**SKRIPSI**

**HALAMAN JUDUL**

**ANALISIS SENTIMEN TERHADAP KEBIJAKAN MASUK SEKOLAH JAM LIMA PAGI DI NTT MENGGUNAKAN ALGORITMA NAÏVE BAYES**



Oleh :

**Wilhelmina Sonya Hoar**

NIM : 19083000192

**PROGRAM S1 SISTEM INFORMASI**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS MERDEKA**

**MALANG**

**2023**

# 

# ANALISIS SENTIMEN TERHADAP KEBIJAKAN MASUK SEKOLAH JAM LIMA PAGI DI NTT MENGGUNAKAN ALGORITMA NAÏVE BAYES



**SKRIPSI**

Diajukan kepada

Program S1 Sistem Informasi Universitas Merdeka Malang

Untuk Memenuhi Salah Satu Persyaratan

Dalam Memperoleh Gelar Sarjana Komputer (S. Kom)

Oleh :

**Wilhelmina Sonya Hoar**

NIM : 19083000192

**PROGRAM S1 SISTEM INFORMASI**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS MERDEKA**

**MALANG**

**2023**

# LEMBAR PERSETUJUAN SKRIPSI

|  |  |  |
| --- | --- | --- |
| Nama | : | Wilhelmina Sonya Hoar |
| NIM | : | 19083000192 |
| Universitas | : | Universitas Merdeka Malang |
| Fakultas | : | Teknologi Informasi |
| Program Studi | : | S1 Sistem Informasi |
| Judul Skripsi | : | ANALISIS SENTIMEN TERHADAP KEBIJAKAN MASUK SEKOLAH JAM LIMA PAGI DI NTT MENGGUNAKAN ALGORITMA NAÏVE BAYES |

|  |  |
| --- | --- |
| Malang, 25 Juli 2023 | |
| DISETUJUI DAN DITERIMA | |
| Ketua Program Studi S1 Sistem Informasi | Dosen Pembimbing |
|  |  |
| Galandaru Swalaganata, S.Si., M.Si | Drs. Anis Zubair, M.Kom |
| Dekan Fakultas Teknologi Informasi | |
|  | |
| Dr. Mardiana Andarwati, M.Si | |

# ANALISIS SENTIMEN TERHADAP KEBIJAKAN MASUK SEKOLAH JAM LIMA PAGI DI NTT MENGGUNAKAN ALGORITMA NAÏVE BAYES

|  |
| --- |
| Dipersiapkan dan disusun oleh: |
| Wilhelmina Sonya Hoar |
| 19083000192 |
|  |
| Telah dipertahankan di depan Dewan Penguji |
| Pada Tanggal 25 Juli 2023 |

|  |  |
| --- | --- |
| Susunan Dewan Penguji | |
| Ketua Penguji | Sekretaris Penguji |
|  |  |
| (Rahmatina Hidayati, S.Kom., M.T.) | (Drs. Anis Zubair, M.Kom.) |
| Anggota Penguji | |
|  | |
| (Ronald David Marcus, S.Kom., M.Kom.) | |

Skripsi ini Telah Diterima Sebagai Salah Satu Persyaratan untuk Memperoleh Gelar Sarjana Komputer

|  |
| --- |
| Malang, 25 Juli 2023 |
| Dekan Fakultas Teknologi Informasi  Universitas Merdeka Malang |
|  |
| Dr. Mardiana Andarwati, M,Si |

# SURAT PERNYATAAN

|  |  |  |
| --- | --- | --- |
| Yang bertanda tangan dibawah ini: | | |
| Nama | : | Wilhelmina Sonya Hoar |
| NIM | : | 19083000192 |
| Program Studi | : | S1 Sistem Informasi |
| Bidang Kajian Skripsi | : | Text Mining |
| Judul Skripsi | : | ANALISIS SENTIMEN TERHADAP KEBIJAKAN MASUK SEKOLAH JAM LIMA PAGI DI NTT MENGGUNAKAN ALGORITMA NAÏVE BAYES |

|  |  |  |
| --- | --- | --- |
| Alamat Rumah Asal | : | Jl. R Suprapto, Fatukbot, Kec. Atambua Selatan, Kab. Belu, NTT |
| No. Telp/Email | : | 082238228188  [wihelminasonya26@gmail.com](mailto:wihelminasonya26@gmail.com) |

Dengan ini menyatakan bahwa saya benar-benar melakukan penelitian dan penulisan skripsi tersebut di atas benar-benar karya saya dan tidak melakukan plagiasi. Jika saya melakukan plagiasi maka saya bersedia untuk dicabut gelar dan akademik saya.

Demikian surat pernyataan ini dibuat untuk dipergunakan sebagaimana mestinya

|  |
| --- |
| Malang, 25 Juli 2023 |
| Peneliti |
|  |
| Wilhelmina Sonya Hoar |

# MOTTO

“Janganlah takut, sebab Aku menyertai engkau, janganlah bimbang, sebab Aku ini Allahmu; Aku akan meneguhkan, bahkan akan menolong engkau; Aku akan memegang engkau dengan tangan kanan-Ku yang membawa kemenangan”

(Yesaya 41:10)

“Serahkanlah kuatirmu kepada TUHAN, maka Ia akan memelihara engkau!

Tidak untuk selama-lamanya dibiarkan-Nya orang benar itu goyah ”

(Mazmur 55:23)

# LEMBAR PERSEMBAHAN

Puji Syukur penulis panjatkan kepada Tuhan Yesus Kristus, yang telah memberikan Kesehatan, Rahmat dan Berkat-Nya, sehingga penulis dapat menyelesaikan skripsi ini dengan baik dan lancar hingga selesai. Walaupun jauh dari kata sempurna, namun penulis bangga telah mencapai pada titik ini, yang akhirnya skripsi ini bisa selesai tepat waktu.

Dengan rasa syukur yang mendalam, dengan telah diselesaikannya skripsi ini penulis mempersembahkannya kepada :

* Bapak Nicolaus Nahak Seran dan (Alm) Ibu Sri Utami, terimakasih untuk semua pengorbanan yang telah diberikan, setiap doa, motivasi, semangat, dan harapan dan cinta kasih yang tiada terhingga yang kalian berikan kepada saya sampai saat ini.
* Saudara-saudara saya Kakak Rico dan Kakak Elen, terima kasih telah hadir untuk menjadi penyemangat dalam mengerjakan skripsi ini
* Keluarga Besar yang ada di Atambua dan di Malang, terima kasih untuk setiap doa, motivasi, nasehat dan semangatnya selama ini
* Untuk Ary Rambing yang selalu mendukung dan memberikan semangat kepada saya dalam menyelesaikan skripsi ini
* Teman- teman terdekat saya Mita, Dewi, Clara, Eka, Dian, dan Ulfa yang selalu memberikan semangat, motivasi, dan dukungan yang selalu membuat saya semangat untuk menyelesaikan skripsi ini
* Dosen Pembimbing saya Pak Anis Zubair yang sudah membimbing serta memberikan masukan dan saran selama ini, sehingga saya dapat menyelesaikan skripsi ini.
* Semua teman-teman Sistem Informasi 2019
* Kepada semua teman-teman, saudara yang tidak bisa saya sebutkan satu persatu, saya persembahkan skripsi ini untuk kalian semua

# KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa atas Berkat dan Rahmat-Nya penulis dapat menyelesaikan skripsi dengan judul: **ANALISIS SENTIMEN TERHADAP KEBIJAKAN MASUK SEKOLAH JAM LIMA PAGI DI NTT MENGGUNAKAN ALGORITMA NAÏVE BAYES**. Pada kesempatan ini penulis menyampaikan rasa terima kasih yang tidak terhingga kepada orang-orang yang berperan sehingga dapat terselesaikannya skripsi ini, antara lain :

1. Ibu Dr. Mardiana Andarwati, SE., M.Si selaku Dekan Fakultas Teknologi

Informasi Universitas Merdeka Malang.

1. Bapak Galandaru Swalaganata, S.Si., M.Si selaku Ketua Program S1 Sistem

Informasi Fakultas Teknologi Informasi Universitas Merdeka Malang.

1. Bapak Drs. Anis Zubair, M.Kom selaku dosen pembimbing yang telah

memberikan arahan dan motivasi dalam menyelesaikan skripsi ini.

1. Bapak dan Ibu Dosen Program S1 Sistem Informasi Fakultas Teknologi

Informasi Universitas Merdeka Malang.

1. Bapak dan Ibu Karyawan S1 Sistem Informasi Fakultas Teknologi Informasi

Universitas Merdeka Malang.

1. Terima kasih terlebih khusus kepada kedua Orang Tua Bapak Nicolaus Nahak Seran dan (Alm) Ibu Sri Utami, yang selalu memberikan kasih sayang dan dukungan doa selalu dalam menyusun skripsi.
2. Teman-teman mahasiswa Program S1 Sistem Informasi Fakultas Teknologi

Informasi Universitas Merdeka Malang.

1. Seluruh pihak yang berperan secara langsung dan tidak langsung yang tidak dapat disebutkan satu per satu.

Akhirnya, dengan segala kerendahan hati peneliti menyadari keterbatasan penelitian ini karena keterbatasan peneliti. Oleh karena itu peneliti mengharapkan kritik dan saran demi kesempurnaan penelitian ini. Semoga karya yang sederhana dapat memberi manfaat bagi semua pihak.

|  |
| --- |
| Malang, 17 Juli 2023 |
| Peneliti |
|  |
|  |
|  |
| Wilhelmina Sonya Hoar |

# DAFTAR ISI

[HALAMAN JUDUL i](#_Toc140568206)

[LEMBAR PERSETUJUAN ii](#_Toc140568207)

[LEMBAR PENGESAHAN Error! Bookmark not defined.](#_Toc140568208)

[LEMBAR PERNYATAAN ORISINALITAS iv](#_Toc140568209)

[MOTTO v](#_Toc140568210)

[LEMBAR PERSEMBAHAN vi](#_Toc140568211)

[KATA PENGANTAR vii](#_Toc140568212)

[DAFTAR ISI viii](#_Toc140568213)

[DAFTAR TABEL xi](#_Toc140568214)

[DAFTAR GAMBAR xii](#_Toc140568215)

[DAFTAR LAMPIRAN xiv](#_Toc140568216)

[ABSTRAK xv](#_Toc140568217)

[ABSTRACT xvi](#_Toc140568218)

[BAB I PENDAHULUAN 1](#_Toc140568219)

[1.1. Latar Belakang 1](#_Toc140568220)

[1.2. Rumusan Masalah 3](#_Toc140568221)

[1.3. Tujuan 4](#_Toc140568222)

[1.4. Manfaat Penelitian 4](#_Toc140568223)

[1.5. Batasan Penelitian 5](#_Toc140568224)

[BAB II KAJIAN PUSTAKA 6](#_Toc140568225)

[2.1 Landasan Teori 6](#_Toc140568226)

[2.1.1 Data *Mining* 6](#_Toc140568227)

[2.1.2 *Text Mining* 8](#_Toc140568228)

[2.1.3 *Machine Learning* 9](#_Toc140568229)

[2.1.4 Analisis Sentimen 10](#_Toc140568230)

[2.1.5 TF-IDF (*Term Frequency-Inverse Document Frequency*) 10](#_Toc140568231)

[2.1.6 Naïve Bayes 12](#_Toc140568232)

[2.1.7 Twitter 13](#_Toc140568233)

[2.1.8 Python 14](#_Toc140568234)

[2.1.9 *Jupyter Notebook* 15](#_Toc140568235)

[2.2 Penelitian Terdahulu 16](#_Toc140568236)

[BAB III METODE PENELITIAN 19](#_Toc140568237)

[3.1 Desain Penelitian 19](#_Toc140568238)

[3.1.1 Jenis Penelitian 19](#_Toc140568239)

[3.1.2 Tahapan Penelitian 20](#_Toc140568240)

[3.2 Populasi dan Sampel 21](#_Toc140568241)

[3.2.1 Populasi 21](#_Toc140568242)

[3.2.2 Sampel 22](#_Toc140568243)

[3.3 Teknik Pengumpulan Data 22](#_Toc140568244)

[3.4 Teknik Analisis Data 24](#_Toc140568245)

[3.4.1 Text Preprocessing 24](#_Toc140568246)

[3.4.2 Pembobotan Kata TF-IDF (Term Frequency-Inverse Document Frequency) 26](#_Toc140568247)

[3.4.3 Splitting Dataset 28](#_Toc140568248)

[3.4.4 Pemodelan Klasifikasi Naïve Bayes 29](#_Toc140568249)

[3.4.5 Evaluasi Model 30](#_Toc140568250)

[3.4.6 Visualisasi Model 32](#_Toc140568251)

[BAB IV HASIL DAN PEMBAHASAN 33](#_Toc140568252)

[4.1 Pengumpulan Data 33](#_Toc140568253)

[4.2 Text Preprocessing 35](#_Toc140568254)

[4.3 Pembobotan Kata TF-IDF (Term Frequency-Inverse Document Frequency) 42](#_Toc140568255)

[4.4 Splitting Dataset 46](#_Toc140568256)

[4.5 Pengujian Model Klasifikasi Naïve Bayes 47](#_Toc140568257)

[4.6 Evaluasi Model 51](#_Toc140568258)

[4.7 Hasil Klasifikasi Model 56](#_Toc140568259)

[4.8 Visualisasi Model 60](#_Toc140568260)

[BAB V PENUTUP 65](#_Toc140568261)

[4.2 Kesimpulan 65](#_Toc140568262)

[4.3 Saran 65](#_Toc140568263)

[DAFTAR PUSTAKA 67](#_Toc140568264)

[LAMPIRAN 69](#_Toc140568265)

# DAFTAR TABEL

Tabel 2. 1 Daftar Penelitian Terdahulu 16

Tabel 3. 1 Hasil Scraping Data 23

Tabel 3. 2 Contoh Penerapan Proses Cleaning 24

Tabel 3. 3 Contoh Penerapan Proses Case Folding 25

Tabel 3. 4 Contoh Penerapan Proses Tokenizing 25

Tabel 3. 5 Contoh Penerapan Proses Normalization 25

Tabel 3. 6 Contoh penerapan Proses Stopword Removal 26

Tabel 3. 7 Contoh Penerapan Stemming 26

Tabel 3. 8 Contoh Dokumen 27

Tabel 3. 9 Contoh Hasil Pembobotan Kata TF-IDF 27

Tabel 4. 1 Hasil Proses Cleaning Dan Case Folding 36

Tabel 4. 2 Hasil Proses Tokenizing 37

Tabel 4. 3 Hasil Proses Normalization 38

Tabel 4. 4 Hasil Proses Stopword Removal 40

Tabel 4. 5 Hasil Proses Stemming 41

Tabel 4. 6 Hasil Pembobotan TF-IDF 44

Tabel 4. 7 Data Sentimen 48

Tabel 4. 8 Frekuensi Kemunculan Kata Pada Data Training 49

Tabel 4. 9 Hasil Confusion Matrix 53

Tabel 4. 10 Laporan Hasil Evaluasi Model 54

Tabel 4. 11 Hasil Klasifikasi Dari Model 56

Tabel 4. 12 Hasil Word Cloud 63

# DAFTAR GAMBAR

Gambar 2. 1 Tahapan Knowledge Discovery Database (Han et al., 2012: 7) 6

Gambar 3. 1 Tahapan Penelitian 20

Gambar 3. 2 Confusion Matrix 30

Gambar 4. 1 Kode Program Langkah 1 Scraping 33

Gambar 4. 2 Kode Program Langkah 2 Scraping 34

Gambar 4. 3 Kode Program Langkah 3 Scraping 35

Gambar 4. 4 Kode Program Langkah 4 Scraping 34

Gambar 4. 5 Hasil Data Scraping File Excel 34

Gambar 4. 6 Kode Program Proses Cleaning dan Case Folding 36

Gambar 4. 7 Kode Program Proses Tokenizing 37

Gambar 4. 8 Kode Program Proses Normalization 37

Gambar 4. 9 Cuplikan Isi File Normalisasi.csv 38

Gambar 4. 10 Kode Program Import nltk 38

Gambar 4. 11 List Stopword Indonesia 39

Gambar 4. 12 Kode Program Proses Stopword Removal 39

Gambar 4. 13 Cuplikan Isi File Stopword.txt 40

Gambar 4. 14 Kode program Proses Stemming 41

Gambar 4. 15 Output Proses Stemming 41

Gambar 4. 16 Kode Program Menentukan Indeks Kata 42

Gambar 4. 17 Output Hasil Proses Term Frequency 43

Gambar 4. 18 Kode Program Menghitung Term Frequency 43

Gambar 4. 19 Output Menghitung Term Frequency 44

Gambar 4. 20 Kode Program MenghitungTF-IDF 44

Gambar 4. 21 Hasil TF-IDF 44

Gambar 4. 22 Kode Program Proses Splitting Dataset 46

Gambar 4. 23 Hasil Splitting Dataset 47

Gambar 4. 24 Kode Program Proses Klasifikasi Naïve Bayes 47

Gambar 4. 25 Kode Program Proses Evaluasi Model 51

Gambar 4. 26 Hasil Evaluasi Model 52

Gambar 4. 27 Kode Program Proses Visualisasi Confusion Matrix 52

Gambar 4. 28 Hasil Visualisasi Confusion Matrix 53

Gambar 4. 29 Kode Program Menghitung Jumlah Sentimen 61

Gambar 4. 30 Kode Program Visualisasi Pie Chart 61

Gambar 4. 31 Hasil Visualisasi Pie Chart 62

Gambar 4. 32 Kode Program Import Word Cloud 62

Gambar 4. 33 Kode Program Visualisasi Word Cloud 63

Gambar 4. 34 Hasil Visusalisasi Word Cloud 63

# DAFTAR LAMPIRAN

**Lampiran 1** Source Code Proses Scraping Data 69

**Lampiran 2** Source Code Proses Preprocessing, TF-IDF, Splitting Dataset, Klasifikasi Naïve Bayes, Evaluasi, Dan Visualisasi 71

# ABSTRAK

Pendidikan memiliki peran yang sangat penting dalam menciptakan generasi muda yang berkualitas dan memiliki kemampuan untuk memajukan bangsa. Dalam upaya meningkatkan kualitas pendidikan, pada tanggal 27 Februari 2023 pemerintah Nusa Tenggara Timur menerapkan kebijakan masuk sekolah jam lima pagi. Namun, kebijakan tersebut menuai pro dan kontra pada masyarakat. Masyarakat kini semakin aktif dalam mengungkapkan pendapat mereka melalui media sosial. Kritik terhadap kebijakan tersebut tercermin dalam komentar-komentar yang muncul di media sosial, khususnya di Twitter.

Sehingga, perlu dilakukan analisis sentimen untuk melihat berapakah respon positif dan negatif terhadap kebijakan tersebut. Peneliti melakukan analisis sentimen menggunakan algoritma Naïve Bayes terhadap hasil pencarian tweets dengan kata kunci “sekolah jam 5 di NTT” pada rentang waktu 27 Februari 2023-23 Maret 2023. Didapatkan sebanyak 777 tweet dengan jumlah sentimen positif 24 tweet dan sentimen negatif 753 tweet. Selanjutnya, data diolah dan dianalisis menggunakan algoritma Naïve Bayes, yaitu algoritma yang umum digunakan pada data mining karena perhitungannya sederhana namun memiliki tingkat akurasi tinggi. Dalam penelitian ini, hasil akurasi sebesar 97% dengan nilai presisi untuk sentimen negatif 98% dan presisi sentimen positif 50%, nilai recall untuk sentimen negatif 99% dan recall sentimen positif 25%, nilai f1-score untuk sentimen negatif 99% dan f1-score sentimen positif 33%.

Kata kunci: Analisis Sentimen, Kebijakan, Naïve Bayes, Twitter

# ABSTRACT

Education is very important to build a quality young generation that can advance the nation. To improve the quality of education, on February 27, 2023, the government of East Nusa Tenggara implemented a 5am school entry policy. However, this policy has caused pros and cons in the community. People are becoming more active expressing their opinions through social media, seen from the comments appearing on social media, especially Twitter, about the policy.

Therefore, it's important to conduct sentiment analysis to know how many positive and negative responses to this policy. Researchers conducted sentiment analysis using Naïve Bayes algorithm on search results of tweets with the keyword "sekolah jam 5 di NTT " on February 27, 2023 to March 23, 2023. Total 777 tweets were collected, 24 positive sentiment tweets and 753 negative sentiment tweets. The data was then processed and analyzed using the Naïve Bayes algorithm, famous for its simple yet highly accurate calculations in data mining. The accuracy rate is 97%, while precision value is 98% for negative sentiment and 50% for positive sentiment, recall value is 99% for negative sentiment and 25% for positive sentiment. The f1-score value is 99% for negative sentiment and 33% for positive sentiment.

Keywords: Sentiment Analysis, Policy, Naïve Bayes, Twitter

# BAB I

# PENDAHULUAN

## Latar Belakang

Pendidikan memiliki peran yang sangat penting dalam menciptakan generasi muda yang berkualitas dan memiliki kemampuan untuk memajukan bangsa. Oleh karena itu, pemerintah Indonesia mengeluarkan berbagai kebijakan untuk memperbaiki mutu pendidikan di Indonesia. Dalam upaya meningkatkan kualitas pendidikan, pada tanggal 27 Februari 2023 pemerintah Nusa Tenggara Timur (NTT) menerapkan kebijakan masuk sekolah jam lima pagi. Kebijakan ini dibuat dengan alasan agar siswa terbiasa menghabiskan waktu tidur dari pukul 22.00 WITA hingga bangun pada pukul 04.00 WITA, melatih kedisiplinan para siswa, dan bisa menciptakan generasi yang lebih unggul.

Namun, kebijakan masuk sekolah jam lima pagi di NTT menuai pendapat berbeda-beda di antara masyarakat baik yang pro maupun kontra. Beberapa masyarakat setuju dengan kebijakan ini, menganggap bahwa siswa harus terbiasa bangun pagi dan mempersiapkan diri untuk belajar. Namun, ada juga masyarakat yang menolak kebijakan ini, merasa bahwa ini akan memberikan dampak negatif bagi kesehatan dan konsentrasi siswa seperti kurang tidur menyebabkan sulit konsentrasi, kesehatan mental bisa terganggu karena bisa meningkatkan risiko depresi, dapat meningkatkan resiko kecelakaan saat berkendara karena masih mengantuk, jalan menuju sekolah yang terkadang masih berupa jalan berbatu dan berlumpur yang membahayakan bagi siswa yang harus berjalan kaki, dan kondisi cuaca di NTT yang kurang bersahabat, terutama saat musim hujan.

Masyarakat kini semakin aktif dalam mengungkapkan pendapat mereka melalui media sosial. Kritik terhadap kebijakan tersebut tercermin dalam komentar-komentar yang muncul di media sosial, khususnya di Twitter. Komentar-komentar tersebut mencakup pandangan positif dan negatif terhadap kebijakan tersebut.

Twitter merupakan salah satu *platform* media sosial yang sering digunakan oleh masyarakat untuk berbagi informasi dan pendapat mengenai suatu isu. Twitter memungkinkan orang di seluruh dunia untuk menjaga komunikasi dengan keluarga, teman,, dan orang yang mereka cintai menggunakan komputer atau ponsel mereka (Fikri et al., 2020). Twitter memberikan kesempatan kepada pengguna untuk mengirim dan membaca pesan dengan lebih dari 280 karakter, yang dikenal sebagai *tweet* (Aditya et al., 2019).

Twitter juga sering digunakan untuk mengukur sentimen atau pandangan publik terhadap suatu topik tertentu, termasuk kebijakan masuk sekolah jam lima pagi di NTT. Melalui komentar di Twitter, dapat dilakukan analisis sentimen yang berguna dalam mengetahui pendapat dan reaksi masyarakat terhadap kebijakan tersebut. Oleh karena itu, diperlukan analisis sentimen terhadap komentar-komentar tersebut untuk memperoleh gambaran tentang respon masyarakat terhadap kebijakan masuk sekolah jam lima pagi di NTT.

Analisis sentimen merupakan bagian dari *Natural Language Processing* (NLP) dan merupakan proses mengidentifikasi isi dari suatu kumpulan data yang diekspresikan dalam bentuk teks opini atau pandangan (perasaan) tentang subjek atau peristiwa yang terkandung di dalamnya, baik positif maupun negatif. atau netral (Fikri et al., 2020). Analisis sentimen adalah proses untuk mengidentifikasi dan mengekstraksi informasi berupa sentimen atau perasaan dari teks atau komentar. Analisis sentimen terhadap kebijakan masuk sekolah jam lima pagi di NTT dapat memberikan manfaat bagi pemerintah untuk mengetahui pandangan publik terhadap kebijakan tersebut dan dapat memberikan masukan bagi pemerintah untuk memperbaiki kebijakan yang sudah ada atau membuat kebijakan yang baru.

Terdapat beberapa cara yang dapat digunakan untuk melakukan analisis sentimen di media sosial, salah satunya adalah menggunakan metode Naive Bayes, yaitu algoritma yang umum digunakan untuk data mining karena mudah digunakan, cepat diproses dan mudah diimplementasikan dalam bentuk yang cukup sederhana, serta efektif (Kaparang et al., 2021). Metode Naïve Bayes telah banyak digunakan dalam berbagai penerapan analisis sentimen pada *platform* media sosial, termasuk pada Twitter.

Dalam penelitian oleh Duei Putri et al., (2022), menggunakan metode pengklasifikasi Naïve Bayes untuk menganalisis opini publik terhadap kegiatan Dewan Perwakilan Rakyat (DPR) yang diekspresikan melalui media sosial Twitter. Dalam penelitian ini, terdapat 1546 komentar dari Twitter yang digunakan. Hasil penelitian menggunakan algoritma Naïve Bayes menunjukkan tingkat akurasi sebesar 0,8 atau 80%. Ini berarti sistem dapat memprediksi dengan akurat 80% dari total data uji sebesar 20%.

Peneliti lainnya Ernamia & Herliana, (2022), menganalisis kuliah daring menggunakan algoritma Naïve Bayes, K-Nearest Neighbor, dan Decision Tree. Penelitian ini bertujuan untuk membandingkan tingkat akurasi dari ketiga algoritma tersebut. Total data yang digunakan sebanyak 500 tweet. Hasil penelitian menunjukkan bahwa algoritma Naïve Bayes mencapai tingkat akurasi sebesar 81,57%, algoritma K-Nearest Neighbor mencapai akurasi sebesar 62,10%, sedangkan akurasi dari Decision Tree adalah 51,89%. Temuan ini membuktikan bahwa algoritma Naïve Bayes memiliki tingkat akurasi yang lebih tinggi dibandingkan dengan kedua algoritma lainnya, dengan akurasi sebesar 81,57%.

Berdasarkan uraian di atas, penulis akan melakukan penelitian yang berjudul **“Analisis Sentimen** **Terhadap Kebijakan Masuk Sekolah Jam Lima Pagi Di NTT Menggunakan Algoritma Naïve Bayes”.** Penelitian ini akan melakukan analisis terhadap hasil pencarian *tweet* dengan kata kunci "sekolah jam 5 di NTT". Data yang diperoleh melalui proses *scraping* data akan diklasifikasikan menjadi dua kategori sentimen, yaitu positif dan negatif. Penelitian ini menggunakan metode Naïve Bayes dengan harapan mampu memberikan analisis data yang sesuai dengan kategori sentimen, serta mencapai akurasi yang optimal.

## Rumusan Masalah

Berdasarkan latar belakang yang ada, maka dapat dirumuskan permasalahan yaitu:

* Bagaimana tingkat akurasi yang didapatkan dari hasil klasifikasi sentimen terhadap kebijakan masuk sekolah jam lima pagi di NTT mengggunakan algoritma Naïve Bayes?
* Bagaimana hasil evaluasi model analisis sentimen menggunakan algoritma Naïve Bayes?
* Bagaimana hasil perbandingan sentimen positif dan negatif pada masyarakat terhadap kebijakan masuk sekolah jam lima pagi di NTT?

## Tujuan

Berdasarkan uraian latar belakang dan rumusan masalah yang telah dijelaskan sebelumnya, tujuan dari penelitian ini adalah:

* Mengetahui tingkat akurasi yang didapatkan dari hasil klasifikasi sentimen terhadap kebijakan masuk sekolah jam lima pagi di NTT mengggunakan algoritma Naïve Bayes.
* Mengetahui hasil evaluasi model analisis sentimen menggunakan algoritma Naïve Bayes.
* Mengetahui hasil perbandingan sentimen positif dan negatif pada masyarakat terhadap kebijakan masuk sekolah jam lima pagi di NTT.

## Manfaat Penelitian

* Bagi Universitas

1. Dapat menjadi karya ilmiah di bidang ilmu teknologi informasi yang berkaitan dengan analisis sentimen
2. Dapat dijadikan sebagai referensi bagi peneliti lain yang tertarik untuk meninjau masalah atau topik yang sama

* Bagi Penulis

1. Meningkatkan pemahaman tentang analisis sentimen.
2. Mengasah dan mengembangkan kemampuan dalam mengumpulkan data, menganalisis data, dan membuat kesimpulan berdasarkan hasil penelitian
3. Meningkatkan kemampuan analisis data, terutama menggunakan algoritma Naïve Bayes.

* Bagi Pemerintah

1. Hasil dari analisis sentimen dapat memberikan informasi yang berguna untuk memperbaiki atau mempertahankan kebijakan tersebut.
2. Hasil analisis sentimen juga dapat berperan sebagai sumber evaluasi untuk kebijakan serupa yang akan diimplementasikan di masa depan

## Batasan Penelitian

Terdapat batasan penelitian dalam melakukan penelitian antara lain:

1. Media sosial yang digunakan adalah Twitter.
2. Total data *tweets* yang digunakan adalah 777 data yang telah difilter dan diambil dari 27 Februari 2023 sampai dengan 23 Maret 2023.
3. *Keyword* yang digunakan adalah: “sekolah jam 5 di NTT”.
4. Data yang digunakan dalam penelitian ini merupakan kumpulan tweet yang menggunakan bahasa Indonesia.
5. Tidak melakukan perbandingan keakuratan algoritma, algoritma yang digunakan hanya Naïve Bayes.
6. Analisis sentimen hanya menampilkan nilai positif dan negatif.
7. Analisis sentimen yang dilakukan menggunakan bahasa pemrograman Python dan tools *Jupyter Notebook*

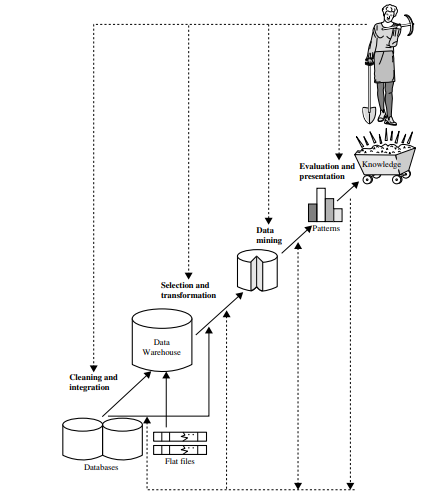
# BAB II

# KAJIAN PUSTAKA

## Landasan Teori

### Data *Mining*

Menurut Han et al., (2012: 6) “istilah lain untuk data mining adalah *Knowledge Discovery from Data* (KDD).” Data mining adalah suatu proses dimana pola-pola dan pengetahuan yang menarik ditemukan dari sejumlah besar data, seperti database, data warehouse, web, repositori informasi lainnya, atau data yang secara dinamis disimpan dalam sistem.



Gambar 2. 1 Tahapan Knowledge Discovery Database (Han et al., 2012: 7)

Berikut ini adalah proses *Knowledge Discovery Database,* sebagai berikut (Han et al., 2012: 6):

1. Data *cleaning* (membersihkan data): Proses untuk menghapus *noise* (gangguan) dan data yang tidak konsisten dari dataset. Contohnya data yang tidak relevan atau tidak akurat, seperti data yang hilang, duplikat, atau salah.
2. Data *integration* (integrasi data): Proses menggabungkan data dari berbagai sumber yang berbeda menjadi satu dataset yang terintegrasi. Sumber data yang berbeda dapat memiliki format yang berbeda, struktur yang berbeda, atau terminologi yang berbeda. Data integration bertujuan untuk menyatukan data yang relevan menjadi satu dataset yang konsisten untuk analisis lebih lanjut.
3. Data *selection* (seleksi data): Proses memilih data yang relevan dan diperlukan untuk tugas analisis tertentu dari dataset yang sudah terintegrasi. Pemilihan data dilakukan berdasarkan kriteria yang telah ditentukan sebelumnya, seperti atribut yang relevan, rentang waktu tertentu, atau kategori tertentu.
4. Data *transformation* (transformasi data): Proses untuk mengubah dan menggabungkan data ke dalam bentuk yang sesuai untuk analisis lebih lanjut. Transformasi data dapat melibatkan operasi seperti penggabungan data, penggalian data, atau penghitungan ringkasan data, sehingga data siap untuk diproses oleh metode analisis yang akan digunakan.
5. Data *mining* (penambangan data): Proses untuk menerapkan metode atau algoritma analisis data yang cerdas untuk mengidentifikasi pola atau tren yang tersembunyi dalam data. Data mining melibatkan penggunaan teknik-teknik seperti *clustering*, klasifikasi, asosiasi, atau prediksi untuk menemukan pola-pola yang berharga dalam data.
6. *Pattern evaluation* (evaluasi pola): Proses untuk mengidentifikasi pola-pola yang sangat menarik atau berharga dari hasil data *mining*. Pola-pola ini dievaluasi berdasarkan kriteria yang bermakna atau kepentingan yang telah ditetapkan sebelumnya, seperti ukuran pola, frekuensi, atau nilai prediktif.
7. *Knowledge presentation* (presentasi pengetahuan): Proses untuk menyajikan pola-pola yang telah dievaluasi atau pengetahuan yang ditemukan kepada pengguna atau *stakeholder*. Penyajian pengetahuan dapat menggunakan teknik visualisasi, representasi grafis, atau laporan yang mudah dimengerti agar dapat dipahami dan digunakan oleh pengguna untuk pengambilan keputusan atau pengelolaan bisnis.

### *Text Mining*

Menurut Han et al., (2012: 596) *text mining* adalah suatu bidang studi yang mencakup pencarian informasi, data mining, machine learning, statistik, dan linguistik komputasi untuk mendapatkan informasi berkualitas tinggi dari teks. Dalam text mining, fokus utamanya adalah pada teks yang mengandung sejumlah besar informasi, seperti artikel berita, makalah teknis, buku, perpustakaan digital, pesan email, blog, dan halaman web.

Berikut ini adalah beberapa contoh penerapan *text mining* yang paling umum, sebagai berikut (Han et al., 2012: 597):

1. *Text Categorization*, yaitu mengelompokkan teks ke dalam kategori atau topik tertentu berdasarkan karakteristik atau konten teks. Contohnya, mengkategorikan teks berita ke dalam kategori berdasarkan topik seperti politik, bisnis, olahraga, atau hiburan.
2. *Text Clustering*, yaitu mengelompokkan teks berdasarkan kesamaan karakteristik atau pola dalam konten teks. Misalnya, mengelompokkan ulasan produk berdasarkan tema atau mengelompokkan berita berdasarkan subjek tertentu. Tujuannya adalah untuk mengidentifikasi kelompok teks yang serupa untuk analisis lebih lanjut.
3. *Sentiment Analysis*, yaitu menganalisis sentimen atau pendapat dalam teks, seperti pendapat positif, negatif, atau netral terhadap suatu topik, entitas, atau konten teks lainnya. Analisis sentimen dapat digunakan untuk memahami sentimen pengguna, konsumen, atau pengguna lainnya terhadap produk, layanan, atau merek, dan dapat digunakan dalam pengambilan keputusan bisnis, perencanaan pemasaran, atau manajemen reputasi.
4. *Document Summarization*, yaitu merangkum dokumen atau teks panjang menjadi ringkasan yang lebih ringkas dan informatif. Peringkasan dokumen dapat membantu dalam mengidentifikasi informasi penting atau pola-pola yang signifikan dalam teks yang kompleks, serta memudahkan pemahaman dan pengambilan keputusan dalam menghadapi jumlah data yang besar.

### *Machine Learning*

Menurut Han et al., (2012: 24) *machine learning* untuk mempelajari cara komputer dapat belajar atau meningkatkan kinerjanya dari data. Program machine learning bertujuan agar komputer dapat belajar secara otomatis untuk mengenali pola yang kompleks dan membuat keputusan cerdas berdasarkan data.

Berikut ini adalah kategori *machine learning*, sebagai berikut (Han et al., 2012: 24):

1. *Supervised learning,* merujuk pada metode klasifikasi. Supervisi dalam proses pembelajaran berasal dari contoh-contoh yang diberikan dalam data latihan yang memiliki label. Dalam *supervised learning*, setiap contoh dalam data latihan memiliki label yang menunjukkan kelas atau keluaran yang seharusnya dihasilkan oleh model.
2. *Unsupervised learning*, adalah metode dalam machine learning di mana model belajar dari data latihan yang tidak memiliki label kelas atau keluaran yang diinginkan. Dalam *unsupervised learning*, tidak ada informasi tentang kelas atau keluaran yang harus diprediksi, sehingga proses pembelajaran dilakukan secara mandiri oleh model.
3. *Semi-supervised learning,* adalah kelas teknik dalam *machine learning* yang menggunakan baik contoh yang diberi label (*labeled examples*) maupun contoh yang tidak diberi label (*unlabeled examples*) dalam proses pembelajaran model. Pendekatan ini memanfaatkan informasi yang ada pada data *labeled* untuk mempelajari model kelas, dan data *unlabeled* untuk menyempurnakan batasan antara kelas-kelas tersebut.
4. *Active learning* merupakan suatu metode dalam bidang machine learning yang mengizinkan pengguna, seperti seorang pakar di bidangnya, untuk berpartisipasi aktif dalam proses pembelajaran. Dalam pendekatan *active learning*, pengguna dapat diminta untuk memberi label pada contoh-contoh data, yang bisa berasal dari kumpulan data *unlabeled* atau dapat disintesis oleh program pembelajaran itu sendiri. Tujuannya adalah untuk meningkatkan kualitas model dengan cara aktif memperoleh pengetahuan dari pengguna, dengan membatasi jumlah contoh yang dapat mereka minta untuk diberi label.

### Analisis Sentimen

Menurut Fink et al., (2011) definisi sentimen sebagai sikap positif atau negatif yang dimiliki oleh individu atau kelompok orang terhadap suatu objek. Objek dalam hal ini dapat berupa entitas (orang, kelompok, organisasi, atau lokasi geografis), peristiwa, tindakan yang melibatkan entitas, dan ide atau konsep. Menurut pengertian ini, sentimen memiliki polaritas atau nilai (positif atau negatif), sumber (orang atau kelompok yang memilikinya), dan target (objek yang menjadi sasaran sentimen). Jadi, tujuan analisis sentimen adalah mendeteksi keberadaan sentimen dalam sebuah teks dan mengidentifikasi nilai, sumber, dan target dari sentimen tersebut.

### TF-IDF (*Term Frequency-Inverse Document Frequency*)

Menurut Fikri et al., (2020) TF-IDF (*Term Frequency-Inverse Document Frequency*) adalah metode yang diterapkan untuk memberikan bobot kata dalam teks. Berikut adalah langkah-langkah dalam menghitung bobot kata menggunakan metode TF-IDF:

1. *Term Frequency* (TF)

TF digunakan untuk menghitung seberapa frekuensi kemunculan suatu kata dalam sebuah dokumen tertentu. TF dinyatakan sebagai rasio antara frekuensi sebuah kata (*term*) di dalam sebuah dokumen dengan jumlah total kata dalam dokumen tersebut. Nilai TF dapat dihitung menggunakan rumus berikut:

|  |  |  |
| --- | --- | --- |
| TF(d,t) = | Frekuensi kemunculan *term* t dalam dokumen d | (2. 1) |
| total *term* dalam dokumen d |

Keterangan:

|  |  |  |
| --- | --- | --- |
| t | = | kata |
| d | = | dokumen |

Hasil dari perhitungan ini akan menghasilkan nilai TF yang berada dalam rentang antara 0 dan 1, yaitu nilai yang lebih tinggi menunjukkan bahwa kata tersebut muncul lebih sering dalam dokumen tersebut.

1. *Inverse Document Frequency* (IDF)

IDF digunakan mengukur seberapa umum suatu kata dalam seluruh korpus dokumen. Menurut Aggarwal, (2015: 432), konsep *inverse document frequency* mengaplikasikan prinsip dengan lebih fleksibel, dimana kata-kata yang sering muncul diberi bobot yang lebih rendah. Dengan kata lain, semakin sering sebuah kata muncul, semakin sedikit bobot yang diberikan kepadanya. IDF dapat dihitung dengan persamaan di bawah ini:

|  |  |  |
| --- | --- | --- |
| IDF(t) = log | total dokumen | (2. 2) |
| *document frequency* *term* t |

Keterangan:

|  |  |
| --- | --- |
| t = | kata |

Hasil dari perhitungan ini akan menghasilkan nilai IDF yang berada dalam rentang lebih besar dari 0, di mana angka yang lebih tinggi menunjukkan bahwa kata tersebut jarang muncul dalam seluruh korpus dokumen.

1. TF-IDF

Setelah menghitung TF dan IDF, bobot kata (TF-IDF) dapat dihitung dengan mengalikan nilai TF dan nilai IDF. TF-IDF dapat dihitung dengan persamaan di bawah ini:

|  |  |  |
| --- | --- | --- |
| TF-IDF(d,t) = | TF(d,t) × IDF(t) | (2. 3) |

Keterangan:

|  |  |
| --- | --- |
| t = | kata |
| d = | dokumen |

Berdasarkan hasil perhitungan ini akan menghasilkan nilai bobot kata yang digunakan untuk mewakili kemunculan kata dalam suatu dokumen tertentu dalam konteks korpus dokumen yang lebih besar.

### Naïve Bayes

Menurut Bustami (2013) algoritma Naïve Bayes merupakan sebuah metode klasifikasi yang menggunakan probabilitas dan statistik, dan awalnya dikembangkan oleh seorang ilmuwan Inggris bernama Thomas Bayes. Naïve Bayes menggunakan pengalaman masa sebelumnya untuk memprediksi probabilitas di masa depan, dan sering disebut sebagai Teorema Bayes. Pada algoritma ini, Teorema Bayes digunakan bersama dengan asumsi Naive, yaitu menganggap bahwa atribut-atribut antar kelas adalah independen. Oleh karena itu, dalam Naïve Bayes, diasumsikan bahwa keberadaan atau ketiadaan suatu fitur dalam suatu kelas tidak terkait dengan fitur-fitur kelas lainnya. Teorema Bayes dapat diungkapkan dalam bentuk persamaan sebagai berikut:

|  |  |  |
| --- | --- | --- |
| P(H | X) = | P(X | H).P(H) | (2. 4) |
| P(X) |

Keterangan:

|  |  |  |
| --- | --- | --- |
| X | = | Data yang memiliki kelas yang belum diketahui. |
| H | = | Hipotesis data X merupakan sebuah kelas yang spesifik |
| P(H | X) | = | Probabilitas hipotesis H berdasar kondisi X  (*posteriori probability*) |
| P(H) | = | Probabilitas hipotesis H (*prior probability*) |
| P(X | H) | = | Probabilitas X mengacu pada kondisi hipotesis H |
| P(X) | = | Probabilitas X |

### Twitter

Twitter adalah sebuah *platform* media sosial yang memungkinkan pengguna untuk membuat dan melihat pesan-pesan singkat yang disebut *tweet*. Setiap tweet memiliki batas karakter maksimum 280 karakter dan dapat mencakup teks, gambar, atau video. Twitter menjadi salah satu *platform* media sosial yang sangat diminati di berbagai penjuru dunia dan digunakan dengan luas oleh individu, organisasi, dan perusahaan untuk berkomunikasi dengan audiens mereka, membagikan informasi, mempromosikan merek atau produk, dan membangun jejaring sosial. Menurut artikel yang diterbitkan Kominfo (dalam Kaparang et al., 2021), berdasarkan data dari PT Bakrie Telecom menyatakan bahwa di Indonesia, terdapat sekitar 19,5 juta pengguna Twitter, yang merupakan bagian dari jumlah total 500 juta pengguna di seluruh dunia. Twitter juga digunakan sebagai sarana untuk mengikuti dan memperoleh berita terkini, mengekspresikan pendapat dan gagasan, serta terhubung dengan orang-orang dengan minat yang sama.

Dalam keterangan yang disampaikan oleh (Twitter, 2023) berikut ini adalah pengertian beberapa istilah yang sering digunakan:

1. *Tweet*: Pesan singkat yang terdiri dari maksimum 280 karakter yang dapat dikirimkan atau dibagikan melalui akun Twitter.
2. *Retweet*: Tindakan mengirimkan kembali (re-share) tweet dari akun Twitter orang lain ke akun Twitter Anda sendiri.
3. *Mention*: Tindakan men-tag (menandai) akun Twitter orang lain dalam sebuah tweet, biasanya digunakan untuk memperluas jangkauan tweet atau memulai percakapan.
4. *Hashtag*: Tanda pagar (#) yang digunakan untuk menandai topik atau topik yang sama dalam sebuah tweet agar lebih mudah ditemukan oleh pengguna Twitter yang mencari topik tersebut.
5. *Direct Message* (DM): Pesan pribadi atau langsung yang dikirimkan ke akun Twitter seseorang tanpa melalui umpan publik.
6. *Follower*: Pengguna Twitter yang memilih untuk mengikuti akun Twitter Anda sehingga mereka dapat melihat tweet Anda di umpan mereka.
7. *Trending topic*: Topik atau topik yang sedang populer atau banyak dibahas di Twitter pada suatu waktu tertentu.
8. *Twitter handle*: Nama pengguna Twitter yang unik yang dimiliki setiap pengguna, diawali dengan tanda '@' dan digunakan untuk mencari atau menyebutkan akun Twitter orang lain.
9. *Twitter chat*: Diskusi atau percakapan online dengan pengguna Twitter lainnya, biasanya diatur pada waktu tertentu dengan menggunakan hashtag tertentu.
10. *Verified account*: Akun Twitter yang telah mendapatkan verifikasi oleh Twitter sebagai akun resmi dan asli untuk individu atau organisasi yang dikenal atau memiliki kepentingan yang signifikan.

### Python

Python merupakan salah satu bahasa pemrograman yang cukup terkenal dan banyak digunakan dalam bidang teknologi informasi seperti pengembangan perangkat lunak, analisis data, pengolahan teks, dan kecerdasan buatan. Terdapat berbagai *library* dan *framework* dalam bahasa pemrograman python yang dapat digunakan untuk membantu proses analisis data (Kirana et al., 2019). Dalam penelitian ini, digunakan beberapa *library* berikut:

1. Pandas

Pandas adalah *library* analisis data yang sering digunakan dalam Python untuk manipulasi dan analisis data. Pandas dapat digunakan untuk membaca, mengelola, dan menyimpan data dalam format tabel, seperti data hasil penelitian analisis sentimen.

1. NLTK (*Natural Language Toolkit*)

NLTK adalah *library* yang umum digunakan untuk pemrosesan bahasa alami (NLP) dalam Python. NLTK menyediakan berbagai alat untuk mengelola, membersihkan, dan mengolah teks, termasuk fitur seperti tokenisasi, stemming, dan klasifikasi teks yang dapat digunakan dalam analisis sentimen.

1. Sastrawi

Sastrawi adalah *library* pemrosesan bahasa alami (NLP) dalam bahasa Indonesia. *Library* ini menyediakan berbagai alat untuk mengelola, membersihkan, dan mengolah teks dalam bahasa Indonesia, termasuk fitur seperti stemming, stopword removal, dan tokenisasi. Sastrawi dapat digunakan dalam penelitian atau proyek yang memerlukan pemrosesan bahasa alami dalam bahasa Indonesia, termasuk dalam analisis sentimen terhadap teks berbahasa Indonesia.

1. *TextBlob*

*TextBlob* adalah *library* NLP yang menyediakan berbagai alat untuk analisis sentimen, termasuk analisis polaritas dan subjektivitas. *TextBlob* juga menyediakan antarmuka yang mudah digunakan untuk melakukan analisis sentimen pada teks dalam bahasa Inggris.

1. *Scikit-learn*

*Scikit-learn* merupakan *library machine learning* populer untuk Python yang menyediakan berbagai algoritma klasifikasi, termasuk algoritma Naïve Bayes. *Scikit-learn* menyediakan implementasi yang efisien dan mudah digunakan dari algoritma Naïve Bayes yang dapat digunakan dalam penelitian analisis sentimen.

### *Jupyter Notebook*

Aplikasi Notebook Jupyter, yang dulunya dikenal sebagai Notebook Ipython, dikembangkan oleh Jupyter Perez dan Granger sebagai sebuah aplikasi berbasis web browser yang populer digunakan dalam penelitian yang sudah bisa mendukung banyak bahasa pemrograman lainnya seperti R, Julia, dan bahasa-bahasa pemrograman lainnya (Duei Putri et al., 2022).

*Jupyter Notebook* memungkinkan pengguna untuk membuat dokumen berbasis web yang menggabungkan kode pemrograman, teks naratif, persamaan matematika, visualisasi data, dan hasil eksekusi kode yang dapat dijalankan secara interaktif. Dokumen yang dihasilkan dalam *Jupyter Notebook* dapat disimpan dalam format .ipynb (IPython Notebook), yang memungkinkan untuk berbagi dan mengulang eksperimen atau analisis data yang telah dilakukan

## Penelitian Terdahulu

Penulis menggunakan penelitian terdahulu sebagai sumber referensi dan inspirasi untuk menambah wawasan dan membantu penulis menyelesaikan penelitian tugas akhir ini. Berikut ini tabel daftar penelitian terdahulu yang didapatkan dan ditemukan penulis dapat dilihat pada tabel 2.1

Tabel 2. 1 Daftar Penelitian Terdahulu

| **Nama Peneliti** | **Judul Penelitian** | **Hasil Penelitian** |
| --- | --- | --- |
| Kaparang et al., (2021) | Analisis Sentimen New Normal Pada Masa Covid-19 Menggunakan Algoritma Naive Bayes Classifier | Penelitian ini bertujuan untuk memodelkan analisis sentimen opini publik terhadap kebijakan new normal pemerintah di Indonesia selama pandemi Covid-19. Metode yang dipergunakan adalah algoritma Naïve Bayes. Hasil dari pengklasifikasian menggunakan Naïve Bayes mencakup akurasi sebesar 80,37%, presisi sebesar 87,38%, *recall* sebesar 82,57%, dan f-*measure* sebesar 84,91%. Dalam penelitian ini, terdapat 5194 tweet yang diklasifikasikan sebagai positif dan 2908 tweet yang diklasifikasikan sebagai negatif. ni mengindikasikan bahwa perasaan positif lebih mendominasi daripada perasaan negatif. |
| Aditya et al., (2019) | Analisis Sentimen Pada Media Sosial Twitter Menggunakan Metode Naïve Bayes Classifier (Studi Kasus: Komentar Publik Kepada Tri Indonesia) | Tujuan penelitian ini adalah mengevaluasi sentimen terhadap akun resmi Tri Indonesia. Data yang digunakan berasal dari tweet orang-orang di media sosial Twitter dan dianalisis menggunakan metode Naïve Bayes Classifier. Dalam penelitian ini, terdapat 237 data komentar yang dibedakan ke dalam dua kategori, antara lain positif serta negatif. Pengujian menggunakan Naïve Bayes Classifier menghasilkan akurasi sebesar 84,28% dalam analisis sentimen pada media sosial Twitter. Hasil klasifikasi untuk class precision prediksi positif menunjukkan tingkat keakuratan sebesar 66,67%, sementara prediksi negatif mencapai tingkat keakuratan sebesar 91,67%. Untuk *recall*, *true* positif menghasilkan tingkat sebesar 76,67%, sedangkan *true* negatif sebesar 87,01%. |
| Duei Putri et al., (2022) | Analisis Sentimen Kinerja Dewan Perwakilan Rakyat (DPR) Pada Twitter Menggunakan Metode Naive Bayes Classifier | Penelitian ini bertujuan untuk menganalisis sikap masyarakat pada kegiatan Dewan Perwakilan Rakyat (DPR) yang diekspresikan melalui Twitter. Terdapat 1546 data yang dihasilkan dari crawling data di twitter pada penelitian ini, kemudian data dianalisis dengan algoritma naïve bayes. Hasil penelitian ini menunjukkan bahwa DPR memperoleh 95 tweet positif dengan polaritas 0,75 atau 75% sentimen positif, 693 tweet netral dengan polaritas 0,79 atau 79% netral, dan 758 tweet negatif dengan polaritas 0,82 atau 82% dengan nilai akurasi 0,8 atau 80%. |
| Gulo, (2021) | Analisis Sentimen Kuliah Online Selama Pandemi Covid-19 Menggunakan Algoritma Naive Bayes | Tujuan dari penelitian ini adalah untuk mengidentifikasi sejauh mana tingkat keakuratan analisis opini menggunakan metode Naïve Bayes, untuk mengetahui hasil analisis opini kuliah online dengan metode Naive Bayes pada masa pandemi Covid-19, dan untuk mengetahui dampak tentang pelaksanaan pembelajaran jarak jauh di masa Covid19. Jumlah data tweet yang dikumpulkan dari hasil crawling di Twitter adalah 2000 tweet. Hasil dari penelitian ini adalah skor akurasi 79,75%, skor presisi 79%, skor recall 80%, dan F1-Score 79%. |

# BAB III

# METODE PENELITIAN

## Desain Penelitian

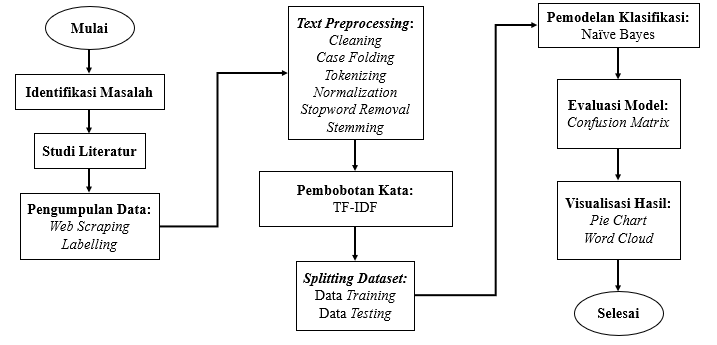
### Jenis Penelitian

Penelitian ini menerapkan metode kuantitatif deskriptif. Menurut Abdullah (2015: 90), jenis penelitian yang bertujuan untuk memberikan deskripsi atau penjelasan tentang berbagai situasi, fenomena, atau variabel penelitian berdasarkan data yang dikumpulkan dari kejadian yang terjadi, di foto, diwawancarai, diamati, serta bahan dokumen yang diteliti. Sehingga, dalam penelitian ini menggunakan pendekatan kuantitatif deskriptif untuk mengumpulkan dan menganalisis data dalam menggambarkan sentimen atau pandangan masyarakat terhadap kebijakan masuk sekolah jam lima pagi di Nusa Tenggara Timur (NTT).

Teknik pengambilan data online dengan menggunakan teknik *web scraping* untuk pengambilan data dari Twitter yang berkaitan dengan topik penelitian, yaitu kebijakan masuk sekolah jam lima pagi di NTT. Hasil *web scraping* menggunakan kata kunci "sekolah jam 5 di NTT" pada Twitter dan pada rentang waktu 27 Februari 2023 sampai dengan 23 Maret 2023.

Setelah data diambil, data kemudian akan diolah dan dianalisis menggunakan algoritma Naïve Bayes untuk melakukan analisis sentimen terhadap tweet yang berkaitan dengan topik penelitian. Dengan menggunakan desain penelitian kuantitatif deskriptif, teknik *web scraping*, dan analisis data menggunakan algoritma Naïve Bayes, penelitian "Analisis Sentimen Terhadap Kebijakan Masuk Sekolah Jam Lima Pagi Di NTT Menggunakan Algoritma Naïve Bayes" dapat memberikan gambaran yang jelas dan terukur tentang sentimen pengguna Twitter terhadap kebijakan tersebut.

### Tahapan Penelitian



Gambar 3. 1 Tahapan Penelitian

Gambar 3.1 menunjukkan tahapan penelitian yang dilakukan. Langkah awal dalam penelitian ini adalah mengidentifikasi permasalahan. Tujuan mengidentifikasi masalah adalah menemukan cara untuk mengkategorikan dan mengklasifikasikan pendapat pengguna Twitter mengenai kebijakan pengaturan jam masuk sekolah pukul lima pagi di Nusa Tenggara Timur (NTT). Setelah mengidentifikasi masalah, langkah berikutnya adalah studi literatur. Studi literatur yang relevan digunakan untuk mengumpulkan informasi terkait proses pemodelan analisis sentimen masyarakat. Metode klasifikasi Naïve Bayes digunakan dalam penelitian ini. Langkah selanjutnya adalah pengumpulan opini pengguna Twitter dengan menggunakan teknik *web scraping* untuk mengambil *tweet-tweet* dari *platform* Twitter, lalu memberi label positif (1) dan negatif (-1). Setelah mengumpulkan data, langkah berikutnya adalah melakukan *preprocessing* atau pengolahan awal data yang melibatkan beberapa proses seperti *Cleaning, Case Folding, Tokenizing, Normalization, Stopword Removal, dan Stemming* sehingga menghasilkan data bersih. Setelah data yang didapat bersih lalu masuk ke tahap pembobotan kata atau TF-IDF (*Term Frequency-Inverse Document Frequency*) yaitu tahapan seleksi data yang sering muncul. Setelah melakukan pembobotan TF-IDF, dilakukan *splitting* data untuk membagi data ke dalam dua bagian, yaitu 80% data latihan untuk melatih model dan 20% data pengujian untuk menguji performa model. Setelah data dibagi menjadi dua, data diolah menggunakan model klasifikasi Naive Bayes. Setelah menyelesaikan langkah klasifikasi data, langkah berikutnya adalah melakukan evaluasi pada model klasifikasi menggunakan *Confusion Matrix*. *Confusion matrix* berisi informasi yang membandingkan hasil kategorisasi yang dihasilkan oleh sistem dengan hasil yang diharapkan. Dengan menggunakan *Confusion Matrix*, dapat menghitung berbagai matriks evaluasi model seperti Akurasi, Presisi, *Recall*, Spesifisitas, dan F1-*Score* dari model klasifikasi. Hal ini memungkinkan untuk mengetahui sejauh mana tingkat akurasi dari model klasifikasi yang digunakan.. Tahap akhir adalah visualisasi hasil menggunakan *pie chart* untuk menampilkan proporsi atau persentase dari setiap kategori sentimen (positif dan negatif) dan *word cloud* adalah metode visualisasi yang digunakan untuk menggambarkan kata-kata yang paling sering muncul dalam data dengan menggunakan ukuran teks yang berbeda-beda.

## Populasi dan Sampel

### Populasi

Menurut Abdullah (2015: 226), populasi mengacu pada keseluruhan unsur yang menunjukkan karakteristik tertentu dan dapat digunakan untuk menarik kesimpulan, dimana keseluruhan unsur menunjukkan jumlah, sedangkan karakteristik tertentu menunjukkan ciri-ciri dari populasi tersebut.

Populasi pada penelitian adalah seluruh pengguna Twitter yang menuliskan *tweet* berbahasa Indonesia tentang kebijakan masuk sekolah jam lima pagi di wilayah Nusa Tenggara Timur (NTT). *Tweet* yang diambil mengandung kata kunci "sekolah jam 5 di NTT" pada rentang waktu tertentu. Populasi ini tidak terbatas pada satu lokasi geografis tertentu dan dapat mencakup pengguna Twitter dari berbagai wilayah di Indonesia yang membahas topik yang sama yaitu tentang kebijakan masuk sekolah jam lima pagi di NTT.

### Sampel

Menurut Abdullah (2015: 228), sampel yang ideal merupakan sampel yang mampu menggambarkan karakteristik populasi secara tepat. Pada penelitian ini menerapkan pendekatan *sampling* kuota yaitu salah satu teknik dari *Non Probability Sampling*. Menurut Hardani (2020: 369) “Ciri utama dari sampel kuota adalah anggota sampel dari tingkat tertentu dipilih berdasarkan jumlah (kuota) tertentu dengan karakteristik tertentu.”

Pengambilan sampel pada penelitian ini tidak dilakukan secara acak, melainkan didasarkan pada keyword tertentu dan jangka waktu tertentu yang telah ditentukan. Sampel pada penelitian ini adalah 777 data *tweet* yang berasal dari hasil *web scraping* data menggunakan kata kunci "sekolah jam 5 di NTT" pada Twitter. Data tersebut telah difilter dan diambil pada rentang waktu 27 Februari 2023 sampai dengan 23 Maret 2023.

## Teknik Pengumpulan Data

Data yang diterapkan dalam penelitian ini merupakan data primer. Data primer merupakan data yang diperoleh secara langsung oleh peneliti dari sumber asli yang menjadi fokus penelitian. Data yang diambil dalam penelitian yaitu hasil *web scraping* di Twitter. *Web scraping* adalah proses ekstraksi data yang lebih spesifik dari halaman web. Scraping mengidentifikasi dan mengekstraksi data yang relevan dari kode HTML halaman *web*, atau dari berkas-berkas yang ada di dalamnya, seperti teks, gambar, tabel, atau elemen lainnya. *Scraping* biasanya dilakukan dengan menggunakan perangkat lunak atau *library* tertentu yang memungkinkan peneliti untuk mengekstraksi data yang diinginkan dari halaman *web* secara otomatis. Peneliti melakukan proses *web scraping* data dari Twitter untuk mendapatkan *tweet-tweet* terkait kebijakan masuk sekolah jam lima pagi di NTT.

Proses web scraping pada penelitian ini yaitu mengimport tautan <https://github.com/JustAnotherArchivist/snscrape.git> ke dalam kode Python yang merupakan URL dari sebuah repositori Git yang dihosting di platform GitHub. Repository ini mengandung kode sumber dari proyek *snscrape* yang dikembangkan oleh pengguna GitHub bernama "JustAnotherArchivist". *Snscrape* adalah suatu perangkat lunak (*library*) Python yang digunakan untuk melakukan pengambilan (*scraping*) data dari media sosial, khususnya dari platform Twitter.

Selanjutnya, teks atau kalimat tweet tersebut diberi label dengan mengimport *library TextBlob*  untuk menghitung nilai polarity berdasarkan perbandingan antara jumlah kata-kata positif dan negatif dalam teks. Nilai polarity dapat berkisar dari -1 (sentimen sangat negatif) hingga 1 (sentimen sangat positif), dengan 0 menunjukkan sentimen netral. Pada penelitian ini hanya menganalisis sentimen positif dan negatif.

Pengambilan data dilakukan melalui media sosial Twitter menggunakan metode *web scraping* hanya menggunakan *tweet* yang berbahasa indonesia dengan kata kunci yang digunakan yaitu “sekolah jam 5 di NTT”. Maksimal 800 sampel digunakan secara keseluruhan dalam penelitian ini dalam rentang waktu 27 Februari 2023 sampai dengan 23 Maret 2023. Sehingga, data yang didapatkan berjumlah 777 data tweet. Setelah itu, data tweet disimpan dalam bentuk Microsoft Excel dengan format “csv” dan “xlsx”.

Berikut ini adalah tabel yang berisi lima *tweet* dari 777 data *tweet* yang didapatkan pada *scraping* data:

Tabel 3. 1 Hasil Scraping Data

| **Tweet** | **Label** |
| --- | --- |
| Heran Bukan Main Siswa di NTT Masuk Sekolah Jam 5 Pagi, Anggota DPR: Anak-Anak Jangan Dijadikan Kelinci Percobaan! https://t.co/PGn707FPyh | 1 |
| Ketika kita nonton, teman² di ntt jam 5 pagi malah sekolah hahaha | 1 |
| Cara Masuk Sekolah Jam 5 Pagi di NTT, Naik Kuda sampai Menginap di Sekolah Biar Tak Telat https://t.co/GjuPO78vH5 #TempoTekno | -1 |
| Seriusan pelajar di NTT disuruh masuk jam 5 pagi??! Kebijakan yang pastinya bikin kita bertanya-tanya Bro &amp; Sis, kok bisa anak sekolah memulai kegiatan belajar sepagi itu?  @psi\_id @psi\_jawatengah @kokokdirgantoro @FurqanAMC @yulizuardirais @elchanatmadja @MikhaRDK @bayu\_romawan https://t.co/ZdTEX69nwQ | -1 |
| Di NTT masuk sekolah jam 5 pagi, kalo di Jekarda gimana tuh ya? Berhasil kayanya ngatasin macet tp ortu2 pasti teriak krn lebih rempong dari pagi-pagi buta 😅 https://t.co/1K99KjAHHG | -1 |

## Teknik Analisis Data

Dalam penelitian ini ada beberapa langkah yang dipergunakan untuk menganalisis data, yaitu:

### *Text Preprocessing*

Metode pembersihan dan persiapan teks untuk diklasifikasi dikenal sebagai text preprocessing. Teks online cenderung memiliki banyak noise dan komponen yang tidak berguna misalnya tag HTML, script, iklan serta banyak kata dalam teks yang tidak terlalu berpengaruh terhadap makna teks secara keseluruhan (Haddi et al., 2013). *Text preprocessing* penting dilakukan karena semakin bersih data yang dihasilkan maka dapat meningkatkan efisiensi dan akurasi analisis sentimen

*Text preprocessing* meliputi *Cleaning, Case Folding, Tokenizing, Normalization, Stopword Removal, dan Stemming.*

* + - 1. *Cleaning*

*Cleaning* yaitu proses pembersihan teks dari karakter atau informasi yang tidak penting, misalnya tanda baca, karakter khusus, atau URL yang tidak relevan. Tujuannya adalah untuk membersihkan teks agar hanya mengandung informasi yang relevan untuk analisis selanjutnya. Berikut ini adalah contoh tahap *cleaning:*

Tabel 3. 2 Contoh Penerapan Proses Cleaning

| **Sebelum Cleaning** | **Setelah Cleaning** |
| --- | --- |
| Saya sngt senang dengan produk ini! :) #produkbagus | Saya sngt senang dengan produk ini produkbagus |

* + - 1. *Case Folding*

*Case Folding* merupakan proses mengkonversi semua karakter dalam teks menjadi huruf kecil untuk menghindari perbedaan dalam penulisan huruf besar/kecil yang bisa mempengaruhi konsistensi dan akurasi analisis teks.

Tabel 3. 3 Contoh Penerapan Proses Case Folding

| **Sebelum Case Folding** | **Setelah Case Folding** |
| --- | --- |
| Saya sngt senang dengan produk ini ProdukBagus | saya sngt senang dengan produk ini produkbagus |

* + - 1. *Tokenizing*

*Tokenizing* adalah proses memisahkan teks menjadi unit-unit terkecil yang disebut token. Token bisa berupa kata, frasa, atau kalimat dan biasanya dipecah berdasarkan spasi, tanda baca, atau aturan gramatikal tertentu.

Tabel 3. 4 Contoh Penerapan Proses Tokenizing

| **Sebelum Tokenizing** | **Setelah Tokenizing** |
| --- | --- |
| saya sngt senang dengan produk ini produkbagus | ['saya', 'sngt', 'senang', 'dengan', 'produk', 'ini', 'produkbagus'] |

* + - 1. *Normalization*

*Normalization* adalah tahap mengubah token dalam teks menjadi bentuk yang standar atau normal, misalnya mengganti singkatan menjadi bentuk lengkap atau menggantikan variasi ejaan menjadi bentuk yang konsisten untuk analisis selanjutnya.

Tabel 3. 5 Contoh Penerapan Proses Normalization

| **Sebelum Normalization** | **Setelah Normalization** |
| --- | --- |
| ['saya', 'sngt', 'senang', 'dengan', 'produk', 'ini', 'produkbagus'] | ['saya', 'sangat', 'senang', 'dengan', 'produk', 'ini', 'produkbagus'] |

* + - 1. *Stopword Removal*

*Stopword Removal* yaitu proses menghapus kata-kata yang dianggap tidak signifikan atau tidak relevan pada analisis teks, misalnya kata penghubung atau kata depan. Tujuannya adalah untuk mengurangi *noise* atau gangguan dalam teks dan memfokuskan pada kata-kata yang lebih penting dalam analisis.

Tabel 3. 6 Contoh penerapan Proses Stopword Removal

| **Sebelum Stopword Removal** | **Setelah Stopword Removal** |
| --- | --- |
| ['saya', 'sangat', 'senang', 'dengan', 'produk', 'ini', 'produkbagus'] | ['sangat', 'senang', 'produk', 'ini', 'produkbagus'] |

* + - 1. *Stemming*

*Stemming* merupakan sebuah proses perubahan token dalam teks menjadi bentuk yang dasar, dengan menghapus imbuhan atau akhiran dari kata. Tujuannya adalah untuk mengurangi variasi kata yang sama menjadi bentuk yang lebih sederhana atau standar, sehingga dapat digunakan untuk analisis kata-kata yang sama tetapi dalam bentuk yang berbeda.

Tabel 3. 7 Contoh Penerapan Stemming

| **Sebelum Stemming** | **Setelah Stemming** |
| --- | --- |
| ['sangat', 'senang', 'produk', 'ini', 'produkbagus'] | ['sangat', 'senang', 'produk', 'produkbagus'] |

### Pembobotan Kata TF-IDF (*Term Frequency-Inverse Document Frequency*)

TF-IDF (*Term Frequency-Inverse Document Frequency*) digunakan sebagai metode untuk memberikan bobot pada kata-kata dalam pemrosesan teks dengan tujuan mengukur tingkat kepentingan kata tersebut dalam sebuah dokumen, dalam kumpulan dokumen. Metode ini mengkombinasikan dua ide, yaitu frekuensi kata atau *Term Frequency* (TF), yang menghitung jumlah kemunculan kata dalam sebuah dokumen, dan *Inverse Document Frequency* (IDF), yang mengukur seberapa tidak umum atau jarang kata tersebut muncul dalam seluruh koleksi dokumen. Berikut ini adalah contoh data yang akan dihitung dengan TF-IDF sebagai berikut:

Tabel 3. 8 Contoh Dokumen

| **ID Dokumen** | **Isi Dokumen** |
| --- | --- |
| d1 | suka bahasa pemrograman python |
| d2 | ada lomba buat aplikasi |
| d3 | lomba pakai bahasa pemrograman python |

Pada tabel Tabel 3. 8 Contoh Dokumen menggunakan contoh sebanyak tiga dokumen yang akan di proses. Langkah awal adalah membuat daftar *term* dari dokumen d1, d2, dan d3. Apabila ada *term* yang berulang maka cukup ditulis satu kali

Selanjutnya, menghitung bobot kata TF-IDF pada setiap data yang bisa dilihat pada tabel *Tabel 3. 9* Contoh Hasil Pembobotan Kata TF-IDF

Tabel 3. 9 Contoh Hasil Pembobotan Kata TF-IDF

| ***TERM*** | **TF** | | | **TF**  **Normalisasi** | | | **DF** | **IDF** | **TF-IDF** | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **d1** | **d2** | **d3** | **d1** | **d2** | **d3** | **d1** | **d2** | **d3** |
| suka | 1 | 0 | 0 | 0,25 | 0 | 0 | 1 | 0,477 | 0,119 | 0 | 0 |
| bahasa | 1 | 0 | 1 | 0,25 | 0 | 0,2 | 2 | 0,176 | 0,044 | 0 | 0,035 |
| pemrograman | 1 | 0 | 1 | 0,25 | 0 | 0,2 | 2 | 0,176 | 0,044 | 0 | 0,035 |
| python | 1 | 0 | 1 | 0,25 | 0 | 0,2 | 2 | 0,176 | 0,044 | 0 | 0,035 |
| ada | 0 | 1 | 0 | 0 | 0,25 | 0 | 1 | 0,477 | 0 | 0,119 | 0 |
| lomba | 0 | 1 | 0 | 0 | 0,25 | 0 | 1 | 0,477 | 0 | 0,119 | 0 |
| buat | 0 | 1 | 0 | 0 | 0,25 | 0 | 1 | 0,477 | 0 | 0,119 | 0 |
| aplikasi | 0 | 1 | 0 | 0 | 0,25 | 0 | 1 | 0,477 | 0 | 0,119 | 0 |
| lomba | 0 | 0 | 1 | 0 | 0 | 0,2 | 1 | 0,477 | 0 | 0 | 0,095 |
| pakai | 0 | 0 | 1 | 0 | 0 | 0,2 | 1 | 0,477 | 0 | 0 | 0,095 |
| **Panjang**  **Dokumen** | **4** | **4** | **5** |  |  |  |  |  |  |  |  |

Pada perhitungan TF (*Term Frequency),* setelah menentukan nilai TF yaitu pada dokumen d1, d2, dan d3 misalnya kata “suka” muncul sebanyak satu kali pada dokumen “d1”, dan seterusnya pada semua term, dilanjutkan dengan TF Normalisasi untuk mencegah data anomali akibat perbedaan panjang dokumen (total *term* dalam dokumen d). Sehingga, perhitungan TF diperbaiki dengan normalisasi dengan menggunakan persamaan TF pada persamaan (2. 1). Sebagai contoh, berikut adalah perhitungan TF pada *term* “suka” dalam dokumen “d1” sebagai berikut:

|  |  |
| --- | --- |
| TF(d1,suka) = | Frekuensi kemunculan *term* suka dalam dokumen d1 |
| total *term* dalam dokumen d1 |
| = | 1/4 |
| = | 0,25 |

Pada perhitungan IDF (*Inverse Document Frequency),* pertama hitung dulu nilai DF yaitu *document frequency* d1, d2, d3 (>0). Setelah mendapatkan hasil, selanjutnya menghitung nilai IDF menggunakan persamaan (2. 2). Contohnya adalah perhitungan pada *term* “suka” sebagai berikut:

|  |  |
| --- | --- |
| IDF(suka) = log | total dokumen |
| *document frequency* *term* suka |
| = log | 3/1 |
| = log | 3 |
| = 0,477 |  |

Selanjutnya adalah perhitungan TF-IDF dengan persamaan (2. *3*). Contohnya pada *term* “suka” dalam dokumen “d1” sebagai berikut:

|  |  |
| --- | --- |
| TF-IDF(d1,suka) = | TF(d1,suka) × IDF(suka) |
| = | 0,25 × 0,477 |
| = | 0,119 |

### *Splitting Dataset*

Pada penelitian ini menggunakan metode *splitting data* 80/20. *Splitting* data 80/20 adalah metode yang umum digunakan dalam pemrosesan data dan pembelajaran mesin untuk membagi kumpulan data menjadi dua kelompok, yaitu data latih dan data uji, dengan perbandingan 80% untuk data latih dan 20% untuk data uji.

Proses splitting data 80/20 dilakukan dengan memisahkan dataset menjadi dua bagian secara acak. Sebanyak 80% data akan digunakan untuk melatih model atau mengembangkan algoritma pembelajaran mesin, sedangkan 20% data akan digunakan untuk menguji kinerja model atau algoritma yang telah dilatih.

### Pemodelan Klasifikasi Naïve Bayes

Pemodelan klasifikasi naïve bayes merupakan langkah di mana data tweet diklasifikasikan untuk menerima analisis sentimen. Untuk mengklasifikasikan sentimen akan menggunakan data hasil proses preprocessing hingga tahap *splitting* data. Setelah data berhasil dilatih, dilakukan pengujian terhadap data uji untuk menguji hasil akurasi klasifikasi yang dibuat.

Menurut Artanti et al., (2018) ada dua langkah yang dilakukan dalam membangun sistem pengklasifikasi menggunakan Naïve Bayes. Langkah pertama yaitu pelatihan sistem (*training*), lalu langkah kedua yaitu pengujian sistem (testing).

* + - 1. Tahap *Training* (Pelatihan)

Pada tahap ini, sistem pengklasifikasi dilatih menggunakan data pelatihan (data *training*) yang berisi data yang sudah diklasifikasikan atau sudah diberi label baik positif dan negatif

Langkah-langkah dalam tahap *training* adalah sebagai berikut :

* 1. Memperkirakan peluang kategori berdasarkan data pelatihan.
  2. Menentukan frekuensi kata (TF-IDF) dari dataset pelatihan
  3. Melakukan perhitungan peluang untuk setiap kategori di dalam set data pelatihan.

|  |  |  |  |
| --- | --- | --- | --- |
| P(wk) | = | Pk + 1 | (3. 1) |
| n + **|**v**|** |

Keterangan:

|  |  |  |
| --- | --- | --- |
| P(wk) | : | Peluang kata (*term)* |
| Pk + 1 | : | Query yang ditambahkan dengan angka satu untuk menghindari hasil perhitungan yang nol. |
| n | : | Banyaknya keseluruhan total term dalam kelas tertentu pada data training |
| **|**v**|** | : | Total term keseluruhan |

* + - 1. Tahap *Testing* (Pengujian)

Pada tahap ini, data pengujian digunakan untuk menguji kemampuan sistem dalam melakukan klasifikasi pada data baru dan memprediksi label dari data tersebut.

Berikut adalah langkah-langkah dalam tahap pengujian yang dapat diuraikan:

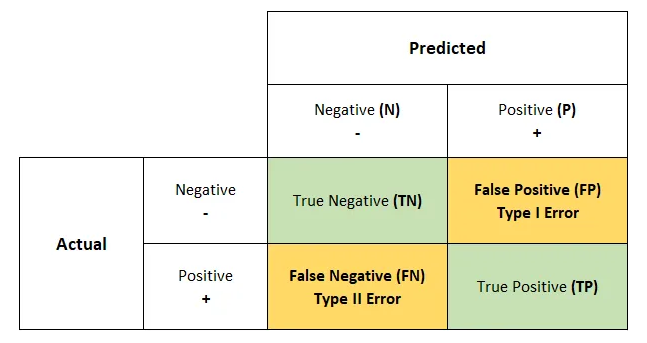
* + - * 1. Menghitung Vmap untuk setiap kelas data uji dengan rumus:

|  |  |  |  |
| --- | --- | --- | --- |
| Vmap | = | P(wk|c).P(c) | (3. 2) |

* + - * 1. Mencari nilai Vmap yang maksimum.
        2. Mengetahui model klasifikasi yang digunakan
        3. Memperoleh kelas prediksi untuk data tersebut

### Evaluasi Model

Setelah menyelesaikan pengujian model, langkah berikutnya adalah melakukan evaluasi terhadap model tersebut. Tujuan evaluasi model untuk mengetahui kinerja model dengan menghasilkan confusion matrix untuk menyajikan perbandingan antara hasil klasifikasi yang diberikan oleh model klasifikasi dengan klasifikasi yang sebenarnya.



Gambar 3. 2 Confusion Matrix

Sumber: medium.com

*Confusion matrix* terdiri dari empat sel utama (Han et al., 2012: 365):

1. *True Positive* (TP): Jumlah data yang sebenarnya positif dan diprediksi sebagai positif oleh model.
2. *False Positive* (FP): Jumlah data yang sebenarnya negatif namun diprediksi sebagai positif oleh model.
3. *True Negative* (TN): Jumlah data yang sebenarnya negatif dan diprediksi sebagai negatif oleh model.
4. *False Negative* (FN): Jumlah data yang sebenarnya positif namun diprediksi sebagai negatif oleh model.

Metode perhitungan *Precision, Accuracy, Recall* dan *F1-Score* digunakan untuk mengukur kinerja klasifikasi*.* Berikut adalah rumus Akurasi, Presisi*, Recall,* dan F1*-Scrore* berdasarkan Han et al., (2012: 365):

1. Akurasi (*Accuracy*)

Akurasi mengukur seberapa banyak prediksi yang benar dari keseluruhan prediksi yang dilakukan oleh model.

|  |  |  |
| --- | --- | --- |
| Akurasi = | (TP + TN) | (3. 3) |
| (TP + FP + TN + FN) |

1. Presisi (*Precision*)

Presisi mengukur seberapa akurat model dalam memprediksi kelas positif.

|  |  |  |
| --- | --- | --- |
| Presisi = | TP | (3. 4) |
| (TP + FP) |

1. *Recall* (*Sensitivitas*)

*Recall* mengukur seberapa baik model dalam mengenali data dengan kelas positif.

|  |  |  |
| --- | --- | --- |
| Recall = | TP | (3. 5) |
| (TP + FN) |

1. F1*-Score*

F1-Score merupakan hasil rata-rata harmonik antara presisi dan recall, dan digunakan untuk menggabungkan kedua matriks tersebut menjadi satu.

|  |  |  |
| --- | --- | --- |
| F1-*Score* = | 2 × (Presisi × *Recall*) | (3. 6) |
| (Presisi + *Recall*) |

### Visualisasi Model

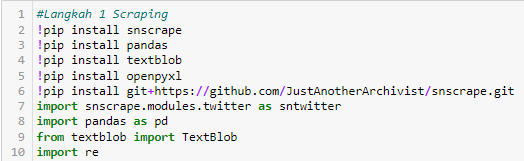
Pada penelitian ini hasil analisis sentimen akan divisualisasikan dengan pie chart dan word cloud. Pie Chart digunakan untuk menampilkan persentase sentimen positif dan negatif dalam dataset analisis sentimen. Pie Chart juga membantu mengidentifikasi sentimen dominan atau sentimen minoritas dalam data analisis sentimen. Sedangkan, Word Cloud digunakan untuk memvisualisasikan kata-kata utama atau kata-kata yang sering muncul dalam analisis teks sentimen baik positif maupun negatif. Word Cloud membantu mengidentifikasi kata-kata yang paling umum atau paling kerap muncul dalam teks dan memberikan gambaran umum dari tema analisis sentimen tersebut.

# BAB IV

# HASIL DAN PEMBAHASAN

## Pengumpulan Data

Langkah pertama ialah mengumpulkan data tweet dengan menginstal *libraries* yang diperlukan dan mengimport modul-modul yang akan digunakan untuk melakukan pengumpulan data *tweet* dari Twitter. Gambar 4.2 menampilkan kode program untuk langkah pertama.



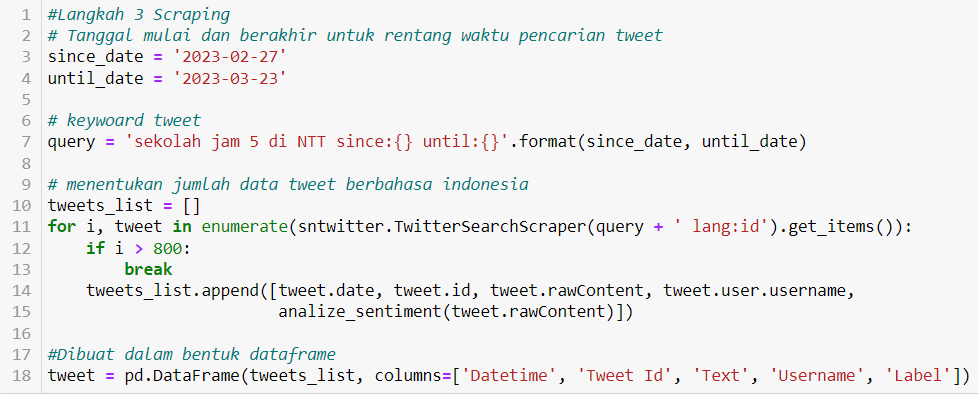
Gambar 4. 1 Kode Program Langkah 1 Scraping

Langkah kedua adalah menerapkan fungsi *clean\_tweet* untuk membersihkan tweet dari karakter tertentu, fungsi *analize\_sentiment* untuk menganalisis sentimen dari tweet, dan fungsi *analize\_subjectorobject* untuk menganalisis apakah subjek atau objek dari tweet tersebut memiliki sentimen positif atau negatif. Gambar 4.2 menampilkan kode program untuk langkah kedua.



Gambar 4. 2 Kode Program Langkah 2 Scraping

Langkah ketiga adalah tahapan untuk mencari dan mengumpulkan tweet berdasarkan rentang waktu dari 27 Februari 2023-23 Maret 2023, dengan kata kunci “sekolah jam 5 di NTT”, dan jumlah tweet yang diambil maksimal sebanyak 800 tweet. Lalu, data tweet yang telah dikumpulkan diubah menjadi sebuah objek *DataFrame*. *DataFrame* akan digunakan untuk menyimpan informasi-informasi terkait tweet, seperti *Datetime, Tweet Id, Text, Username,* dan Label. Gambar 4.2 menampilkan kode program untuk langkah ketiga.



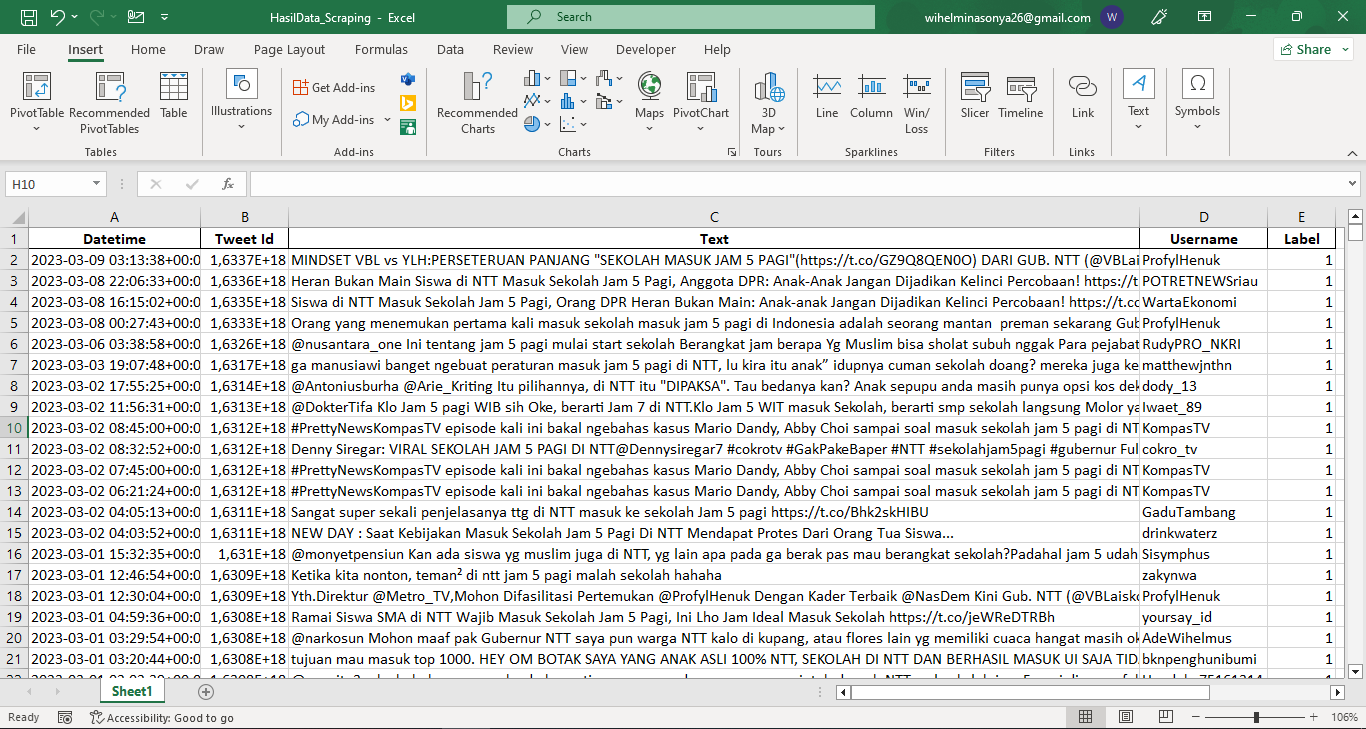
Gambar 4. 3 Kode Program Langkah 3 Scraping

Setelah *DataFrame* terbentuk, langkah keempat adalah menyimpan *DataFrame* tweet ke dalam file Excel dengan nama "HasilData\_Scraping.xlsx". Gambar 4.2 menampilkan kode program untuk langkah keempat.



Gambar 4. 4 Kode Program Langkah 4 *Scraping*

Hasil akhirnya adalah *DataFrame tweet* yang disimpan dalam file Excel dengan nama "HasilData\_Scraping.xlsx" yang berisi data tweet dengan struktur tabel yang terorganisir dengan baik dan menggunakan header yang sesuai dengan nama kolom *DataFrame* yang bisa dilihat pada gambar 4.5.

**

Gambar 4. 5 Hasil Data *Scraping* File Excel

Sehingga, jumlah keseluruhan data tweet yang didapat melalui proses scraping. dalam rentang waktu 27 Februari 2023-23 Maret 2023 dengan kata kunci “sekolah jam 5 di NTT” adalah 777 tweet dengan jumlah label positif (1) adalah 24 *tweet* dan label negatif (-1) adalah 753 *tweet.*

## *Text Preprocessing*

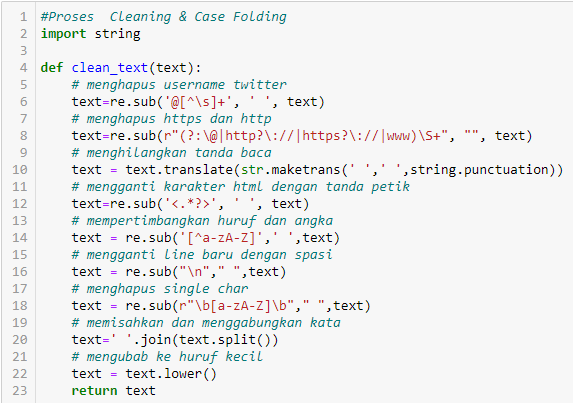
Setelah data sudah terkumpul dan diberi label, tahap berikutnya adalah *text preprocessing.* Tujuan *text preprocessing* yaitu menghasilkan teks yang lebih terstruktur, lebih bersih, dan lebih siap untuk dianalisis oleh model atau algoritma yang telah ditentukan. *Text preprocessing* meliputi *Cleaning, Case Folding, Tokenizing, Normalization, Stopword Removal,* dan *Stemming.*

1. *Cleaning* dan *Case Folding*

Proses *cleaning* berfungsi untuk menghilangkan unsur-unsur yang tidak relevan dalam teks, seperti menghapus tanda baca, angka, karakter khusus, atau simbol yang tidak relevan. Sedangkan, *Case folding* adalah proses yang bertujuan untuk mengkonversi semua karakter huruf dalam sebuah teks menjadi huruf kecilagar tidak ada perbedaan dalam pengenalan kata-kata yang memiliki huruf besar dan huruf kecil.

Pada proses *cleaning* dan *case folding* menggunakan modul python yaitu *string* dan *re*. S*tring* digunakan untuk mengakses konstanta *string.punctuation* yang berisi semua tanda baca. Selain itu, terdapat modul bawaan Python seperti *re* yang digunakan untuk bekerja dengan *regular expression* dalam pemrosesan teks.

Kode program untuk melakukan proses *cleaning* dan *case folding* bisa dilihat pada Gambar 4.6



Gambar 4. 6 Kode Program Proses *Cleaning* dan *Case Folding*

Perbandingan sebelum dan setelah proses *cleaning* dan *case folding* pada teks tweet bisa dilihat pada tabel 4.1.

Tabel 4. 1 Hasil Proses *Cleaning* Dan *Case Folding*

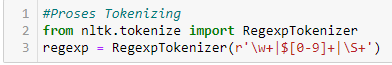
| **Sebelum Cleaning Dan Case Folding** | **Setelah Cleaning Dan Case Folding** |
| --- | --- |
| Kemendikbud Rekomendasikan Kaji Ulang Masuk Sekolah Jam 5.30 Pagi di NTT https://t.co/lL5y4rg9Xo #TempoTekno | kemendikbud rekomendasikan kaji ulang masuk sekolah jam pagi di ntt tempotekno |

1. *Tokenizing*

Proses *tokenizing* berfungsi membagi teks menjadi unit-unit lebih kecil atau token. Pemecahan teks menjadi token membantu pemrosesan teks yang lebih efisien dan analisis yang lebih mendalam.

Pada proses tokenisasi ini digunakan modul *nltk.tokenize* dan spesifiknya kelas *RegexpTokenizer* untuk melakukan pemisahan teks menjadi token.

Gambar 4.7 menampilkan kode program yang digunakan untuk melakukan proses *tokenizing.*



Gambar 4. 7 Kode Program Proses *Tokenizing*

Tabel 4.2 memperlihatkan perbedaan antara sebelum dan sesudah melalui proses tokenisasi pada teks tweet.

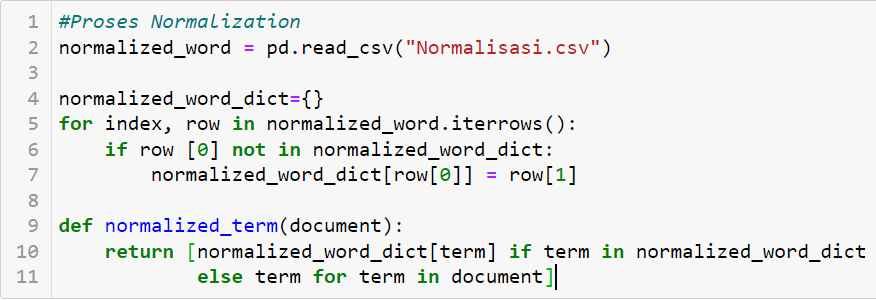
Tabel 4. 2 Hasil Proses *Tokenizing*

| **Sebelum Tokenizing** | **Setelah Tokenizing** |
| --- | --- |
| kemendikbud rekomendasikan kaji ulang masuk sekolah jam pagi di ntt tempotekno | ['kemendikbud', 'rekomendasikan', 'kaji', 'ulang', 'masuk', 'sekolah', 'jam', 'pagi', 'di', 'ntt', 'tempotekno'] |

1. *Normalization*

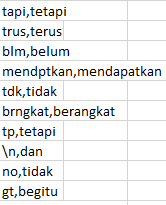
Proses *normalization* berfungsi mengkonversi kata-kata yang ada dalam teks menjadi bentuk dasarnya. Sehingga, mempermudah pengenalan kata yang memiliki makna yang sama.

Gambar 4.8 menampilkan kode program yang digunakan untuk melaksanakan proses normalisasi.



Gambar 4. 8 Kode Program Proses *Normalization*

Kode program ini digunakan untuk untuk membaca file CSV dengan nama "Normalisasi.csv" dan menyimpannya ke dalam variabel *normalized\_word*. File CSV ini berisi data untuk normalisasi kata-kata. Setelah membaca file CSV dan membentuk kamus *normalized\_word\_dict*, fungsi *normalized\_term* digunakan untuk menerapkan normalisasi kata dalam sebuah dokumen. Fungsi tersebut menggantikan kata-kata dalam dokumen dengan bentuk normalisasinya yang sesuai berdasarkan kamus *normalized\_word\_dict.*



Gambar 4. 9 Cuplikan Isi File Normalisasi.csv

Perbandingan sebelum dan setelah *normalization* pada teks tweet bisa dilihat pada tabel 4.3.

Tabel 4. 3 Hasil Proses *Normalization*

| **Sebelum Normalization** | **Setelah Normalization** |
| --- | --- |
| ['kemendikbud', 'rekomendasikan', 'kaji', 'ulang', 'masuk', 'sekolah', 'jam', 'pagi', 'di', 'ntt', 'tempotekno'] | ['kemendikbud', 'rekomendasikan', 'kaji', 'ulang', 'masuk', 'sekolah', 'jam', 'pagi', 'di', 'ntt', 'tempotekno'] |

1. *Stopword Removal*

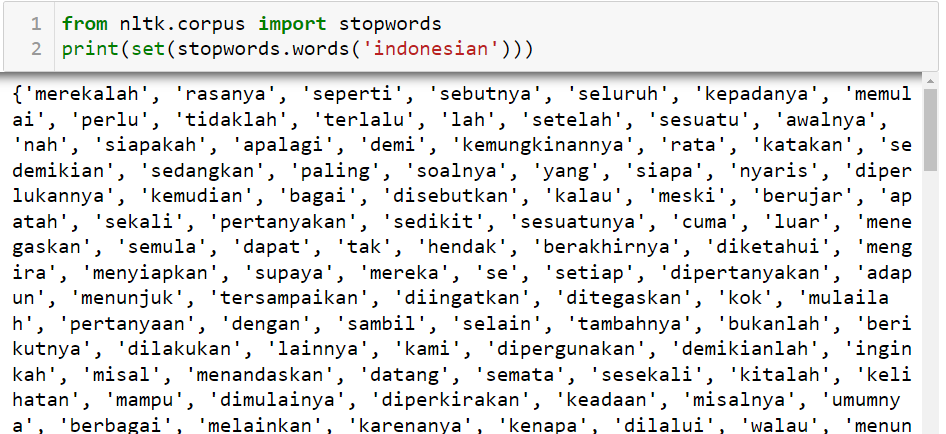
Penghapusan kata-kata umum, yang biasa dikenal sebagai *stopword removal*, bertujuan untuk mengeliminasi kata-kata yang umumnya tidak memiliki relevansi penting dalam analisis teks. Pada tahap ini, menggunakan NLTK ( *Natural Language ToolKit*).

Langkah pertama pada proses stopword removal adalah mengimport nltk. Gambar 4.10 menampilkan kode program yang digunakan untuk mengimport nltk. Kode tersebut menggunakan modul nltk untuk mengunduh korpus "stopwords" (kata-kata yang umumnya diabaikan dalam pemrosesan teks) menggunakan fungsi *nltk.download***()**.



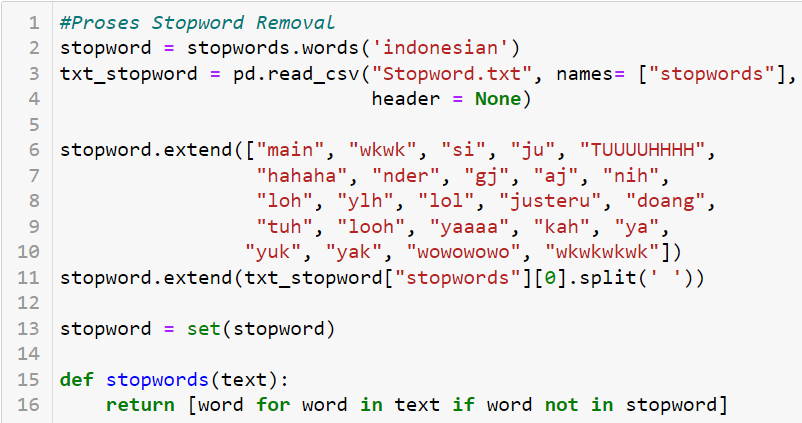
Gambar 4. 10 Kode Program *Import* nltk

Selanjutnya, pada gambar 4.11 kode tersebut menggunakan modul nltk untuk mengimport fungsi *stopwords* dari korpus bahasa. Kemudian, kode mencetak (*print*) set dari daftar stopwords dalam bahasa Indonesia. set ini berisi kumpulan kata-kata yang umumnya diabaikan dalam pemrosesan teks.

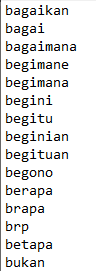


Gambar 4. 11 *List Stopword* Indonesia

Berikutnya, dalam Gambar 4.12 dapat ditemukan kode program yang bertujuan untuk melakukan penghapusan stopword. Selain menggunakan daftar stopwords Indonesia yang telah disediakan oleh modul nltk, bisa ditambah daftar kata yang tidak diperlukan dalam analisis sentimen dengan cara menambahkan kata-kata tersebut ke daftar stopwords menggunakan file tambahan *Stopword.txt* dan metode *stopword.extend.*



Gambar 4. 12 Kode Program Proses *Stopword Removal*



Gambar 4. 13 Cuplikan Isi File Stopword.txt

Perbandingan sebelum dan setelah proses *stopword removal* pada teks tweet bisa dilihat pada tabel 4.4.

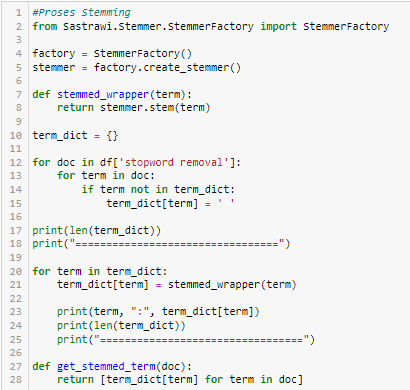
Tabel 4. 4 Hasil Proses *Stopword Removal*

| **Sebelum Stopword Removal** | **Setelah Stopword Removal** |
| --- | --- |
| ['kemendikbud', 'rekomendasikan', 'kaji', 'ulang', 'masuk', 'sekolah', 'jam', 'pagi', 'di', 'ntt', 'tempotekno'] | ['kemendikbud', 'rekomendasikan', 'kaji', 'ulang', 'masuk', 'sekolah', 'jam', 'pagi', 'ntt', 'tempotekno'] |

1. *Stemming*

Proses *stemming* berfungsi mengubah kata-kata menjadi bentuk kata dasar (*root word*) dengan cara menghilangkan imbuhan, baik itu akhiran atau awalan sehingga kata-kata tersebut dapat direduksi menjadi bentuk dasarnya. Tahap ini menggunakan bantuan *library* Sastrawi.

Gambar 4.14 menampilkan kode program yang digunakan untuk melakukan proses stemming.



Gambar 4. 14 Kode program Proses Stemming



Gambar 4. 15 Output Proses Stemming

Perbandingan sebelum dan setelah proses *stemming* pada teks *tweet* bisa dilihat pada tabel 4.5.

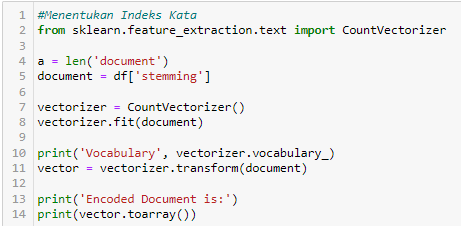
Tabel 4. 5 Hasil Proses Stemming

| **Sebelum Stemming** | **Setelah Stemming** |
| --- | --- |
| ['kemendikbud', 'rekomendasikan', 'kaji', 'ulang', 'masuk', 'sekolah', 'jam', 'pagi', 'ntt', 'tempotekno'] | ['kemendikbud', 'rekomendasi', 'kaji', 'ulang', 'masuk', 'sekolah', 'jam', 'pagi', 'ntt', 'tempotekno'] |

## Pembobotan Kata TF-IDF (*Term Frequency-Inverse Document Frequency*)

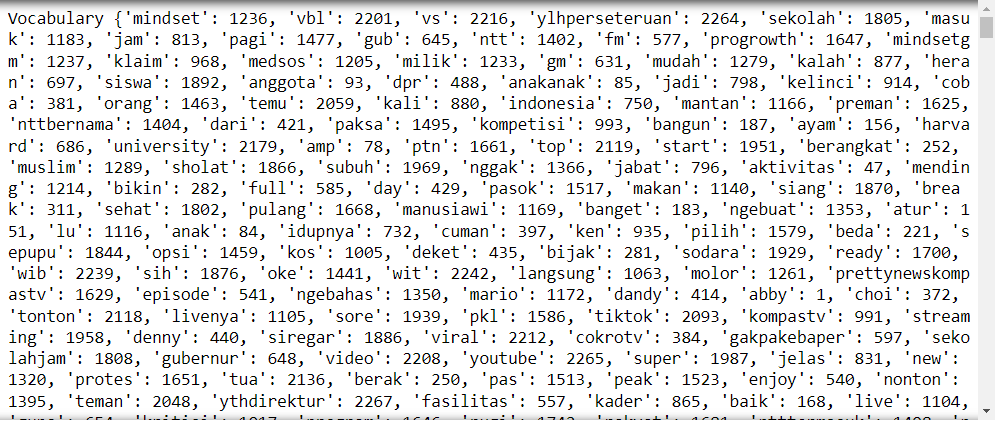
Setelah tahapan *preprocessing,* data yang sudah bersih diberi bobot TF-IDF. Pemberian bobot TF-IDF sebelum dianalisis dengan algoritma yang telah ditentukan penting untuk mengurangi pengaruh kata-kata umum dan meningkatkan penekanan pada kata-kata penting, sehingga meningkatkan kualitas analisis dan klasifikasi.

Langkah pertama, menentukan indeks kata. Tujuannya agar pemrosesan lebih efisien karena operasi pada struktur data numerik cenderung lebih cepat daripada operasi pada teks. Kode program untuk menentukan indeks katadapat dilihat pada Gambar 4.16. Kode tersebut merupakan proses untuk menentukan indeks kata dari dokumen menggunakan *CountVectorizer*.



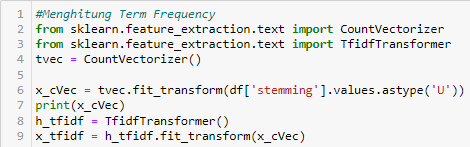
Gambar 4. 16 Kode Program Menentukan Indeks Kata

Output dari menentukan indeks kata dapat dilihat pada gambar 4.17. Output yang dihasilkan adalah kamus yang menunjukkan indeks dari setiap kata dalam representasi vektor. Indeks merupakan angka unik yang diberikan kepada setiap kata dalam urutan yang dihasilkan oleh CountVectorizer. Misalnya, kata ‘*mindset’* memiliki indeks 1236, yang artinya kata ‘*mindset*’ adalah kata ke-1236 dalam kamus atau urutan kata yang sedang diproses.



Gambar 4. 17 Output Hasil Proses *Term Frequency*

Selanjutnya, menghitung *term frequency.* Gambar 4.18 menunjukkan kode program yang digunakan untuk menghitung frekuensi kata (*term frequency*). Kode tersebut merupakan proses penghitungan *term frequency* menggunakan *CountVectorizer* dan *TfidfTransformer* dari *scikit-learn*.



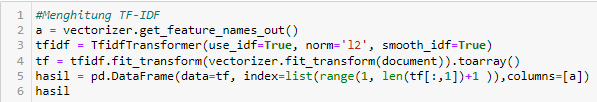
Gambar 4. 18 Kode Program Menghitung *Term Frequency*

Gambar 4.19 menunjukkan hasil dari perhitungan frekuensi kata-kata dalam dokumen. *Output* tersebut memberikan informasi mengenai seberapa sering kata-kata tersebut muncul dalam dokumen. Misalnya, pada baris pertama "(0, 1236) 2", yang artinya dalam dokumen pertama (indeks 0), kata dengan indeks 1236 (mindset) muncul sebanyak 2 kali dalam dokumen tersebut.

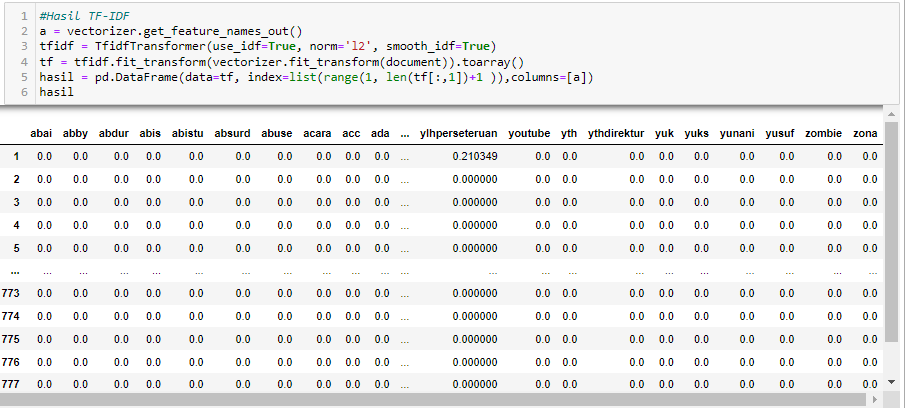


Gambar 4. 19 Output Menghitung *Term Frequency*

Selanjutnya, menghitung TF-IDF (*Term Frequency-Inverse Document Frequency*) lalu menyimpan hasilnya ke dalam bentuk DataFrame. Kode program menghitung TF-IDFbisa dilihat pada Gambar 4.20 sedangkan Gambar 4.21 adalah *output* dari hasil perhitungan TF-IDF.



Gambar 4. 20 Kode Program Menghitung TF-IDF



Gambar 4. 21 Hasil TF-IDF

Di bawah ini terdapat contoh hasil perhitungan TF-IDF (*Term Frequency-Inverse Document Frequency*) yang dilakukan secara manual pada tweet “protes orang tua murid masuk sekolah jam pagi ntt” yang bisa dilihat pada tabel 4.6.

Tabel 4. 6 Hasil Pembobotan TF-IDF

| **TERM** | **TF** | **TF Normalisasi** | **DF** | **IDF** | **TF-IDF** |
| --- | --- | --- | --- | --- | --- |
| protes | 1 | 0,11111 | 12 | 1,81124 | 0,20125 |
| orang | 1 | 0,11111 | 63 | 1,09108 | 0,12123 |
| tua | 1 | 0,11111 | 27 | 1,45906 | 0,16212 |
| murid | 1 | 0,11111 | 27 | 1,45906 | 0,16212 |
| masuk | 1 | 0,11111 | 556 | 0,14535 | 0,01615 |
| sekolah | 1 | 0,11111 | 775 | 0,00112 | 0,00012 |
| jam | 1 | 0,11111 | 777 | 0 | 0 |
| pagi | 1 | 0,11111 | 609 | 0,1058 | 0,01176 |
| ntt | 1 | 0,11111 | 773 | 0,00224 | 0,00025 |
| **Panjang** | 9 |  |  |  |  |
| **Dokumen** |

Langkah awal adalah menentukan nilai *Term frequency* (TF) adalah metode yang digunakan untuk mengukur seberapa sering sebuah kata muncul dalam sebuah dokumen. Setelah menentukan nilai TF, misalnya term “protes” muncul sebanyak satu kali dan seterusnya pada semua term di dalam dokumen, dilanjutkan dengan TF Normalisasi menggunakan persamaan TF pada persamaan (2. 1). Sebagai contoh, berikut adalah perhitungan TF pada term “protes” sebagai berikut:

|  |  |
| --- | --- |
| TF(protes) = | Frekuensi kemunculan *term* protes dalam document |
| total *term* dalam dokumen |
| = | 1/9 |
| = | 0,11111 |

Selanjutnya, untuk menghitung nilai IDF (*Inverse Document Frequency*), pertama hitung dulu nilai DF yaitu *document frequency*. Nilai DF adalah frekuensi kemunculan suatu kata atau term dalam keseluruhan dokumen yang ada pada dataset. Jika kata tersebut muncul di setiap dokumen, artinya IDF-nya akan menjadi 0. Contohnya adalah perhitungan pada term “protes”. Jadi, terdapat 777 dokumen *tweet* pada dataset dan kata “protes” terdapat pada 12 dokumen, maka pembobotan IDF *term* “protes” menggunakan persamaan (2. 2) sebagai berikut:

|  |  |  |  |
| --- | --- | --- | --- |
| IDF(protes) | = | log | total dokumen |
| *document frequency* *term* protes |
|  | = | log | 777/12 |
|  | = | log | 64,75 |
|  | = | 1,81124 | |

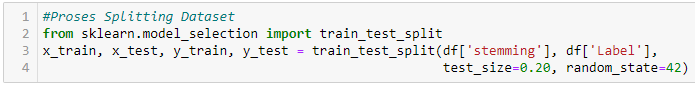
Selanjutnya adalah perhitungan TF-IDF. TF-IDF menghasilkan nilai yang lebih tinggi untuk kata-kata yang sering muncul dalam suatu dokumen tetapi jarang muncul dalam dokumen-dokumen lain dalam dataset. Perhitungan TF-IDF menggunakan persamaan (2. *3*), contohnya pada *term* “protes” sebagai berikut:

|  |  |
| --- | --- |
| TF-IDF(protes) = | TF(protes) × IDF(protes) |
| = | 0,11111 × 1,81124 |
| = | 0,20125 |

## Splitting Dataset

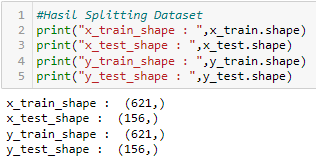
Setelah data diberi nilai bobot TF-IDF, langkah berikutnya adalah membagi data menjadi dua bagian, yaitu data *training* dan data *testing*. Dalam penelitian ini, proporsi pembagian datanya adalah 80% untuk data *training* dan 20% untuk data *testing.*

Gambar 4.22 menampilkan kode program untuk melakukan proses *splitting dataset.*



Gambar 4. 22 Kode Program Proses *Splitting Dataset*

Kode tersebut menggunakan fungsi *train\_test\_split* dari pustaka *scikit-learn* (*sklearn*) untuk membagi dataset menjadi set latih dan set uji. *test\_size=*0.20 digunakan untuk menentukan ukuran set pengujian dalam proporsi dataset. Dalam penelitian ini, 20% dari data akan menjadi set pengujian, sementara 80% akan menjadi set pelatihan. *random\_state*=42 digunakan untuk menentukan bilangan acak yang digunakan untuk membagi dataset. Gambar 4.23 menampilkan hasil pembagian jumlah data yang telah dilakukan. Oleh karena itu, terdapat 621 data latih dan 156 data uji yang telah dihasilkan.

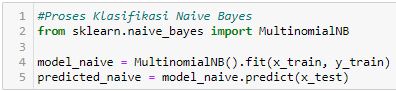


Gambar 4. 23 Hasil *Splitting Dataset*

## Pengujian Model Klasifikasi Naïve Bayes

Setelah pembagian dataset dilakukan maka data siap untuk diterapkan model klasifikasi Naïve Bayes. Gambar 4.24 menampilkan kode program untuk melakukan klasifikasi Naïve Bayes.

Proses klasifikasi Naive Bayes menggunakan metode *MultinomialNB* dari pustaka *scikit-learn* (*sklearn*). Pertama, mengimport kelas *MultinomialNB*. Selanjutnya, membuat objek *model\_naive* sebagai *instance* dari kelas *MultinomialNB*() dan melatihnya menggunakan data pelatihan *x\_train* dan *y\_train*. Setelah model dilatih, lanjut menggunakan data uji *x\_test* untuk melakukan prediksi menggunakan *metode predict*() dan hasil prediksinya disimpan dalam variabel *predicted\_naive*.



Gambar 4. 24 Kode Program Proses Klasifikasi Naïve Bayes

Berikut ini merupakan contoh penghitungan klasifikasi sentimen secara manual menggunakan algoritma Naïve Bayes. Tabel 4.7 menampilkan kumpulan data yang terdiri dari 2 data training dan 1 data testing.

Data uji akan digunakan untuk menguji sejauh mana kinerja model klasifikasi yang dilatih menggunakan data latih. Dengan menggunakan model yang sudah dipelajari dari data training, akan mencoba memprediksi sentimen dari isi dokumen d3, sehingga dapat menentukan labelnya, yaitu -1 atau 1.

Tabel 4. 7 Data Sentimen

|  | **ID Dokumen** | **Isi Dokumen** | **Label** |
| --- | --- | --- | --- |
| **Data Training** | d1 | masuk sekolah jam pagi ntt dokter anak imunitas tubuh siswa turun | -1 |
| d2 | ramai siswa sma ntt wajib masuk sekolah jam pagi jam ideal masuk sekolah | 1 |
| **Data Testing** | d3 | siswa ntt masuk sekolah jam pagi orang dpr heran anakanak jadi kelinci coba | ? |

Langkah awal, menghitung probabilitas kelas yaitu menghitung jumlah masing-masing kelas dalam dataset training. Caranya adalah membagi jumlah masing-masing kelas dengan jumlah total data training. Diketahui bahwa jumlah seluruh data training adalah 621 *tweet* dengan 601 adalah *tweet* negatif dan 20 adalah *tweet* positif.

Menghitung probabilitas kelas dapat memberikan wawasan tentang sejauh mana model Anda dapat menggeneralisasi data *training*. Probabilitas yang tinggi atau dekat dengan 1 menunjukkan bahwa model dengan tingkat kepercayaan yang tinggi dalam memprediksi kelas tertentu. Perhitungan peluang kemunculan kategori positif dan negatif (probabilitas kelas) sebagai berikut:

|  |  |  |
| --- | --- | --- |
| P(negatif) | = | Jumlah data training negatif |
| Jumlah semua data training |
|  | = | 601 |
|  |  | 621 |
|  | = | 0,96779 |
| P(positif) | = | Jumlah data training positif |
| Jumlah semua data training |
|  | = | 20 |
|  |  | 621 |
|  | = | 0,03221 |

Selanjutnya, pada tabel 4.8 untuk melihat seberapa sering dokumen d3 muncul dalam dokumen-dokumen yang ada dalam data training, yang terbagi menjadi dua kategori yaitu negatif dan positif. Misalnya, kata "siswa" muncul sebanyak 142 kali dalam dokumen-dokumen dengan sentimen negatif dan hanya 7 kali dalam dokumen-dokumen dengan sentimen positif.

Tabel 4. 8 Frekuensi Kemunculan Kata Pada Data *Training*

| **Term** | **Frekuensi Kemunculan Kata Pada Data Training** | |
| --- | --- | --- |
| **Negatif** | **Positif** |
| siswa | 142 | 7 |
| ntt | 721 | 24 |
| masuk | 496 | 20 |
| sekolah | 750 | 23 |
| jam | 853 | 36 |
| pagi | 570 | 19 |
| orang | 53 | 2 |
| dpr | 20 | 1 |
| heran | 2 | 1 |
| anakanak | 2 | 1 |
| jadi | 5 | 1 |
| kelinci | 2 | 1 |
| coba | 24 | 1 |

Setelah menentukan frekuensi kemunculan kata pada data training, selanjutnya adalah menghitung probabilitas kondisional yaitu P(term|Negatif) dan P(term|Positif) dengan membagi frekuensi kemunculan kata dengan total kata dalam kelas yang sesuai menggunakan persamaan (3. 1). Sebelumnya, diketahui bahwa total jumlah term yang ada pada data *training yaitu* 9718 term, dengan jumlah term berkategori negatif yaitu 9343 term dan jumlah term yang ada pada data *training* berkategori positif yaitu 375 term.

**Kelas Negatif**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| P(siswa | Negatif) | **=** | (142 + 1)/9343 + 9718 | **=** | 0,0075 |
| P(ntt **|** Negatif) | **=** | (721 + 1)/ 9343 + 9718 | **=** | 0,03788 |
| P(masuk **|** Negatif) | **=** | (496 + 1)/ 9343 + 9718 | **=** | 0,02607 |
| P(sekolah **|** Negatif) | **=** | (750+1)/ 9343 + 9718 | **=** | 0,0394 |
| P(jam **|** Negatif) | **=** | (853 + 1)/ 9343 + 9718 | **=** | 0,0448 |
| P(pagi **|** Negatif) | **=** | (570 + 1)/ 9343 + 9718 | **=** | 0,02996 |
| P(orang **|** Negatif) | **=** | (53 + 1)/ 9343 + 9718 | **=** | 0,00283 |
| P(dpr **|** Negatif) | **=** | (20 + 1)/ 9343 + 9718 | **=** | 0,0011 |
| P(heran **|** Negatif) | **=** | (2 + 1)/ 9343 + 9718 | **=** | 0,00016 |
| P(anakanak **|** Negatif) | **=** | (2 + 1)/ 9343 + 9718 | **=** | 0,00016 |
| P(jadi **|** Negatif) | **=** | (5 + 1)/ 9343 + 9718 | **=** | 0,00031 |
| P(kelinci **|** Negatif) | **=** | (2 + 1)/ 9343 + 9718 | **=** | 0,00016 |
| P(coba **|** Negatif) | **=** | (24 + 1)/ 9343 + 9718 | **=** | 0,00131 |

**Kelas Positif**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| P(siswa | Positif) | **=** | (7 + 1)/ 375 + 9718 | **=** | 0,00079 |
| P(ntt **|** Positif) | **=** | (24 + 1)/ 375 + 9718 | **=** | 0,00248 |
| P(masuk **|** Positif) | **=** | (20 + 1)/ 375 + 9718 | **=** | 0,00208 |
| P(sekolah **|** Positif) | **=** | (23 + 1)/ 375 + 9718 | **=** | 0,00238 |
| P(jam **|** Positif) | **=** | (36 + 1)/ 375 + 9718 | **=** | 0,00367 |
| P(pagi **|** Positif) | **=** | (19 + 1)/ 375 + 9718 | **=** | 0,00198 |
| P(orang **|** Positif) | **=** | (2 + 1)/ 375 + 9718 | **=** | 0,0003 |
| P(dpr **|** Positif) | **=** | (1 + 1)/ 375 + 9718 | **=** | 0,0002 |
| P(heran **|** Positif) | **=** | (1 + 1)/ 375 + 9718 | **=** | 0,0002 |
| P(anakanak **|** Positif) | **=** | (1 + 1)/ 375 + 9718 | **=** | 0,0002 |
| P(jadi **|** Positif) | **=** | (1 + 1)/ 375 + 9718 | **=** | 0,0002 |
| P(kelinci **|** Positif) | **=** | (1 + 1)/ 375 + 9718 | **=** | 0,0002 |
| P(coba **|** Positif) | **=** | (1 + 1)/ 375 + 9718 | **=** | 0,0002 |

Setelah memperoleh nilai probabilitas untuk setiap kelas, langkah selanjutnya adalah melakukan klasifikasi pada tweet menggunakan persamaan (3. 2). Berikut adalah hasil klasifikasi yang didapatkan.

**Kelas Negatif**

|  |  |  |
| --- | --- | --- |
| P(Negatif | c) | **=** | 0,0075 × 0,03788 × 0,02607 × 0,0394 × 0,0448 × 0,02996 × 0,00283 × 0,0011 × 0,00016 × 0,00016 × 0,00031 × 0,00016 × 0,00131 |
|  | **=** | 2,0282 × 10-33 |

**Kelas Positif**

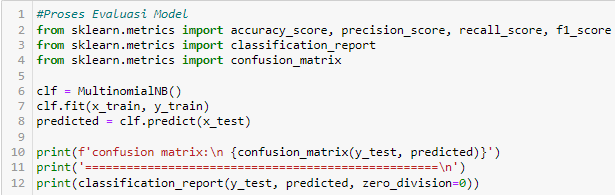
|  |  |  |
| --- | --- | --- |
| P(Positif | c) | **=** | 0,00079 × 0,00248 × 0,00208 × 0,00238 × 0,00367 × 0,00198 × 0,0003 × 0,0002 × 0,0002 × 0,0002 × 0,0002 × 0,0002 × 0,0002 |
|  | **=** | 1,3532×10-42 |

Berdasarkan perhitungan di atas, didapatkan hasil probabilitas 2,0282 × 10-33 untuk kelas negatif dan 1,3532×10-42 untuk kelas positif. Karena nilai probabilitas kelas negatif lebih tinggi daripada nilai probabilitas kelas positif, maka data uji pada dokumen d3 diklasifikasikan sebagai "negatif” atau “-1”.

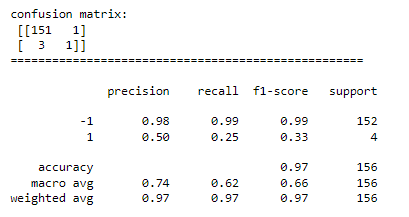
## Evaluasi Model

Setelah proses pengujian model selesai, langkah berikutnya adalah melakukan evaluasi terhadap model untuk menilai performanya dan menghasilkan *confusion matrix*.

Kode program proses evaluasi model dapat dilihat pada gambar 4.25. Kode tersebut melakukan evaluasi model menggunakan pustaka *scikit-learn* untuk melakukan perhitungan metrik evaluasi seperti akurasi, presisi, *recall*, F1-*Score*, dan *confusion matrix*. Model klasifikasi Naïve Bayes Multinomial (MultinomialNB) dilatih dengan data pelatihan (x*\_train* dan y*\_train*), kemudian digunakan untuk melakukan prediksi pada data pengujian (x*\_test*). Hasil prediksi tersebut kemudian digunakan untuk mencetak *confusion matrix* dan laporan klasifikasi yang mencakup metrik-metrik evaluasi tersebut. Gambar 4.26 menunjukkan hasil evaluasi model dengan menggunakan *confusion matrix,* yang membandingkan hasil klasifikasi model dengan hasil klasifikasi sebenarnya.



Gambar 4. 25 Kode Program Proses Evaluasi Model

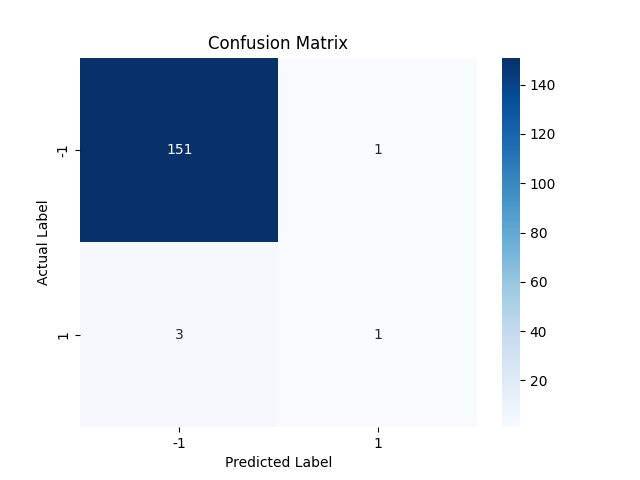


Gambar 4. 26 Hasil Evaluasi Model

Setelah itu, dilakukan visualisasi dari hasil *confusion matrix*. Kode program yang digunakan untuk memvisualisasikan *confusion matrix* dapat ditemukan pada gambar 4.27. Kode tersebut melakukan visualisasi *confusion matrix* dengan menggunakan pustaka *matplotlib* dan *seaborn*. *Confusion matrix* dan label-labelnya didefinisikan terlebih dahulu, kemudian dibuat *heatmap confusion matrix* yang menampilkan angka pada setiap sel. Setelah itu, dilakukan penyesuaian label pada sumbu x dan y, serta memberikan judul pada plot. Terakhir, plot tersebut disimpan sebagai file gambar dengan format "confusion\_matrix.png". Gambar 4.28 menampilkan hasil visualisasi dari *confusion matrix.*



Gambar 4. 27 Kode Program Proses Visualisasi *Confusion Matrix*



Gambar 4. 28 Hasil Visualisasi *Confusion Matrix*

Hasil evaluasi model dengan confusion matrix pada gambar 4.28 juga dapat dilihat pada tabel 4.6.

Tabel 4. 9 Hasil *Confusion Matrix*

|  | | Predicted | |
| --- | --- | --- | --- |
| Negatif  -1 | Positif  1 |
| Actual | Negatif  -1 | 151 (TN) | 1 (FP) |
| Positif  1 | 3 (FN) | 1 (TP) |

Dari tabel dan juga gambar dari hasil *confusion matrix* didapatkan informasi sebagai berikut:

1. *True Positive* (TP): Jumlah data yang sebenarnya positif dan diprediksi sebagai positif oleh model sebanyak 1.
2. *False Positive* (FP): Jumlah data yang sebenarnya negatif namun diprediksi sebagai positif oleh model sebanyak 1.
3. *True Negative* (TN): Jumlah data yang sebenarnya negatif dan diprediksi sebagai negatif oleh model sebanyak 151.
4. *False Negative* (FN): Jumlah data yang sebenarnya positif namun diprediksi sebagai negatif oleh model 3.

Data yang dihasilkan *confusion matrix* akan menjadi nilai-nilai yang digunakan untuk menghitung beberapa metrik seperti akurasi, presisi, *recall*, dan f1-*score*. Tabel 4.7 menunjukkan hasil evaluasi model yang juga terdapat pada gambar 4.26, dengan hasil *accuracy* sebesar 0,97.

Tabel 4. 10 Laporan Hasil Evaluasi Model

|  |  |  |  |
| --- | --- | --- | --- |
| Jenis Sentimen | Precision | Recall | F1-Score |
| Negatif  -1 | 0,98 | 0,99 | 0,99 |
| Positif  1 | 0,50 | 0,25 | 0,33 |

Akurasi mengukur sejauh mana model dapat membuat prediksi yang benar secara keseluruhan. Perhitungan akurasi manual dari hasil evaluasi model sebagai berikut:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Akurasi | = | (TP + TN)  (TP + FP + TN + FN) | × | 100% |
|  | = | (1 + 151)  (1 + 1 + 151 + 3) |  |  |
|  |
|  | = | 152  156 |  |  |
|  |
|  | = | 0,97 |  |  |
|  | = | 97% |  |  |

Presisi untuk sentimen negatif mengukur seberapa akurat model dalam memprediksi kelas negatif. Perhitungan manual presisi untuk sentimen negatif dari hasil evaluasi model adalah sebagai berikut:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Presisi Negatif | = | TN  (TN + FN) | × | 100% |
|  | = | 151  (151 + 3) |  |  |
|  |
|  | = | 151  154 |  |  |
|  |
|  | = | 0,98 |  |  |
|  | = | 98% |  |  |

Presisi untuk sentimen positif mengukur seberapa akurat model dalam memprediksi kelas positif. Perhitungan manual presisi untuk sentimen positif dari hasil evaluasi model adalah sebagai berikut:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Presisi Positif | = | TP  (TP + FP) | × | 100% |
|  | = | 1  (1 + 1) |  |  |
|  |
|  | = | 1  2 |  |  |
|  |
|  | = | 0,50 |  |  |
|  | = | 50% |  |  |

*Recall* untuk sentimen negatif mengukur seberapa baik model dalam mengenali data dengan kelas negatif. Perhitungan manual *recall* untuk sentimen negatif dari hasil evaluasi model adalah sebagai berikut:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Recall Negatif | = | TN  (TN + FP) | × | 100% |
|  | = | 151  (151 + 1) |  |  |
|  |
|  | = | 151 |  |  |
|  | 152 |
|  | = | 0,99 |  |  |
|  | = | 99% |  |  |

*Recall* untuk sentimen positif mengukur seberapa baik model dalam mengenali data dengan kelas positif. Perhitungan manual *recall* untuk sentimen positif dari hasil evaluasi model adalah sebagai berikut:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Recall Positif | = | TP  (TP + FN) | × | 100% |
|  | = | 1  (1 + 3) |  |  |
|  |
|  | = | 1  4 |  |  |
|  |
|  | = | 0,25 |  |  |
|  | = | 25% |  |  |

F1*-Score* adalah *harmonic* mean antara presisi dan recall yang berguna ketika ingin mencapai keseimbangan antara presisi dan recall. Perhitungan manual F1*-Score* untuk sentimen negatif dari hasil evaluasi model adalah sebagai berikut:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| F1-Score  Negatif | = | 2 × (Presisi Negatif × Recall Negatif)  (Presisi Negatif + Recall Negatif) | × | 100% |
|  | = | 2 × (0,98 × 0,99)  (0,98 + 0,99) |  | |
| = | 1,94  1,97 |
| = | 0,99 |
| = | 99% |

Perhitungan manual F1*-Score* untuk sentimen positif dari hasil evaluasi model adalah sebagai berikut:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| F1-Score  Positif | = | 2 × (Presisi Positif × Recall Positif)  (Presisi Positif + Recall Positif) | × | 100% |
|  | = | 2 × (0,50 × 0,25)  (0,50 + 0,25) |  |  |
|  |
|  | = | 0,25  0,75 |  |  |
|  |
|  | = | 0,33 |  |  |
|  | = | 33% |  |  |

## Hasil Klasifikasi Model

Pada hasil klasifikasi ini akan ditampilkan seluruh data uji sebanyak 156 data yang telah diprediksi dengan model, yang disajikan pada tabel 4.8.

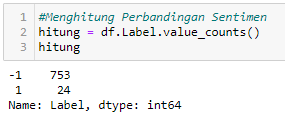
Tabel 4. 11 Hasil Klasifikasi Dari Model

| **Actual Label** | **Predicted Label** | **Data Ke-** |
| --- | --- | --- |
| -1 | -1 | 1 |
| -1 | -1 | 2 |
| -1 | -1 | 3 |
| -1 | -1 | 4 |
| -1 | 1 | 5 |
| -1 | -1 | 6 |
| -1 | -1 | 7 |
| -1 | -1 | 8 |
| -1 | -1 | 9 |
| -1 | -1 | 10 |
| -1 | -1 | 11 |
| -1 | -1 | 12 |
| -1 | -1 | 13 |
| -1 | -1 | 14 |
| -1 | -1 | 15 |
| -1 | -1 | 16 |
| -1 | -1 | 17 |
| -1 | -1 | 18 |
| -1 | -1 | 19 |
| -1 | -1 | 20 |
| -1 | -1 | 21 |
| -1 | -1 | 22 |
| -1 | -1 | 23 |
| -1 | -1 | 24 |
| 1 | -1 | 25 |
| -1 | -1 | 26 |
| -1 | -1 | 27 |
| -1 | -1 | 28 |
| -1 | -1 | 29 |
| -1 | -1 | 30 |
| -1 | -1 | 31 |
| -1 | -1 | 32 |
| -1 | -1 | 33 |
| -1 | -1 | 34 |
| -1 | -1 | 35 |
| -1 | -1 | 36 |
| -1 | -1 | 37 |
| -1 | -1 | 38 |
| -1 | -1 | 39 |
| -1 | -1 | 40 |
| -1 | -1 | 41 |
| -1 | -1 | 42 |
| -1 | -1 | 43 |
| -1 | -1 | 44 |
| -1 | -1 | 45 |
| -1 | -1 | 46 |
| -1 | -1 | 47 |
| -1 | -1 | 48 |
| -1 | -1 | 49 |
| -1 | -1 | 50 |
| -1 | -1 | 51 |
| -1 | -1 | 52 |
| -1 | -1 | 53 |
| -1 | -1 | 54 |
| -1 | -1 | 55 |
| -1 | -1 | 56 |
| -1 | -1 | 57 |
| -1 | -1 | 58 |
| -1 | -1 | 59 |
| -1 | -1 | 60 |
| -1 | -1 | 61 |
| -1 | -1 | 62 |
| -1 | -1 | 63 |
| -1 | -1 | 64 |
| -1 | -1 | 65 |
| -1 | -1 | 66 |
| -1 | -1 | 67 |
| -1 | -1 | 68 |
| -1 | -1 | 69 |
| -1 | -1 | 70 |
| -1 | -1 | 71 |
| -1 | -1 | 72 |
| -1 | -1 | 73 |
| -1 | -1 | 74 |
| -1 | -1 | 75 |
| -1 | -1 | 76 |
| -1 | -1 | 77 |
| -1 | -1 | 78 |
| -1 | -1 | 79 |
| -1 | -1 | 80 |
| -1 | -1 | 81 |
| -1 | -1 | 82 |
| -1 | -1 | 83 |
| -1 | -1 | 84 |
| -1 | -1 | 85 |
| -1 | -1 | 86 |
| -1 | -1 | 87 |
| -1 | -1 | 88 |
| -1 | -1 | 89 |
| -1 | -1 | 90 |
| -1 | -1 | 91 |
| -1 | -1 | 92 |
| 1 | 1 | 93 |
| -1 | -1 | 94 |
| -1 | -1 | 95 |
| -1 | -1 | 96 |
| -1 | -1 | 97 |
| -1 | -1 | 98 |
| -1 | -1 | 99 |
| -1 | -1 | 100 |
| -1 | -1 | 101 |
| -1 | -1 | 102 |
| -1 | -1 | 103 |
| -1 | -1 | 104 |
| -1 | -1 | 105 |
| -1 | -1 | 106 |
| -1 | -1 | 107 |
| -1 | -1 | 108 |
| -1 | -1 | 109 |
| -1 | -1 | 110 |
| -1 | -1 | 111 |
| -1 | -1 | 112 |
| -1 | -1 | 113 |
| -1 | -1 | 114 |
| -1 | -1 | 115 |
| -1 | -1 | 116 |
| -1 | -1 | 117 |
| -1 | -1 | 118 |
| -1 | -1 | 119 |
| -1 | -1 | 120 |
| -1 | -1 | 121 |
| -1 | -1 | 122 |
| -1 | -1 | 123 |
| -1 | -1 | 124 |
| -1 | -1 | 125 |
| -1 | -1 | 126 |
| -1 | -1 | 127 |
| -1 | -1 | 128 |
| -1 | -1 | 129 |
| -1 | -1 | 130 |
| -1 | -1 | 131 |
| -1 | -1 | 132 |
| 1 | -1 | 133 |
| -1 | -1 | 134 |
| -1 | -1 | 135 |
| -1 | -1 | 136 |
| -1 | -1 | 137 |
| -1 | -1 | 138 |
| 1 | -1 | 139 |
| -1 | -1 | 140 |
| -1 | -1 | 141 |
| -1 | -1 | 142 |
| -1 | -1 | 143 |
| -1 | -1 | 144 |
| -1 | -1 | 145 |
| -1 | -1 | 146 |
| -1 | -1 | 147 |
| -1 | -1 | 148 |
| -1 | -1 | 149 |
| -1 | -1 | 150 |
| -1 | -1 | 151 |
| -1 | -1 | 152 |
| -1 | -1 | 153 |
| -1 | -1 | 154 |
| -1 | -1 | 155 |
| -1 | -1 | 156 |

## Visualisasi Model

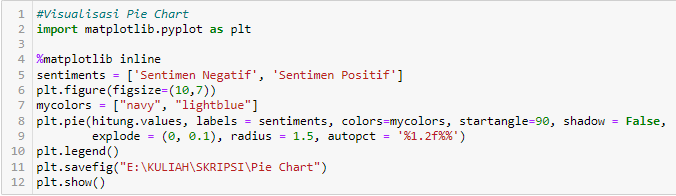
* + - 1. *Pie Chart*

Pie Chart adalah metode visualisasi yang berguna untuk memperlihatkan proporsi atau persentase dari sentimen positif dan negatif dalam dataset analisis sentimen. Diketahui jumlah sentimen negatif sebanyak 753 dan sentimen positif sebanyak 24. Kode program untuk menghitung jumlah sentimen negatif dan positif bisa dilihat pada gambar 4.29.



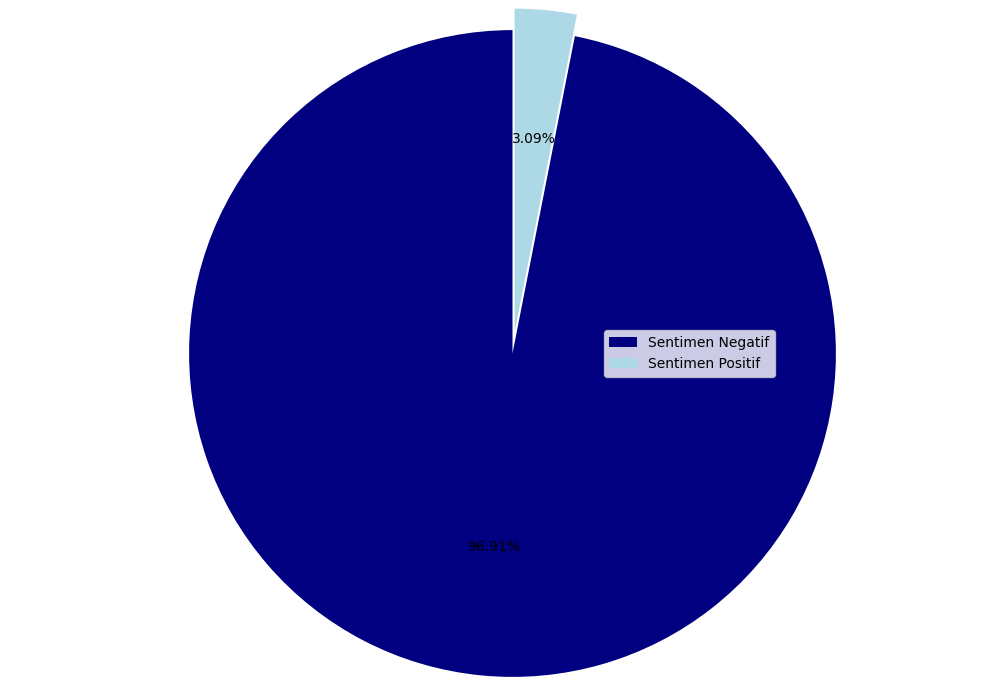
Gambar 4. 29 Kode Program Menghitung Jumlah Sentimen

Berdasarkan nilai-nilai yang telah dihitung sebelumnya dalam variabel 'hitung', lanjut membuat visualisasi *Pie Chart* yang menggambarkan persentase perbandingan antara sentimen negatif dan sentimen positif menggunakan *library matplotlib.*Kemudian, objek *pie chart* dibuat dengan mengatur ukuran, warna, dan atribut lainnya yang bisa dilihat pada gambar 4.30.



Gambar 4. 30 Kode Program Visualisasi *Pie Chart*

Gambar 4.31 menunjukkan visualisasi pie chart yang menampilkan hasil data secara grafis. Berdasarkan hasil *pie chart* diketahui bahwa sentimen negatif sebesar 96,91% dan sentimen positif sebesar 3,09%, yang artinya bahwa lebih dominan *tweet* yang negatif dibandingkan *tweet* positif.



Gambar 4. 31 Hasil Visualisasi *Pie Chart*

* + - 1. *Word Cloud*

*Word cloud* digunakan menunjukkan kata-kata kunci atau kata yang sering muncul dalam analisis sentimen pada teks, baik positif maupun negatif. Untuk membuat visualisasi kata-kata dalam bentuk *word cloud* menggunakan *library wordcloud*. Gambar 4.32 menampilkan kode program yang digunakan untuk mengimport *library word cloud*.



Gambar 4. 32 Kode Program *Import library Word Cloud*

Proses untuk visualisasi *word cloud* menggunakan secara keseluruhan dari semua kata-kata dalam dataset. Kemudian, objek *word cloud* dibuat dengan mengatur ukuran, latar belakang, dan atribut lainnya yang bisa dilihat pada gambar 4.32.



Gambar 4. 33 Kode Program Proses Visualisasi *Word Cloud*

Tampilan visual dari *word cloud* dapat ditemukan pada gambar 4.34. Gambar tersebut menunjukkan sekumpulan kata yang sering muncul dalam data teks. Kata-kata tersebut ditampilkan dengan ukuran yang berbeda-beda sesuai dengan frekuensi kemunculannya.



Gambar 4. 34 Hasil Visualisasi *Word Cloud*

Tabel 4.12 menunjukkan sepuluh term atau kata yang paling banyak muncul dari visualisasi menggunakan *word cloud*

Tabel 4. 12 Hasil Word Cloud

| **Term** | **Banyaknya muncul** |
| --- | --- |
| jam | 1060 |
| sekolah | 884 |
| ntt | 793 |
| pagi | 670 |
| masuk | 602 |
| siswa | 148 |
| bijak | 140 |
| anak | 124 |
| gubernur | 122 |
| sma | 98 |

# BAB V

# PENUTUP

## Kesimpulan

Berdasarkan apa yang telah ditemukan pada pengujian algoritma Naive Bayes dalam analisis sentimen terhadap kebijakan masuk sekolah jam lima pagi di NTT, berikut ini beberapa hasil yang diperoleh, diantaranya:

1. Dalam penelitian ini, ditemukan bahwa algoritma Naïve Bayes berhasil mencapai tingkat akurasi yang tinggi mencapai 97%.
2. Berdasarkan hasil perhitungan evaluasi model dalam penelitian ini, didapatkan bahwa presisi untuk sentimen negatif mencapai 98%, sedangkan presisi untuk sentimen positif mencapai 50%. Selain itu, nilai recall untuk sentimen negatif mencapai 99%, sementara recall untuk sentimen positif mencapai 25%. Untuk f1-score, nilai untuk sentimen negatif mencapai 99%, sedangkan nilai f1-score untuk sentimen positif mencapai 33%.
3. Berdasarkan persentase pada *pie chart*, diketahui bahwa lebih dominan *tweet* yang negatif yaitu 96,91% dibandingkan *tweet* positif yaitu 3,09%. Jadi, bisa disimpulkan bahwa lebih banyak masyarakat yang tidak setuju terhadap kebijakan tersebut.

## Saran

Berdasarkan temuan penelitian ini, masih ada beberapa aspek yang perlu diperhatikan. Karena itu, peneliti berharap agar penelitian ini dapat dikembangkan lebih lanjut. Berikut ini adalah beberapa saran yang diajukan:

1. Menggunakan algoritma analisis sentimen lainnya agar dapat membandingkan hasil klasifikasi.
2. Menggunakan rasio pembagian dataset lainnya seperti 60% data latih dan 40% data uji atau 70% data latih dan 30% data uji agar dapat membandingkan hasil klasifikasi.
3. Melakukan pengumpulan lebih banyak data terutama sentimen kelas positif untuk mencapai keseimbangan dalam jumlah data di setiap kelasnya dan meningkatkan kemampuan model dalam mengenali sentimen positif.
4. Melakukan pengembangan *interface* dari proses menguji model dan menampilkan visualisasi dari hasil performa metode yang digunakan.

# DAFTAR PUSTAKA

Abdullah, M. 2015. *Metode penelitian kuantitatif*. Yogyakarta: Aswaja Pressindo. Tersedia di: http://idr.uin-antasari.ac.id/5014/1/Metodologi Penelitian Kuantitatif.pdf (Diakses pada 12 April 2023)

Aditya, D., Mubarok, A., Kom, M., & Susanti, S. 2019. Analisis Sentimen Pada Media Sosial Twitter Menggunakan Metode Naïve Bayes Classifier (Studi Kasus: Komentar Publik Kepada Tri Indonesia). *Jurnal Informatika*, *6*(1), 1–8. http://ejournal.bsi.ac.id/ejurnal/index.php/ji/article/view/-

Aggarwal, C. C. (2015). *Data Mining: The Textbook*. Springer International Publishing Switzerland.

Artanti, D. P., Syukur, A., Prihandono, A., & Setiadi, D. R. I. M. (2018). *Analisa Sentimen Untuk Penilaian Pelayanan Situs Belanja Online Menggunakan Algoritma Naïve Bayes*. 8–9.

http://jurnal.atmaluhur.ac.id/index.php/knsi2018/article/view/353/278

Bustami, B. 2013. Penerapan Algoritma Naive Bayes Untuk Mengklasifikasi Data Nasabah Asuransi. *TECHSI-Jurnal Teknik Informatika*, *5*(2), 128–146. https://doi.org/https://doi.org/10.29103/techsi.v5i2.154

Duei Putri, D., Nama, G. F., & Sulistiono, W. E. 2022. Analisis Sentimen Kinerja Dewan Perwakilan Rakyat (DPR) Pada Twitter Menggunakan Metode Naive Bayes Classifier. *Jurnal Informatika Dan Teknik Elektro Terapan*, *10*(1), 34–40. https://doi.org/10.23960/jitet.v10i1.2262

Ernamia, E. M. A., & Herliana, A. 2022. Analisis Sentimen Kuliah Daring Dengan Algoritma Naïve Bayes, K-Nn Dan Decision Tree. *Jurnal Responsif : Riset Sains Dan Informatika*, *4*(1), 70–80. https://doi.org/10.51977/jti.v4i1.614

Fikri, M. I., Sabrila, T. S., & Azhar, Y. 2020. Perbandingan Metode Naïve Bayes dan Support Vector Machine pada Analisis Sentimen Twitter. *SMATIKA JURNAL*, *10*(02), 71–76. https://doi.org/10.32664/smatika.v10i02.455

Fink, C. R., Chou, D. S., Kopecky, J. J., & Llorens, A. J. 2011. Coarse- and Fine-Grained Sentiment Analysis of Social Media Text. *Johns Hopkins Apl Technical Digest*, *30*(1), 22–30. Tersedia di:

https://secwww.jhuapl.edu/techdigest/content/techdigest/pdf/V30-N01/30-01-Fink.pdf (Diakses pada 14 April 2023)

Gulo, P. C. S. 2021. Analisis Sentimen Kuliah Online Selama Pandemi Covid-19 Menggunakan Algoritma Naive Bayes. Tersedia di:

https://repository.unja.ac.id/24353/3/PIKIR CLAUDIA SEPTIANI GULO.pdf (Diakses pada 31 Maret 2023). *Skripsi.* Program Studi Sistem Informasi Jurusan Teknik Elektro dan Informatika Fakultas Sains Dan Teknologi Universitas Jambi.

Haddi, E., Liu, X., & Shi, Y. 2013. The Role of Text Pre-processing in Sentiment Analysis. *Procedia Computer Science*, *17*, 26–32.

https://doi.org/10.1016/j.procs.2013.05.005

Han, J., Kamber, M., & Pei, J. 2012. *Data Mining Concept and Techniques Third Edition*. Waltham: Morgan Kaufmann Publishers.

Hardani, dkk. 2020. *Buku Metode Penelitian Kualitatif dan Kualitatif*. In *Repository.Uinsu.Ac.Id* (Issue April). Tersedia di:

https://www.researchgate.net/profile/Hardani-Msi/publication/340021548\_Buku\_Metode\_Penelitian\_Kualitatif\_Kuantitatif/links/5e72e011299bf1571848ba20/Buku-Metode-Penelitian-Kualitatif-Kuantitatif.pdf (Diakses pada 06 April 2023)

Kaparang, S. G., Kaparang, D. R., & Rantung, V. P. 2021. Analisis Sentimen New Normal Pada Masa Covid-19 Menggunakan Algoritma Naive Bayes Classifier. *Jointer - Journal of Informatics Engineering*, *2*(01), 16–23. https://doi.org/10.53682/jointer.v2i01.33

Kirana, M. C., Perkasa, N. P., Zainuddin, M., Maidel, L., Informatika, T., & Batam, P. N. 2019. Visualisasi Kualitas Penyebaran Informasi Gempa Bumi di Indonesia Menggunakan Twitter. *Journal of Applied Informatics and Computing*, *3*(1), 23–32.

https://doi.org/https://doi.org/10.30871/jaic.v0i0.1246

Twitter. 2023. *Pusat Bantuan*. https://help.twitter.com/id (Diakses pada 01 April 2023)

# LAMPIRAN

**Lampiran 1** Source Code Proses Scraping Data

#install module

!pip install snscrape

!pip install pandas

!pip install textblob

!pip install openpyxl

!pip install git+https://github.com/JustAnotherArchivist/snscrape.git

#import module

import snscrape.modules.twitter as sntwitter

import pandas as pd

#import module

from textblob import TextBlob

import re

def clean\_tweet(tweet):

return ' '.join(re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])|(\w+:\/\/\S+)", " ", tweet).split())

#sentiment analisis dengan polarity menggunakan library textblob

def analize\_sentiment(tweet):

analysis = TextBlob(clean\_tweet(tweet))

if analysis.sentiment.polarity > 0:

return 1

# elif analysis.sentiment.polarity == 0:

# return 0

else:

return -1

def analize\_subjectorobject(tweet):

#polaritas [1],[0],[2] untuk 1 termasuk negatif,0 neutral dan 2 termasuk positif. Misal terdapat polarity 0,8, yang berarti pernyataannya positif

#dan subyektivitas 0,75 merujuk bahwa itu adalah opini publik dan bukan informasi faktual.

analysis = TextBlob(clean\_tweet(tweet))

if analysis.sentiment.polarity > 0:

return 'Positif'

# elif analysis.sentiment.polarity == 0:

# return 'Neutral'

else:

return 'Negatif'

# Tanggal mulai dan berakhir untuk rentang waktu pencarian tweet

since\_date = '2023-02-27'

until\_date = '2023-03-23'

# keywoard tweet

query = 'sekolah jam 5 di NTT since:{} until:{}'.format(since\_date, until\_date)

# menentukan jumlah data tweet berbahasa indonesia

tweets\_list = []

for i, tweet in enumerate(sntwitter.TwitterSearchScraper(query + ' lang:id').get\_items()):

if i > 800:

break

tweets\_list.append([tweet.date, tweet.id, tweet.rawContent, tweet.user.username, analize\_sentiment(tweet.rawContent)])

# Creating a dataframe from the tweets list above

tweet = pd.DataFrame(tweets\_list, columns=['Datetime', 'Tweet Id', 'Text', 'Username', 'Label'])

tweet

tweet.groupby('Label').describe()

tweet\_positif=tweet[tweet['Label'] > 0]

tweet\_positif.head()

tweet\_negatif=tweet[tweet['Label'] < 0]

tweet\_negatif.head()

frames = [tweet\_positif, tweet\_negatif]

hasil = pd.concat(frames).drop\_duplicates().reset\_index(drop=True)

hasil.to\_csv(r'HasilData\_Scraping.csv', index=False)

tweet = pd.read\_csv('HasilData\_Scraping.csv')

tweet.groupby('Label').describe()

tweet = pd.DataFrame(tweet)

tweet.to\_excel ('HasilData\_Scraping.xlsx', index = False, header=True)

**Lampiran 2** Source Code Proses Preprocessing, TF-IDF, Splitting Dataset, Klasifikasi Naïve Bayes, Evaluasi, Dan Visualisasi

#Preprocessing

#Langkah 1 Preprocessing

!pip install nltk

!pip install Sastrawi

import pandas as pd

import re

data = pd.read\_excel("HasilData\_Scraping.xlsx")

data.head()

data.shape

df = data

df = pd.read\_excel('HasilData\_Scraping.xlsx', usecols=['Text', 'Label'])

df

#Proses Cleaning & Case Folding

import string

def clean\_text(text):

# menghapus username twitter

text=re.sub('@[^\s]+', ' ', text)

# menghapus https dan http

text=re.sub(r"(?:\@|http?\://|https?\://|www)\S+", "", text)

# menghilangkan tanda baca

text = text.translate(str.maketrans(' ',' ',string.punctuation))

# mengganti karakter html dengan tanda petik

text=re.sub('<.\*?>', ' ', text)

# mempertimbangkan huruf dan angka

text = re.sub('[^a-zA-Z]',' ',text)

# mengganti line baru dengan spasi

text = re.sub("\n"," ",text)

# menghapus single char

text = re.sub(r"\b[a-zA-Z]\b"," ",text)

# memisahkan dan menggabungkan kata

text=' '.join(text.split())

# mengubab ke huruf kecil

text = text.lower()

return text

df['cleaning & case folding'] = df['Text'].apply(clean\_text)

df.head()

#Proses Tokenizing

from nltk.tokenize import RegexpTokenizer

regexp = RegexpTokenizer(r'\w+|$[0-9]+|\S+')

df['tokenizing'] = df['cleaning & case folding'].apply(regexp.tokenize)

df.head()

#Proses Normalization

normalized\_word = pd.read\_csv("Normalisasi.csv")

normalized\_word\_dict={}

for index, row in normalized\_word.iterrows():

if row [0] not in normalized\_word\_dict:

normalized\_word\_dict[row[0]] = row[1]

def normalized\_term(document):

return [normalized\_word\_dict[term] if term in normalized\_word\_dict else term for term in document]

df['normalization'] = df['tokenizing'].apply(normalized\_term)

df.head()

#Stopword removal

import nltk

nltk.download('stopwords')

from nltk.corpus import stopwords

print(set(stopwords.words('indonesian')))

#Proses Stopword Removal

stopword = stopwords.words('indonesian')

txt\_stopword = pd.read\_csv("Stopword.txt", names= ["stopwords"],

header = None)

stopword.extend(["main", "wkwk", "si", "ju", "TUUUUHHHH",

"hahaha", "nder", "gj", "aj", "nih",

"loh", "ylh", "lol", "justeru", "doang",

"tuh", "looh", "yaaaa", "kah", "ya",

"yuk", "yak", "wowowowo", "wkwkwkwk"])

stopword.extend(txt\_stopword["stopwords"][0].split(' '))

stopword = set(stopword)

def stopwords(text):

return [word for word in text if word not in stopword]

df['stopword removal'] = df['normalization'].apply(stopwords)

df.head()

#Proses Stemming

! pip install swifter

from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

factory = StemmerFactory()

stemmer = factory.create\_stemmer()

def stemmed\_wrapper(term):

return stemmer.stem(term)

term\_dict = {}

for doc in df['stopword removal']:

for term in doc:

if term not in term\_dict:

term\_dict[term] = ' '

print(len(term\_dict))

print("=================================")

for term in term\_dict:

term\_dict[term] = stemmed\_wrapper(term)

print(term, ":", term\_dict[term])

print(len(term\_dict))

print("=================================")

def get\_stemmed\_term(doc):

return [term\_dict[term] for term in doc]

df['stemming'] = df['stopword removal'].apply(get\_stemmed\_term)

df.head()

df.to\_excel('Hasil\_Processing.xlsx', index=False, header=True)

Hasil\_Processing = pd.read\_excel('Hasil\_Processing.xlsx')

Hasil\_Processing.head()

df = pd.read\_excel('Hasil\_Processing.xlsx', usecols=['Label', 'stemming'])

df

#Pembobotan TF-IDF

#Menentukan Indeks Kata

from sklearn.feature\_extraction.text import CountVectorizer

a = len('document')

document = df['stemming']

vectorizer = CountVectorizer()

vectorizer.fit(document)

print('Vocabulary', vectorizer.vocabulary\_)

vector = vectorizer.transform(document)

print('Encoded Document is:')

print(vector.toarray())

#Menghitung Term Frequency

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.feature\_extraction.text import TfidfTransformer

tvec = CountVectorizer()

x\_cVec = tvec.fit\_transform(df['stemming'].values.astype('U'))

print(x\_cVec)

h\_tfidf = TfidfTransformer()

x\_tfidf = h\_tfidf.fit\_transform(x\_cVec)

#Menghitung TF-IDF

a = vectorizer.get\_feature\_names\_out()

tfidf = TfidfTransformer(use\_idf=True, norm='l2', smooth\_idf=True)

tf = tfidf.fit\_transform(vectorizer.fit\_transform(document)).toarray()

hasil = pd.DataFrame(data=tf, index=list(range(1, len(tf[:,1])+1 )),columns=[a])

hasil

# Dokumen yang akan dihitung DF

document = ['protes', 'orang', 'tua', 'murid', 'masuk', 'sekolah', 'jam', 'pagi', 'ntt']

# Inisialisasi objek CountVectorizer

vectorizer = CountVectorizer()

# Melakukan transformasi untuk menghitung Term Frequency (TF)

tf\_matrix = vectorizer.fit\_transform(df['stemming'])

# Menghitung nilai DF

df\_values = tf\_matrix.astype(bool).sum(axis=0)

# Membuat DataFrame hasil DF

df\_result = pd.DataFrame(df\_values.A1, index=vectorizer.get\_feature\_names\_out(), columns=['DF'])

# Memfilter hasil DF sesuai dengan dokumen yang diinginkan

df\_result = df\_result[df\_result.index.isin(document)]

# Menampilkan hasil DF

print("\nDF:")

print(df\_result)

#Proses Splitting Dataset

from sklearn.model\_selection import train\_test\_split

x\_train, x\_test, y\_train, y\_test = train\_test\_split(df['stemming'], df['Label'],

test\_size=0.20, random\_state=42)

#Hasil Splitting Dataset

print("x\_train\_shape : ",x\_train.shape)

print("x\_test\_shape : ",x\_test.shape)

print("y\_train\_shape : ",y\_train.shape)

print("y\_test\_shape : ",y\_test.shape)

import pandas as pd

# Membuat DataFrame dari data hasil splitting dataset

df\_train = pd.DataFrame({'x\_train': x\_train, 'y\_train': y\_train})

df\_test = pd.DataFrame({'x\_test': x\_test, 'y\_test': y\_test})

# Simpan data pelatihan ke dalam file Excel

df\_train.to\_excel('data\_pelatihan.xlsx', index=False)

# Simpan data uji ke dalam file Excel

df\_test.to\_excel('data\_uji.xlsx', index=False)

#Klasifikasi Naïve Bayes

x\_train = vectorizer.transform(x\_train)

x\_test = vectorizer.transform(x\_test)

#Proses Klasifikasi Naive Bayes

from sklearn.naive\_bayes import MultinomialNB

model\_naive = MultinomialNB().fit(x\_train, y\_train)

predicted\_naive = model\_naive.predict(x\_test)

import pandas as pd

from sklearn.feature\_extraction.text import CountVectorizer

# Baca data pelatihan dari file Excel

df\_train = pd.read\_excel('data\_pelatihan.xlsx')

# Filter data yang memiliki label 1

df\_label\_1 = df\_train[df\_train['y\_train'] == 1]

# Ambil kolom teks (x\_train) pada data dengan label 1

texts = df\_label\_1['x\_train'].tolist()

# Inisialisasi CountVectorizer

vectorizer = CountVectorizer()

# Hitung kemunculan term pada seluruh dokumen

term\_counts = vectorizer.fit\_transform(texts)

# Ambil daftar term

terms = vectorizer.get\_feature\_names\_out()

# Hitung kemunculan term yang diinginkan

term\_to\_count = {}

target\_terms = ['siswa', 'ntt', 'masuk', 'sekolah', 'jam', 'pagi', 'orang', 'dpr', 'heran', 'anakanak', 'jadi', 'kelinci', 'coba']

for term in target\_terms:

term\_index = vectorizer.vocabulary\_.get(term)

if term\_index is not None:

term\_count = term\_counts[:, term\_index].sum()

term\_to\_count[term] = term\_count

# Print hasil perhitungan

for term, count in term\_to\_count.items():

print(f"Kemunculan term '{term}': {count}")

import pandas as pd

# Membaca file data\_pelatihan.xlsx

df\_train = pd.read\_excel('data\_pelatihan.xlsx')

# Menghitung jumlah term berlabel positif

term\_positif = df\_train[df\_train['y\_train'] == 1]['x\_train'].str.split().explode().value\_counts()

# Menghitung jumlah term berlabel negatif

term\_negatif = df\_train[df\_train['y\_train'] == -1]['x\_train'].str.split().explode().value\_counts()

# Menghitung total jumlah seluruh term positif

total\_term\_positif = term\_positif.sum()

# Menghitung total jumlah seluruh term negatif

total\_term\_negatif = term\_negatif.sum()

print("Total jumlah seluruh term positif:", total\_term\_positif)

print("Total jumlah seluruh term negatif:", total\_term\_negatif)

#Evaluasi Model

#Proses Evaluasi Model

from sklearn.metrics import accuracy\_score, precision\_score, recall\_score, f1\_score

from sklearn.metrics import classification\_report

from sklearn.metrics import confusion\_matrix

clf = MultinomialNB()

clf.fit(x\_train, y\_train)

predicted = clf.predict(x\_test)

print(f'confusion matrix:\n {confusion\_matrix(y\_test, predicted)}')

print('===================================================\n')

print(classification\_report(y\_test, predicted, zero\_division=0))

#Visualisasi Confusion Matrix

import matplotlib.pyplot as plt

import seaborn as sns

# Definisikan confusion matrix dan label

confusion\_mat = [[151, 1],

[3, 1]]

labels = ['-1', '1']

# Buat heatmap confusion matrix

sns.heatmap(confusion\_mat, annot=True, fmt='d', cmap='Blues', xticklabels=labels, yticklabels=labels)

# Atur label sumbu x dan y

plt.xlabel('Predicted Label')

plt.ylabel('Actual Label')

# Atur judul plot

plt.title('Confusion Matrix')

# Simpan plot ke dalam file

plt.savefig('E:\KULIAH\SKRIPSI\confusion\_matrix.png')

import pandas as pd

# Membuat dataframe dari label prediksi, label aktual, dan tweet

df\_result = pd.DataFrame({'Actual Label': y\_test, 'Predicted Label': predicted, 'Tweet': x\_test})

# Mencetak dataframe ke layar

print(df\_result)

# Menyimpan dataframe ke file Excel

df\_result.to\_excel('hasil\_prediksi.xlsx', index=False)

#Visualisasi

#Pie Chart

#Menghitung Perbandingan Sentimen

hitung = df.Label.value\_counts()

hitung

#Visualisasi Pie Chart

import matplotlib.pyplot as plt

%matplotlib inline

sentiments = ['Sentimen Negatif', 'Sentimen Positif']

plt.figure(figsize=(10,7))

mycolors = ["navy", "lightblue"]

plt.pie(hitung.values, labels = sentiments, colors=mycolors, startangle=90, shadow = False,

explode = (0, 0.1), radius = 1.5, autopct = '%1.2f%%')

plt.legend()

plt.savefig("E:\KULIAH\SKRIPSI\Pie Chart")

plt.show()

#Word Cloud

! pip install wordcloud

from wordcloud import WordCloud

#Proses Word Cloud

# Menggabungkan semua kata-kata dari dataset

all\_words = ' '.join([text for text in df['stemming']])

# Membuat objek WordCloud

wordcloud = WordCloud(width=800, height=500, background\_color='white').generate(all\_words)

# Menampilkan word cloud

plt.figure(figsize=(10, 7))

plt.imshow(wordcloud, interpolation='bilinear')

plt.axis('off')

plt.title('Word Cloud - Sentimen Positif dan Negatif')

plt.savefig('E:/KULIAH/SKRIPSI/WordCloud.png')

plt.show()

from collections import Counter

# Menghitung frekuensi kata-kata

word\_freq = Counter(all\_words.split())

# Menampilkan 10 kata dengan frekuensi tertinggi

top\_words = word\_freq.most\_common(10)

for word, freq in top\_words:

print(f"Kata: {word}, Frekuensi: {freq}")

# Menampilkan jumlah kata unik

unique\_words = len(word\_freq)

print(f"Jumlah kata unik: {unique\_words}")

**LAMPIRAN 3**

**BIODATA PENELITI**

|  |  |  |
| --- | --- | --- |
| Nama Lengkap | : | **Wilhelmina Sonya Hoar** |
| NIM | : | 19083000192 |
| Tempat/TanggalLahir | : | Atambua, 26 Oktober 2001 |
| Alamat Asal | : | Jl. R Suprapto, Fatukbot, Kec. Atambua Selatan, Kab. Belu, NTT |
| Alamat di Malang | : | Jl. Kelengkeng No. 03, Bareng, Kec. Klojen |
| Nama Orang Tua | : | Ayah Nicolaus Nahak Seran dan Ibu (Alm) Sri Utami |
| Alamat Orang Tua | : | Jl. R. Suprapto, Fatukbot, Kec. Atambua Selatan, Kab. Belu, NTT |

**Pendidikan Formal:**

|  |  |  |
| --- | --- | --- |
| 2007-2013 | : | SD Katolik Motabuik |
| 2013-2016 | : | SMP Negeri 2 Atambua |
| 2016-2019 | : | SMA Negeri 1 Atambua |
| 2019-2023 | : | Program S1 Sistem Informasi Universitas Merdeka Malang |

**Pelatihan:**

|  |  |  |
| --- | --- | --- |
| 2022 | : | Peserta MSIB Studi Independen Information Worker |

|  |
| --- |
| Malang, 18 Juli 2023 |
|  |
|  |
|  |
| Wilhelmina Sonya Hoar |

**LAMPIRAN 4**



UNIVERSITAS MERDEKA MALANG

**FAKULTAS TEKNOLOGI INFORMASI** PROGRAM S1 SISTEM INFORMASI

Jl. Terusan Raya Dieng No. 62-64 Malang, Tlp. (0341) 568395 Psw 657

**BERITA ACARA BIMBINGAN SKRIPSI**

|  |  |  |  |
| --- | --- | --- | --- |
| 1. | Nama | : | Wilhelmina Sonya Hoar |
| 2. | NIM | : | 19083000192 |
| 3. | Judul Tugas Akhir | : | Analisis Sentimen Terhadap Kebijakan Masuk Sekolah Jam Lima Pagi Di Ntt Menggunakan Algoritma Naïve Bayes |
| 4. | Nama Pembimbing | : | 1. Drs. Anis Zubair, M.Kom |
| 5. | Keterangan Konsultasi | : |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Tanggal** | **Paraf Pembimbing** | **Materi Bimbingan** | **Keterangan** |
| 1. | 21-03-2023 |  | Konfirmasi dosen pembimbing dan dimintai data scraping |  |
| 2. | 25-03-2023 |  | Mengirim hasil scraping data twitter |  |
| 3. | 26-03-2023 |  | Konsultasi Judul dan persetujuan ACC judul skripsi |  |
| 4. | 14-04-2023 |  | Konsultasi Bab 1-3 dan revisi |  |
| 5. | 17-04-2023 |  | Bimbingan dan Persetujuan ACC Sempro |  |
| 6. | 03-07-2023 |  | Konsultasi Bab 4-5 |  |
| 7. | 04-07-2023 |  | Revisi Bab 4-5 |  |
| 8. | 05-07-2023 |  | Konsultasi kembali Bab 4-5 |  |
| 9. | 06-07-2023 |  | ACC Bab 4-5 dan lanjut pengecekan plagiasi |  |
| 10. | 18-07-2023 |  | Konsultasi keseluruhan laporan skripsi dan persetujuan ACC Semhas |  |

|  |
| --- |
| Malang, |
| Mengetahui |
| Ketua Program Studi S1 Sistem Informasi, |
|  |
|  |
| Galandaru Swalaganata, S.Si., M.Si |

**LAMPIRAN 5**



UNIVERSITAS MERDEKA MALANG

**FAKULTAS TEKNOLOGI INFORMASI** PROGRAM S1 SISTEM INFORMASI

Jl. Terusan Raya Dieng No. 62-64 Malang, Tlp. (0341) 568395 Psw 657

**KETERANGAN PERSETUJUAN SKRIPSI**

Yang bertanda tangan di bawah ini, Tim Dosen Pembimbing Skripsi dari mahasiswa:

|  |  |  |
| --- | --- | --- |
| Nama | : | Wilhelmina Sonya Hoar |
| NIM | : | 19083000192 |
| Judul Tugas Akhir | : | Analisis Sentimen Terhadap Kebijakan Masuk Sekolah Jam Lima Pagi Di Ntt Menggunakan Algoritma Naïve Bayes |

Menyatakan bahwa skripsi mahasiswa yang bersangkutan telah selesai dan siap diujikan pada ujian akhir skripsi.

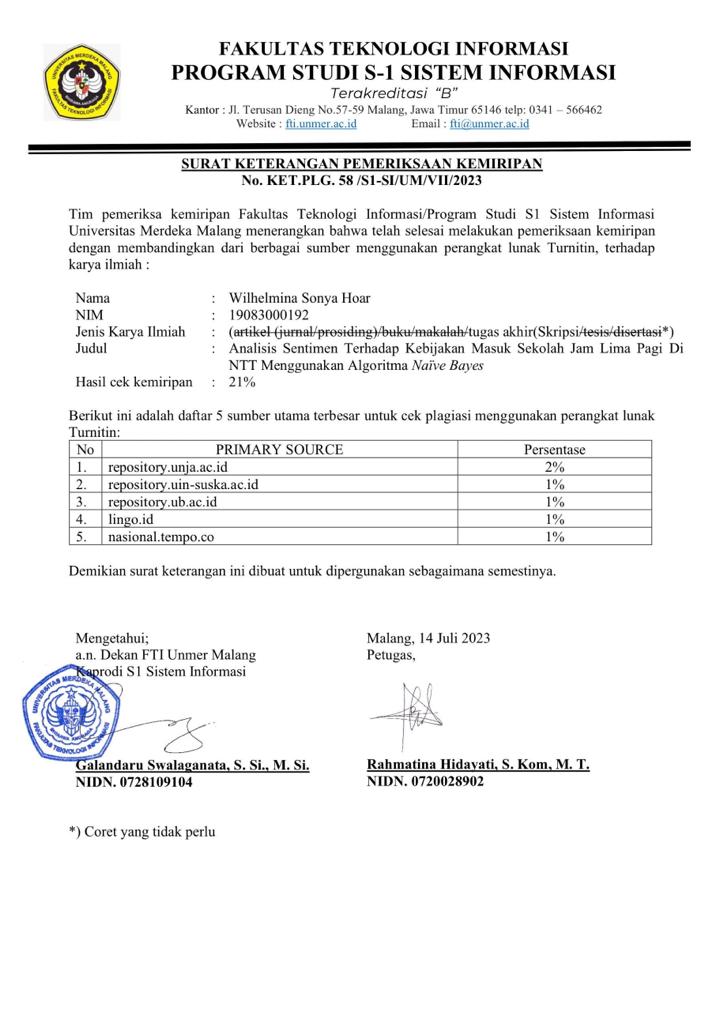
Atas kerjasamanya diucapkan terima kasih.

Malang, 18 Juli 2023

Dosen Pembimbing I,

(Drs. Anis Zubair, M.Kom )

**LAMPIRAN 6**



**LAMPIRAN 7**



UNIVERSITAS MERDEKA MALANG

**FAKULTAS TEKNOLOGI INFORMASI** PROGRAM S1 SISTEM INFORMASI

Jl. Terusan Raya Dieng No. 62-64 Malang, Tlp. (0341) 568395 Psw 657

**LEMBAR PENILAIAN UJIAN SKRIPSI**

|  |  |  |  |
| --- | --- | --- | --- |
| 1. | Nama Dosen Penguji | : | * + - 1. Rahmatina Hidayati, S.Kom., M.T. |
|  |  |  | * + - 1. Ronald David Marcus, S.Kom., M.Kom. |
| 2. | Nama Mahasiswa | : | Wilhelmina Sonya Hoar |
| 3. | NIM | : | 19083000192 |
| 4. | Judul | : | Analisis Sentimen Terhadap Kebijakan Masuk Sekolah Jam Lima Pagi Di Ntt Menggunakan Algoritma Naïve Bayes |
| 5. | Nilai | : |  |

|  |  |  |
| --- | --- | --- |
| **ASPEK YANG DINILAI** | **BOBOT** | **NILAI ANGKA** |
| **1. Isi TugasAkhir**   1. Latar Belakang dan Rumusan Masalah 2. Kaitan antara teori dengan topik tugas akhir 3. Metode Penelitian 4. Analisis dan Pembahasan | **40%** |  |
| **2. Penguasaan Materi Tugas Akhir:**   1. Kemampuan dalam menyampaikan jawaban 2. Ketepatan dalam menjawab 3. Konsistensi jawaban | **50%** |  |
| **3. Presentasi Tugas Akhir** | **10%** |  |
| **NILAI AKHIR** |  |  |

|  |  |
| --- | --- |
| **NilaiAngka** | **NilaiHuruf** |
| 80 – 100 | A |
| 76 – 79 | A- |
| 71 – 75 | B+ |
| 66 – 70 | B |
| 63 – 65 | B- |
| 60 – 62 | C+ |
| 56 – 59 | C |
| 45 – 55 | D |
| 0 – 44 | E |

|  |
| --- |
| Malang, |
| Dosen Penguji, |

**LAMPIRAN 8**



UNIVERSITAS MERDEKA MALANG

**FAKULTAS TEKNOLOGI INFORMASI** PROGRAM S1 SISTEM INFORMASI

Jl. Terusan Raya Dieng No. 62-64 Malang, Tlp. (0341) 568395 Psw 657

**BERITA ACARA UJIAN TUGAS AKHIR**

|  |  |  |
| --- | --- | --- |
| 1. Nama Mahasiswa | : | Wilhelmina Sonya Hoar |
| 2. NIM | : | 19083000192 |
| 3. Judul | : | Analisis Sentimen Terhadap Kebijakan Masuk Sekolah Jam Lima Pagi Di Ntt Menggunakan Algoritma Naïve Bayes |
| 4. Hari/Tanggal Ujian Akhir | : | Selasa, 25 Juli 2023 |
| 5. Tempat Ujian Akhir | : | Ruang 4.1 Fakultas Teknologi Informasi  Universitas Merdeka Malang |
| 6. Majelis Penguji Ujian Akhir | : |  |

|  |  |  |
| --- | --- | --- |
| **Majelis Penguji** | **Nama Penguji** | **Tanda tangan** |
| 1. Ketua 2. Sekretaris 3. Anggota | Rahmatina Hidayati, S.Kom., M.T.  Drs. Anis Zubair, M.Kom.  Ronald David Marcus, S.Kom., M.Kom. | ………  ………  ……… |
| 4. Mahasiswa | Wilhelmina Sonya Hoar | ……… |

Malang,

Mengetahui

Ketua Program Studi S1 Sistem Informasi,

Galandaru Swalaganata, S.Si., M.Si

**LAMPIRAN 9**



UNIVERSITAS MERDEKA MALANG

**FAKULTAS TEKNOLOGI INFORMASI** PROGRAM S1 SISTEM INFORMASI

Jl. Terusan Raya Dieng No. 62-64 Malang, Tlp. (0341) 568395 Psw 657

**PERBAIKAN (REVISI) SKRIPSI**

|  |  |
| --- | --- |
| 1. Nama Dosen Penguji | : Rahmatina Hidayati, S.Kom., M.T. |
| 2. Nama Mahasiswa | : Wilhelmina Sonya Hoar |
| 3. NIM | : 19083000192 |
| 4. Judul Tugas Akhir | : Analisis Sentimen Terhadap Kebijakan Masuk Sekolah Jam Lima Pagi Di Ntt Menggunakan Algoritma Naïve Bayes |
| 5. Hari/Tanggal Ujian Akhir | : Selasa, 25 Juli 2023 |
| 6. Tempat Ujian Akhir | : Ruang 4.1 Fakultas Teknologi Informasi  Universitas Merdeka Malang |

7. Keterangan Perbaikan/Revisi Skripsi:

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Bab** | **Halaman** | **Uraian** |
| 1  2 | I  I dan V | 1  3, 4, 65 | Tambahkan tanggal kebijakan dikeluarkan di Latar Belakang  Tujuan dan kesimpulan adalah jawaban dari rumusan masalah. Jika ada poin, samakan jumlahnya |

|  |  |  |
| --- | --- | --- |
| Persetujuan |  | Malang, 14 Agustus 2023 |
| Telah diperbaiki/direvisi |  | Dosen Penguji |
|  |  |  |
|  |  |  |
|  |  |  |
| (Wilhelmina Sonya Hoar) |  | (Rahmatina Hidayati, S.Kom., M.T.) |