'userfriendly', 'the', 'certificates', 'wont',	'userfriendly', 'certificates', 'wont',
'appear', 'although', 'i', 'have', 'entered',	'appear', 'although', 'entered',
'the', 'right', 'data', 'and', 'to', 'think', 'about',	'right', 'data', 'think',
'it', 'why', 'cant', 'they', 'make', 'it', 'easier',	'cant', 'make', 'easier',
'by', 'only', 'inputting', 'the', 'data', 'once',	'inputting', 'data',
'on', 'the', 'profile', 'and', 'it', 'will', 'integrate',	'profile', 'integrate',
'into', 'the', 'certificates', 'and', 'vaccines', 'data',	'certificates', 'vaccines', 'data',
'not', 'to', 'mention', 'the', 'checkin', 'issues',	'mention', 'checkin', 'issues',
'please', 'take', 'everyones', 'review',	'please', 'take', 'everyones', 'review',
'here', 'as', 'a', 'constructive', 'feedback',	'constructive', 'feedback',
'and', 'improve', 'the', 'app']	'improve', 'app']

4. *Stemming*

Tabel 4.5. Hasil *stemming*

Sebelum	Sesudah
['bold', 'move', 'make', 'app', 'mandatory', 'far', 'userfriendly', 'certificates', 'wont', 'appear', 'although', 'entered', 'right', 'data', 'think', 'cant', 'make', 'easier', 'inputting', 'data', 'profile', 'integrate', 'certificates', 'vaccines', 'data', 'mention', 'checkin', 'issues', 'please', 'take', 'everyones', 'review', 'constructive', 'feedback', 'improve', 'app']	bold move make app mandatori far userfriendli certif wont appear although enter right data think cant make easier input data profil integr certif vaccin data mention checkin issu pleas take everyon review construct feedback improv app

4.3 Pelabelan Data

Data yang diperoleh setelah *data preprocessing* adalah 9172 data. Selanjutnya dilakukan proses pelabelan pada data ulasan dengan menggunakan *vader lexicon*. Jumlah data yang berlabel negatif sebanyak 5624 data dan positif sebanyak 3548 data. Berikut simulasi perhitungan nilai sentimen pada data ulasan.

Tabel 4.6. Simulasi perhitungan skor sentimen

Data Ulasan	Nilai	Label Sentimen
bold move make app mandatori far **userfriendli** certif **wont appear** although enter right data think **cant make easier** input data profil integr certif vaccin data mention checkin issu pleas take everyon review construct feedback **improv** app	$3 - 2 = 1$	Positif

4.4 Pembobotan data dengan *TF-IDF*

Pembobotan pada hasil *data preprocessing* dengan *TF-IDF* digunakan untuk mengetahui nilai frekuensi sebuah kata di dalam dokumen. Berikut hasil simulasi pembobotan lima sampel data ulasan.

Tabel 4.7. Hasil simulasi pembobotan *TF-IDF*

Kata	*tf*					*df*	*IDF*
	D1	D2	D3	D4	D5		
good	1	1	0	1	0	3	0,222
latest	0	0	1	0	0	1	0,699
android	0	0	1	0	0	1	0,699
version	0	0	1	0	0	1	0,699
still	0	0	1	0	0	1	0,699
cant	0	0	2	0	0	1	0,699
updat	0	0	2	1	0	2	0,398
care	0	0	1	0	0	1	0,699
protect	0	0	1	0	0	1	0,699
googl	0	0	1	0	0	1	0,699
play	0	0	2	0	0	1	0,699
stuck	0	0	1	0	0	1	0,699
percent	0	0	1	0	0	1	0,699
around	0	0	1	0	0	1	0,699
work	0	0	0	1	0	1	0,699
fine	0	0	0	1	0	1	0,699
data	0	0	0	0	1	1	0,699
secur	0	0	0	0	1	1	0,699
inform	0	0	0	0	1	1	0,699
guarante	0	0	0	0	1	1	0,699

4.5 Penerapan Metode *ADASYN*

Data ulasan dibagi menjadi dua bagian yaitu 80% atau 7337 data *training* dan 20% atau 1835 data *testing*. Data *training* yang digunakan tidak seimbang karena jumlah kelas negatif lebih banyak dibandingkan jumlah kelas positif, yaitu 4507 sentimen negatif dan 2830 sentimen positif. Oleh karena itu, dalam penelitian ini metode *ADASYN* digunakan untuk menangani data yang tidak seimbang. Setelah diterapkan metode *ADASYN*, jumlah sentimen pada data *training* yaitu 4507 negatif dan 4256 positif. Berikut penerapan metode *ADASYN* pada 10 sampel data ulasan.

Input

(1) $m = 10$ sampel data tersebut memiliki jumlah kelas positif $m_s = 3$ dan kelas negatif $m_l = 7$. Sehingga $m_s \leq m_l$ dan $m_s + m_l = m$.

Prosedur

(1) Menghitung tingkat ketidakseimbangan kelas dengan persamaan (2.1).

$$\begin{aligned} d &= \frac{3}{7} \\ &= 0,428 \end{aligned}$$

(2) Dengan $d_{th} = 0,75$ (Rahayu dkk., [19]), hasil kalkulasi ketidakseimbangan kelas di atas memenuhi kondisi $d < d_{th}$. Maka dilanjutkan ke prosedur penghitungan selanjutnya.

   (a) Menghitung jumlah sampel data sintetis yang perlu dihasilkan untuk kelas minoritas menggunakan persamaan (2.2) dengan $\beta = 0,9$ (Rahayu dkk., [19]), maka:

$$\begin{aligned} G &= (7 - 3) \times 0,9 \\ &= 3,6 \end{aligned}$$

   (b) Menghitung rasio r_i menggunakan persamaan (2.3), sebelumnya dilakukan kalkulasi Δ untuk setiap data pada kelas minoritas positif.

Evaluasi *nearest neighbor* pada data pertama ($D1$) dengan nilai $K = 5$ yang ditunjukkan pada Tabel 4.8.

Tabel 4.8. *Nearest neighbor* untuk Data 1 ($D1$)

Data	Kelas	Jarak
$D0$	Negatif	1,078
$D8$	Negatif	1,414
$D7$	Negatif	1,414
$D4$	Negatif	1,414
$D8$	Negatif	1,414

Tabel 4.8 menunjukkan terdapat 5 data dengan kelas selain kelas positif, sehingga $\Delta_{D1} = 5$. Maka, rasio untuk data $D1$ adalah

$$\begin{aligned} r_{D1} &= \frac{\Delta_{D1}}{K} \\ &= \frac{5}{5} \\ &= 1 \end{aligned}$$

Dengan cara yang sama maka diperoleh nilai Δ untuk setiap data pada Tabel 4.9.

Tabel 4.9. *Nearest neighbor* untuk setiap data pada kelas positif

Data Evaluasi	Data Terdekat	Δ_i	r_i
$D1$	$D0$, $D8$, $D7$, $D4$, $D2$	5	1
$D6$	$D4$, $D8$, $D7$, $D0$, $D5$	5	1
$D9$	$D7$, $D8$, $D4$, $D2$, $D6$	4	0,8

(c) Normalisasi r_i untuk mendapatkan distribusi kerapatan ($\hat{r}$) sehingga didapatkan hasil pada Tabel 4.10.

Tabel 4.10. Distribusi kerapatan untuk data pada kelas positif

Data Evaluasi	$\hat{r}_i$
$D1$	0,3571
$D6$	0,3571
$D9$	0,2857

(d) Menghitung jumlah sampel data sintesis untuk tiap data ke-i (x_i) dengan persamaan 2.4. Sebagai contoh data ke-1 maka,

$$\begin{aligned} g_1 &= \hat{r}_1 \times G \\ &= 0,3571 \times 3,6 \\ &= 1,286 \end{aligned}$$

Untuk semua data diperoleh g_i dan dilakukan pembulatan untuk mendapatkan jumlah duplikasi data sintetis seperti ditunjukkan pada Tabel 4.12.

Tabel 4.11. Jumlah duplikasi data sintetis

Data	$\hat{r}_i$	g_i	Sintetis
$D1$	0,3571	1,286	1
$D6$	0,3571	1,286	1
$D9$	0,2857	1,029	1

(e) untuk setiap sampel data kelas minoritas x_i, menghasilkan sampel data sintetis sebanyak g_i. Sehingga diperoleh 3 data sintetis dan nilai rasio ketidakseimbangan kelas di atas *threshold* yaitu 0,86.

4.6 Penerapan Metode *Multinomial Naïve Bayes*

Pengklasifikasian pada data *testing* ke dalam dua kelas yaitu sentimen positif dan negatif dengan menggunakan metode *multinomial naïve Bayes*. Berikut enam data yang akan digunakan sebagai simulasi dalam penerapan metode *multinomial naïve Bayes*.

Tabel 4.12. Data Ulasan

Data	Ulasan	Sentimen Label
training	thank admin final certif claim	1
	alway forc close open vaccin certif	0
	work fine updat good	1
	app crash tri open certif	0
testing	updat cant open vaccin certif	?

Sebelumnya dilakukan pembobotan pada data *training* yang ditunjukkan sebagai berikut.

Tabel 4.13. Pembobotan *TF-IDF* pada data *training*

Kata	*tf*				*df*	*IDF*
	D1	D2	D3	D4		
thank	1	0	0	0	1	0,602
admin	1	0	0	0	1	0,602
final	1	0	0	0	1	0,602
certif	1	1	0	1	3	0,125
claim	1	0	0	0	1	0,602
good	1	0	1	0	2	0,301
alway	0	1	0	0	1	0,602
forc	0	1	0	0	1	0,602
close	0	1	0	0	1	0,602
open	0	1	0	1	2	0,301
vaccin	0	1	0	0	1	0,602
work	0	0	1	0	1	0,602
fine	0	0	1	0	1	0,602
updat	0	0	1	0	1	0,602
app	0	0	0	1	1	0,602
crash	0	0	0	1	1	0,602
tri	0	0	0	1	1	0,602

Tahapan perhitungan pada proses klasifikasi adalah sebagai berikut.

(1) Menghitung probabilitas *prior* pada kelas positif dan negatif dengan persamaan (2.6).

$$P(positif) = \frac{N_{positif}}{N} \qquad P(negatif) = \frac{N_{negatif}}{N}$$
$$= \frac{2}{4} \qquad = \frac{2}{4}$$
$$= 0,5 \qquad = 0,5$$

(2) Menghitung probabilitas kata ke-n dengan persamaan (2.7).

$$P(updat|positif) = \frac{0,602+1}{4,941+9,780} \qquad P(updat|negatif) = \frac{0+1}{5,066+9,780}$$
$$= \frac{1,602}{14,721} \qquad = \frac{1}{14,846}$$
$$= 0,109 \qquad = 0,067$$
$$P(cant|positif) = \frac{0+1}{4,941+9,780} \qquad P(cant|negatif) = \frac{0+1}{5,066+9,780}$$
$$= \frac{1}{14,721} \qquad = \frac{1}{14,846}$$
$$= 0,068 \qquad = 0,067$$
$$P(open|positif) = \frac{0+1}{4,941+9,780} \qquad P(open|negatif) = \frac{0,602+1}{5,066+9,780}$$
$$= \frac{1}{14,721} \qquad = \frac{1,602}{14,846}$$
$$= 0,068 \qquad = 0,107$$
$$P(vaccin|positif) = \frac{0+1}{4,941+9,780} \qquad P(vaccin|negatif) = \frac{0,602+1}{5,066+9,780}$$
$$= \frac{1}{14,721} \qquad = \frac{1,602}{14,846}$$
$$= 0,068 \qquad = 0,107$$
$$P(certif|positif) = \frac{0,125+1}{4,941+9,780} \qquad P(certif|negatif) = \frac{0,250+1}{5,066+9,780}$$
$$= \frac{1,125}{14,721} \qquad = \frac{1,250}{14,846}$$
$$= 0,076 \qquad = 0,084$$

(3) Menghitung probabilitas dokumen menentukan kelas dengan persamaan 2.5.

$$\begin{aligned} P(positif|testing) &\propto P(positif) \times P(updat|positif) \times P(cant|positif) \times \\ &\quad P(open|positif) \times P(vaccin|positif) \times P(certif|positif) \\ &\propto 0,5 \times 0,109 \times 0,068 \times 0,068 \times 0,068 \times 0,076 \\ &\propto 0,130 \times 10^{-16} \\ P(negatif|testing) &\propto P(negatif) \times P(updat|negatif) \times P(cant|negatif) \times \\ &\quad P(open|negatif) \times P(vaccin|negatif) \times P(certif|negatif) \\ &\propto 0,5 \times 0,067 \times 0,067 \times 0,107 \times 0,107 \times 0,084 \\ &\propto 0,216 \times 10^{-16} \end{aligned}$$

(4) Berdasarkan nilai probabilitas paling tinggi yaitu $0,216 \times 10^{-16}$, maka kelas pada data *testing* yaitu negatif.

4.7 Pengujian Hasil Klasifikasi

Hasil klasifikasi data ulasan Peduli Lindungi yang telah diperoleh kemudian diuji keakuratannya menggunakan matriks *confusion* yang ditunjukkan sebagai berikut.

Tabel 4.14. Matriks *confusion* hasil klasifikasi

Aktual	Prediksi	
	Positif	Negatif
Positif	581	137
Negatif	121	996

Berdasarkan Tabel 4.14 diperoleh bahwa dari 1835 data yang diuji terdapat 1577 data yang diklasifikasikan secara benar. Dengan persamaan (2.8) diperoleh nilai akurasi 85,94%, sensitivitas 80,92% dan FP_{rate} 0,108.

Berdasarkan persamaan (2.9) diperoleh nilai AUC 0,851 dan nilai tersebut

termasuk dalam *good classification*. Dengan demikian, klasifikasi data ulasan aplikasi Peduli Lindungi di *google play store* menggunakan metode *ADASYN-multinomial naïve Bayes* menunjukkan akurasi 85,94% dan termasuk *good classification* dengan nilai *AUC* 0,851.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian dan pembahasan diperoleh kesimpulan untuk data *testing* dengan 1835 data ulasan diklasifikasikan dengan benar ada 1577 data yaitu 581 data untuk kelas sentimen positif dan 996 data untuk kelas sentimen negatif. Data yang diklasifikasikan salah yaitu sentimen positif yang diklasifikasikan sebagai sentimen negatif ada 137 data dan sentimen negatif yang yang diklasifikasikan sebagai sentimen positif ada 121 data. Klasifikasi data ulasan aplikasi Peduli Lindungi di *google play store* menggunakan metode *ADASYN-multinomial naïve Bayes* menunjukkan akurasi 85,94% dan termasuk *good classification* dengan nilai *AUC* 0,851.

5.2 Saran

Pada penelitian ini dibahas penerapan metode *ADASYN-multinomial naïve Bayes* pada klasifikasi data ulasan aplikasi Peduli Lindungi. Metode *ADASYN* digunakan untuk mengatasi data penelitian yang tidak seimbang. Selain dengan metode *ADASYN*, terdapat cara lain yang dapat digunakan yaitu menggunakan metode *modified ADASYN*. Bagi pembaca yang tertarik dapat melanjutkan penelitian ini dengan menggunakan metode *modified ADASYN* dalam mengatasi data yang tidak seimbang.

Daftar Rujukan

[1] Abbas, M., K.A. Memon, A.A. Jamali, S. Memon, and A. Ahmed, *Multinomial Naive Bayes Classification Model for Sentiment Analysis*. IJCSNS Int. J. Comput. Sci. Netw. Security, Vol. 19, No. 3, 2019.

[2] Agustina, N., D.H. Citra, W. Purnama, C. Nisa, dan A.R. Kurnia, *Implementasi Algoritme Naive Bayes untuk Analisis Sentimen Ulasan Shopee pada Google Play Store*. MALCOM: Indonesian Journal of Machine Learning and Computer Science, Vol. 2, No. 1, 2022.

[3] Bekkar, M., H.K. Djemma, and T.A. Alitouche, *Evaluation Measures for Models Assessment over Imbalanced Data Sets*. Journal of Information Engineering and Applications. Vol. 3, No. 10, 2013.

[4] Bramer, M., *Principles of Data Mining*. London: Springer, 2007.

[5] Elbagir, S. and J. Yang, *Twitter Sentiment Analysis Using Natural Language Toolkit and VADER sentiment*. In Proceedings of the international multiconference of engineers and computer scientists, Vol. 122, No. 16, 2019.

[6] Gorunescu, F., *Data Mining: Concepts, Models and Techniques*. Berlin: Springer-Verlag, 2011.

[7] Gunawan, F., M.A. Fauzi, dan P.P. Adikara, *Analisis Sentimen Pada Ulasan Aplikasi Mobile Menggunakan Naive Bayes dan Normalisasi Kata Berbasis Levenshtein Distance (Studi Kasus Aplikasi BCA Mobile)*. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN, 2548, 964X, 2017.

[8] He, H. and Y. Ma, *Imbalanced Learning: Foundations, Algorithms, and Applications*. ISBN: 9781118074626, 2013.

[9] He, H., Y. Bai, E.A. Garcia, and S. Li, *ADASYN: Adaptive synthetic Sampling Approach for Imbalanced Learning*. 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence). IEEE, 2008.

[10] Herdiana, D., *Aplikasi Peduli Lindungi: Perlindungan Masyarakat Dalam Mengakses Fasilitas Publik Di Masa Pemberlakuan Kebijakan PPKM*. Jurnal Inovasi Penelitian. Vol. 2, No. 6, 2021.

[11] Hutto, C. and E. Gilbert, *Vader: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text*. In Proceedings of the international AAAI conference on web and social media, Vol. 8, No. 1, 2014.

[12] Liu, L. and M.T. Ozsu, *Encyclopedia of Database Systems*. In Encyclopedia of Database Systems. Springer, 2009.

[13] Maarif, A.A., *Penerapan Algoritme TF-IDF untuk Pencarian Karya Ilmiah*. Jurnal Jurusan Teknik Informatika. Fakultas Ilmu Komputer. Universitas Dian Nuswantoro, 2015.

[14] Moraes, R., J.F. Valiati, and W.P. Gavião Neto, *Document-Level Sentiment Classification: An Empirical Comparison between SVM and ANN*. Expert Systems with Applications, Vol. 40, No. 2, 2013.

[15] Nurhidayati, N., S. Sugiyah, dan K. Yuliantari, *Pengaturan Perlindungan Data Pribadi Dalam Penggunaan Aplikasi Pedulilindungi*. Widya Cipta: Jurnal Sekretari Dan Manajemen, Vol. 5, No. 1, 2021.

[16] Nurrohmat, M.A. and S.N. Azhari, *Sentiment Analysis of Novel Review Using Long Short-Term Memory Method*. IJCCS (Indonesian Journal of Computing and Cybernetics Systems), Vol. 13, No. 3, 2019.

[17] Peduli Lindungi, [online], https://play.google.com/store/apps/details?id=com.telkom.tracencare&hl=en&gl=US, diakses tanggal 15 September 2022.

[18] Pintoko, B.M. dan K.L. Muslim, *Analisis Sentimen Jasa Transportasi Online pada Twitter Menggunakan Metode Naïve Bayes Classifier*. E-Proceeding of Engineering: Vol. 5, No. 3, 2018.

[19] Rahayu, S., T.B. Adji, dan N.A. Setiawan, *Penghitungan k-NN pada Adaptive Synthetic-Nominal (ADASYN-N) dan Adaptive Synthetic-kNN (ADASYN-KNN) untuk Data Nominal-Multi Kategori*. Jurnal Otomasi Kontrol Dan Instrumentasi, Vol. 9, No. 2, 2017.

[20] Sabily, A.F., P.P. Adikara, dan M.A. Fauzi, *Analisis Sentimen Pemilihan Presiden 2019 pada Twitter menggunakan Metode Maximum Entropy*. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN, 2019.

[21] Song, J., K.T. Kim, B. Lee, S. Kim, and H.Y. Youn, *A Novel Classification Approach based on Naïve Bayes for Twitter Sentiment Analysis*. KSII Transactions on Internet and Information Systems (TIIS), Vol. 11, No. 6, 2017.

[22] Turland, M., *php—architect's Guide to Web Scraping with PHP*, 2010.

Lampiran 1

Berikut program *python* yang digunakan untuk *scraping* data ulasan aplikasi Peduli Lindungi di *google play store*.

Menginstall package scraping
```
!pip install google-play-scraper
```

Mengimpor *library* yang dibutuhkan
```
from google_play_scraper import app, Sort, reviews_all
import pandas as pd
import numpy as np
```

Scraping ulasan pada aplikasi Peduli Lindungi
```
us_reviews = reviews_all(
    "com.telkom.tracencare",
    sleep_milliseconds = 0, # defaults to 0
    lang = 'en', # defaults to "en"
    country = 'id', # defaults to "us"
    sort = Sort.MOST_RELEVANT # defaults to Sort.MOST_RELEVA
NTS
)
```

Menampilkan data dalam bentuk tabel
```
df_review = pd.DataFrame(np.array(us_reviews),columns=['revi
ew'])
df_review = df_review.join(pd.DataFrame(df_review.pop('revie
w').tolist()))
df_review.head()
```

Menampilkan beberapa kolom yang dibutuhkan
```
my_df = df_review[['userName', 'score', 'at', 'content']]
my_df.columns = ["Nama reviewer", "Rating", "Tanggal ulasan"
, "Ulasan"]
my_df.head()
```

Memfilter data berdasarkan tanggal ulasan
```
data_review = my_df[(my_df["Tanggal ulasan"] >= '2021-12-
16') & (my_df["Tanggal ulasan"] <= '2022-04-16')]
data_review = data_review.reset_index()
data_review = data_review.drop('index', axis=1)
data_review.head()
```

Menampilkan jumlah data ulasan
```
len(data_review)
```

Menyimpan data dalam bentuk *csv* dan *excel*
```
data_review.to_csv('Scrapped_data.csv', index = False)
data_review.to_excel('Scrapped_data.xlsx', index = False)
```

Lampiran 2

Berikut program *python* yang digunakan untuk analisis sentimen pada data ulasan aplikasi Peduli Lindungi di *google play store*.

Menginstall *package* yang dibutuhkan
```
!pip3 uninstall googletrans
!pip3 install googletrans==3.1.0a0
```

Mengimpor *library* yang dibutuhkan
```
from googletrans import Translator
from sklearn.pipeline import Pipeline
import numpy as np
import pandas as pd
import re
import string

# Filtering
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords

# Stemming
from nltk.stem.porter import PorterStemmer
from sklearn.pipeline import Pipeline

# Labeling
from nltk.sentiment import SentimentIntensityAnalyzer
import nltk
nltk.download('vader_lexicon')

# TF-IDF
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer

# Splitting data
from sklearn.model_selection import train_test_split

# Model
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, precision_score,
 recall_score, f1_score
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import roc_auc_score, auc, roc_curve
```

Mengimpor dan menampilkan data dalam bentuk tabel
```
path = 'Scrapped_data.csv'
data = pd.read_csv(path)
data.head()
```

Menampilkan jumlah baris dan kolom data
```
data.shape
```

Menghilangkan variabel yang tidak dipakai
```
df_data = data.copy()
df_data = data.drop(columns = ['Nama reviewer','Rating','Tan
ggal ulasan'])
```

Menerjemahkan ke dalam bahasa Inggris
```
translator = Translator()
df_data['Ulasan'] = df_data['Ulasan'].apply(translator.trans
late, src='id', dest='en').apply(getattr, args=('text',))
df_data_en = df_data.copy()
df_data_en.to_csv('data_en.csv', index=False)
df_data_en.head()
```

 # Data Preprocessing

Case Folding
```
def case_folding(content):
  content = content.lower()
  content = content.strip(' ')
  content = re.sub(r'[?|$|.,|!_:)(-+)]', '', content)
  content = re.sub(r'\d', '', content)
  content = re.sub('[^\w\s]', '',content)
  content = re.sub('  ', ' ',content)
  return content
df_data_en['Ulasan'] = df_data_en['Ulasan'].apply(case_foldi
ng)
df_data_en.head()
```

Tokenizing
```
def token(content):
  nstr = content.split(' ')
  dat = []
  a = -1
  for hu in nstr:
    a = a+1
  if hu == '':
```

```
    dat.append(a)
  return nstr
df_data_en['Ulasan'] = df_data_en['Ulasan'].apply(token)
df_data_en.head()
## Filtering
def stopwords_removal(content):
  filtering = stopwords.words('english')
  x = []
  data = []
  def myFunc(x):
    if x in filtering:
      return False
    else:
      return True
  fit = filter(myFunc, content)
  for x in fit:
    data.append(x)
  return data
df_data_en['Ulasan'] = df_data_en['Ulasan'].apply(stopwords_
removal)
df_data_en.head()


## Stemming
def stemming(content):
  stemmer = nltk.porter.PorterStemmer()
  do = []
  for w in content:
    dt = stemmer.stem(w)
    do.append(dt)
  d_clean = []
  d_clean = " ".join(do)
  return d_clean
df_data_en['Ulasan'] = df_data_en['Ulasan'].apply(stemming)

df_data_en.to_csv('data_cleans.csv', index=False)
data_cleans = pd.read_csv('data_cleans.csv', encoding='latin
1')
data_cleans.head()


# Menampilkan hasil data preprocessing
data_clean = data_cleans.dropna()
data_clean.to_csv('data_clean.csv', index=False)
data_cleans = pd.read_csv('data_clean.csv', encoding='latin1
')
data_cleans.head()
```

Mengecek data *missing value*
```
data_cleans.info()
```

Melakukan pelabelan pada data ulasan aplikasi Peduli Lindungi
```
sia = SentimentIntensityAnalyzer()
data_cleans["Ulasan"][0:10].apply(lambda x: sia.polarity_sco
res(x))

data_cleans['Ulasan'][0:10].apply(lambda x: sia.polarity_sco
res(x)["compound"])

data_cleans["polarity_score"] = data_cleans['Ulasan'].apply(
lambda x: sia.polarity_scores(x)["compound"])
data_cleans.head()

data_cleans["Ulasan"][0:10].apply(lambda x: "pos" if sia.pol
arity_scores(x)["compound"] > 0 else "neg")

data_cleans["sentiment_label"] = data_cleans["Ulasan"].apply
(lambda x: 1 if sia.polarity_scores(x)["compound"] > 0 else
0)
data_cleans.head()
```

Menampilkan jumlah kelas sentimen pada data
```
data_cleans["sentiment_label"].value_counts()
```

Mengubah tipe data
```
data_cleans = data_cleans.astype({'sentiment_label': 'catego
ry'})
data_cleans = data_cleans.astype({'Ulasan': 'string'})
```

Melakukan pembobotan *TF-IDF*
```
tf = TfidfVectorizer()
text_tf = tf.fit_transform(data_cleans['Ulasan'].astype('U')
)
text_tf
```

Membagi data *training* sebanyak 80% dan data *testing* sebanyak 20%
```
x_train, x_test, y_train, y_test = train_test_split(text_tf,
 data_cleans['sentiment_label'], test_size=0.2, random_state
=42)
```

Menampilkan kelas sentimen pada data *training* dan *testing*

```
y_train.value_counts()
y_test.value_counts()
```

Proses ADASYN

```
data_ada = data_cleans.sample(10, random_state = 46).reset_i
ndex().drop("index", axis=1)
data_ada

data_ada['sentiment_label'].value_counts()

tf = TfidfVectorizer()
text_tf_ada = tf.fit_transform(data_ada['Ulasan'].astype('U'
))
text_tf_ada

from scipy.spatial.distance import euclidean
from sklearn.metrics.pairwise import euclidean_distances
import numpy as np
import matplotlib.pyplot as plt
from sklearn.neighbors import NearestNeighbors

k = 5

# Build K Nearest Neighbors model
knn_model = NearestNeighbors(n_neighbors=k, metric="euclidea
n").fit(text_tf_ada)
distances, indices = knn_model.kneighbors()


# Print the 'k' nearest neighbors
print("K Nearest Neighbors dg pusat D1:")
for rank, index in enumerate(indices[1][:k], start=1):
    print(str(rank) + " ==>", index)
print("K Nearest Neighbors dg pusat D6")
for rank, index in enumerate(indices[6][:k], start=1):
    print(str(rank) + " ==>", index)
print("K Nearest Neighbors dg pusat D9")
for rank, index in enumerate(indices[9][:k], start=1):
    print(str(rank) + " ==>", index)
from sklearn.metrics.pairwise import euclidean_distances

a = [0, 8, 7, 4, 2]
b = [4, 8, 7, 0, 5]
c = [7, 8, 4, 2, 6]
print("Jarak euclidean K Nearest Neighbors dg pusat D0:")
```

```
for i in a:
  print(str(i), "==>", euclidean_distances(text_tf_ada[0],te
xt_tf_ada[i]))
print("Jarak euclidean K Nearest Neighbors dg pusat D1:")
for i in b:
  print(str(i), "==>", euclidean_distances(text_tf_ada[6],te
xt_tf_ada[i]))
print("Jarak euclidean K Nearest Neighbors dg pusat D2:")
for i in c:
  print(str(i), "==>", euclidean_distances(text_tf_ada[9],te
xt_tf_ada[i]))
```

Algoritma *multinomial naïve Bayes*

Imbalanced data modelling

```
clf = MultinomialNB().fit(x_train,y_train)
predicted = clf.predict(x_test)
false_positive_rate, true_positive_rate, tresholds = roc_cur
ve(y_test,clf.predict(x_test))
print('MultinomialNB Accuracy: ', accuracy_score(y_test,pred
icted))
print('MultinomialNB Precision: ', precision_score(y_test,pr
edicted, average='binary', pos_label=0))
print('MultinomialNB Recall: ', recall_score(y_test,predicte
d, average='binary', pos_label=0))
print('MultinomialNB f1-
score: ', f1_score(y_test,predicted, average='binary', pos_l
abel=0))
print('MultinomialNB AUC: ', auc(false_positive_rate, true_p
ositive_rate))

print(f'confusion matrix:\n {confusion_matrix(y_test,predict
ed)}')
print('=====================================\n')
print(classification_report(y_test,predicted,zero_division=0
))
```

ADASYN modelling

```
from imblearn.over_sampling import ADASYN

x = x_train
y = y_train

print(x.shape)
print(y.shape)
print(y.value_counts())
```

```
ada = ADASYN(sampling_strategy= 'minority')
x_ada, y_ada =ada.fit_resample(x, y)

print(y_ada.value_counts())

from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, precision_score,
 recall_score, f1_score
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import roc_auc_score,auc,roc_curve

clf = MultinomialNB().fit(x_ada,y_ada)
predicted = clf.predict(x_test)
false_positive_rate, true_positive_rate, tresholds = roc_cur
ve(y_test,clf.predict(x_test))
print('MultinomialNB Accuracy: ', accuracy_score(y_test,pred
icted))
print('MultinomialNB Precision: ', precision_score(y_test,pr
edicted, average='binary', pos_label=0))
print('MultinomialNB Recall: ', recall_score(y_test,predicte
d, average='binary', pos_label=0))
print('MultinomialNB f1-
score: ', f1_score(y_test,predicted, average='binary', pos_l
abel=0))
print('MultinomialNB AUC: ', auc(false_positive_rate, true_p
ositive_rate))

print(f'confusion matrix:\n {confusion_matrix(y_test,predict
ed)}')
print('=====================================\n')
print(classification_report(y_test,predicted,zero_division=0
))
```

Menampilkan hasil matriks *confusion*

```
import matplotlib.pyplot as plt
import numpy
from sklearn import metrics

confusion_matrix = metrics.confusion_matrix(y_test,predicted
)

cm_display = metrics.ConfusionMatrixDisplay(confusion_matrix
 = confusion_matrix, display_labels = [False, True])

cm_display.plot()
```

```
plt.show()

# Menampilkan true_positives, false_positives, true_negatives, false_negatives
tn, fp, fn, tp = metrics.confusion_matrix(y_test, predicted)
.ravel()
print("True Negatives: ",tn)
print("False Positives: ",fp)
print("False Negatives: ",fn)
print("True Positives: ",tp)

# Menggabungkan data ulasan, label sentimen, dan hasil prediksi
#Write  predictions to file
x_test = pd.DataFrame(x_test).reset_index().drop('index', ax
is = 1)
test_y = pd.DataFrame(y_test).reset_index().drop('index', ax
is = 1)
yhat = pd.DataFrame(predicted)

# x_test.rename(columns= {0: "Test Data"}, inplace = True)
test_y.rename(columns= {0: 'Label Sentimen'}, inplace = True
)
yhat.rename(columns = {0: 'Hasil Prediksi'}, inplace = True)

new = pd.concat([x_test, test_y, yhat], axis = 1)
new.to_excel('Hasil_testing.xlsx', index = False)
```