

**METODE PEMBOBOTAN TF-IDF UNTUK KLASIFIKASI TEKS
QUICK COUNT PEMILIHAN PRESIDEN INDONESIA 2024 PADA X
TWITTER DENGAN METODE NAIVE BAYES**

SKRIPSI

**Diajukan Oleh:
Aditya Pranata
2011102441052**



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS & TEKNOLOGI
UNIVERSITAS MUHAMMADIYAH KALIMANTAN TIMUR
SAMARINDA
JULI 2024**

**METODE PEMBOBOTAN TF-IDF UNTUK KLASIFIKASI TEKS
QUICK COUNT PEMILIHAN PRESIDEN INDONESIA 2024 PADA X
TWITTER DENGAN METODE NAIVE BAYES**

SKRIPSI

Diajukan Sebagai Salah Satu Persyaratan Untuk Memperoleh Gelar Sarjana Teknik
Informatika Fakultas Sains Dan Teknologi Universitas Muhammadiyah Kalimantan Timur

**Diajukan Oleh:
Aditya Pranata
2011102441052**



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS & TEKNOLOGI
UNIVERSITAS MUHAMMADIYAH KALIMANTAN TIMUR
SAMARINDA
JULI 2024**

LEMBAR PERSETUJUAN

**METODE PEMBOBOTAN TF-IDF UNTUK KLASIFIKASI TEKS QUICK
COUNT PEMILIHAN PRESIDEN INDONESIA 2024 PADA X TWITTER
DENGAN METODE NAIVE BAYES**

SKRIPSI

**Diajukan Oleh:
Aditya Pranata
2011102441052**

**Disetujui untuk diujikan
Pada tanggal 26 Juni 2024**

Pembimbing



Rudiman, S.Kom., M.Sc

NIDN 1105068202

Mengetahui,

Koordinator Tugas Akhir/Skripsi/Tesis/Disertasi



Abdul Rahim, S.Kom., M.Cs

NIDN 0009047901

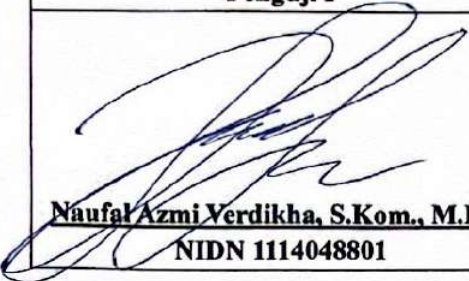

LEMBAR PENGESAHAN

**METODE PEMBOBOTAN TF-IDF UNTUK KLASIFIKASI TEKS QUICK
COUNT PEMILIHAN PRESIDEN INDONESIA 2024 PADA X TWITTER
DENGAN METODE NAIVE BAYES**

SKRIPSI

**Diajukan Oleh:
Aditya Pranata
2011102441052**

**Diseminarkan dan Diujikan
Pada Tanggal 03 Juli 2024**

Penguji I	Penguji II
 <u>Naufal Azmi Verdikha, S.Kom., M.Eng</u> NIDN 1114048801	 <u>Rudiman, S.Kom., M.Sc</u> NIDN 1105068202

Mengetahui,

Ketua

Program Studi Teknik Informatika



Arbansyah, S.Kom., M.TI
NIDN 1118019203

PERNYATAAN KEASLIAN PENELITIAN

Saya yang bertanda tangan di bawah ini:

Nama : Aditya Pranata

NIM : 2011102441052

Program Studi : S1 Teknik Informatika

Judul Penelitian : Metode Pembobotan TF-IDF untuk Klasifikasi Teks *Quick Count* Pemilihan Presiden Indonesia 2024 pada X Twitter Dengan Metode *Naive Bayes*

Menyatakan bahwa Skripsi yang saya tulis ini benar-benar hasil karya saya sendiri, dan bukan merupakan hasil plagiasi/falsifikasi/fabrikasi baik sebagian atau seluruhnya.

Atas pernyataan ini, saya siap menanggung resiko atau sanksi yang dijatuhkan kepada saya apabila kemudian ditemukan adanya pelanggaran terhadap etika keilmuan dalam skripsi saya ini, atau klaim dari pihak lain terhadap keaslian karya saya ini.

Samarinda, 15 Juli 2024

Yang membuat pernyataan



Aditya Pranata

NIM: 2011102441052

ABSTRAK

Quick Count dalam Pemilihan Presiden Indonesia 2024 memicu beragam respons masyarakat di media sosial X Twitter. Banyaknya jumlah dan variasi opini yang diekspresikan menimbulkan kesulitan dalam mengidentifikasi dan mengklasifikasikan sentimen secara akurat. Penelitian ini bertujuan mengukur akurasi metode *Naive Bayes* dengan pembobotan TF-IDF dalam mengklasifikasikan teks *Quick Count* Pemilihan Presiden Indonesia 2024 di media sosial X Twitter. Pengumpulan data dilakukan melalui *crawling*, menghasilkan 2113 tweet yang kemudian melalui proses labeling oleh ahli. Tahapan *preprocessing* meliputi *case folding*, *cleansing*, *stopword removal*, dan *stemming*. Pembobotan kata menggunakan TF-IDF, diikuti dengan pembagian data menjadi 80% data *training* dan 20% data *testing*. Klasifikasi teks menggunakan algoritma *Naive Bayes*, menghasilkan akurasi 74,46%. Hasil ini menunjukkan performa yang cukup baik dalam mengklasifikasikan sentimen terkait *Quick Count* Pemilihan Presiden 2024 di media sosial X Twitter.

Kata kunci: *Quick Count*, Klasifikasi Teks, Pembobotan TF-IDF, *Naive Bayes*, X Twitter

ABSTRACT

The Quick Count in the 2024 Indonesian Presidential Election triggered a variety of public responses on social media X Twitter. The large number and variety of opinions expressed creates difficulties in identifying and classifying sentiments accurately. This study aims to measure the accuracy of the Naive Bayes method with TF-IDF weighting in classifying the Quick Count text of the 2024 Indonesian Presidential Election on X Twitter social media. Data collection was done through crawling, resulting in 2113 tweets which then went through the labeling process by experts. Preprocessing stages include case folding, cleansing, stopword removal, and stemming. Word weighting using TF-IDF, followed by data division into 80% training data and 20% testing data. Text classification using Naive Bayes algorithm, resulting in 74.46% accuracy. This result shows a fairly good performance in classifying sentiments related to the 2024 Presidential Election Quick Count on social media X Twitter.

Keywords: Quick Count, Text Classification, TF-IDF Weighting, Naive Bayes, X Twitter

PRAKATA

Puji syukur kehadiran Allah SWT atas limpahan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul "Metode Pembobotan TF-IDF untuk Klasifikasi Teks *Quick Count* Pemilihan Presiden Indonesia 2024 pada X Twitter Dengan Metode *Naive Bayes*". Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata 1 di Program Studi Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Muhammadiyah Kalimantan Timur.

Penulis menyadari bahwa penyelesaian skripsi ini tidak terlepas dari bantuan dan dukungan berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Bapak Rudiman, S.Kom., M.Sc sebagai dosen pembimbing dan sekaligus penguji 2 yang telah memberikan arahan dan bimbingan selama proses penyusunan skripsi.
2. Bapak Naufal Azmi Verdikha, S.Kom., M.Eng selaku dosen penguji 1 yang telah memberikan masukan berharga untuk penyempurnaan skripsi ini.
3. Bapak Arbansyah, S.Kom., M.TI selaku Ketua Program Studi Teknik Informatika yang telah memberikan dukungan akademik selama masa perkuliahan.
4. Bapak Abdul Rahim, S.Kom., M.Cs selaku Koordinator Skripsi yang telah memberikan dukungan dan arahan dalam proses penyelesaian skripsi.
5. Bapak Sayekti Harits Suryawan, S.Kom, M.Kom selaku pembimbing akademik yang telah memberikan bimbingan dan arahan selama masa perkuliahan.
6. Kedua orang tua dan keluarga penulis atas dukungan moral dan material selama proses penyelesaian skripsi.
7. Saudara kandung penulis yang telah memberikan semangat, inspirasi, dan dukungan tak terhingga selama proses pengerjaan skripsi
8. Rekan-rekan mahasiswa Program Studi Teknik Informatika atas kebersamaan dan dukungan selama masa perkuliahan hingga penyelesaian skripsi.

Penulis berharap skripsi ini dapat memberikan manfaat bagi pengembangan ilmu pengetahuan, khususnya di bidang Teknik Informatika. Kritik dan saran yang membangun sangat penulis harapkan untuk penyempurnaan karya-karya ilmiah di masa mendatang.

Samarinda, 15 Juli 2024

Penyusun,



Aditya Pranata

DAFTAR ISI

LEMBAR PERSETUJUAN.....	i
LEMBAR PENGESAHAN.....	ii
PERNYATAAN KEASLIAN PENELITIAN.....	iii
ABSTRAK	iv
ABSTRACT	v
PRAKATA.....	vi
DAFTAR ISI	vii
DAFTAR TABEL.....	ix
DAFTAR GAMBAR.....	x
DAFTAR LAMPIRAN	xi
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	4
1.3 Tujuan Penelitian.....	4
1.4 Manfaat Penelitian.....	4
BAB II METODE PENELITIAN	5
2.1 Objek Penelitian	5
2.2 Alat dan Bahan	5
2.3 Prosedur Penelitian.....	6
2.3.1 Pengumpulan Data	7
2.3.2 <i>Labeling</i> Data	8
2.3.3 <i>Pre-Processing</i>	9
2.3.4 Pembobotan Kata	10
2.3.5 <i>Split</i> Data	12
2.3.6 Klasifikasi.....	13
2.3.7 Evaluasi	15
BAB III HASIL ANALISIS DAN PEMBAHASAN	16
3.1 Pengumpulan Data	16
3.2 <i>Labeling</i> Data	17
3.3 <i>Pre-Processing</i>	18
3.3.1 <i>Case folding</i>	18
3.3.2 <i>Cleansing</i>	19
3.3.3 <i>Stopword Removal</i>	20

3.3.4	<i>Stemming</i>	21
3.4	Visualisasi.....	21
3.5	Pembobotan Kata	22
3.6	<i>Split Data</i>	23
3.7	Klasifikasi.....	23
3.8	Evaluasi	24
BAB IV PENUTUP.....		25
4.1	Simpulan.....	25
4.2	Saran.....	25
DAFTAR RUJUKAN.....		26
RIWAYAT HIDUP.....		28
LAMPIRAN		29

DAFTAR TABEL

Tabel	Halaman
2.1 <i>Confusion matrix</i>	15
3.1 Hasil Pengumpulan Data	16
3.2 Hasil <i>Case folding</i>	18
3.3 Hasil <i>Cleansing</i>	19
3.4 Hasil <i>Stopword Removal</i>	20
3.5 Hasil <i>Stemming</i>	21

DAFTAR GAMBAR

Gambar	Halaman
2.1 Alur Penelitian	6
2.2 <i>Pre-Processing</i>	9
3.1 Tampilan Hasil Distribusi Sentimen.....	17
3.2 Tampilan Hasil <i>Word Cloud</i>	22
3.3 Tampilan Hasil Pembobotan Kata	22
3.4 Tampilan Hasil <i>Split Data</i>	23
3.5 Tampilan Hasil <i>Confusion matrix</i> Metode <i>Naive Bayes</i>	24

DAFTAR LAMPIRAN

Lampiran	Halaman
L1 CV <i>Expert Labeling</i>	29
L2 Code Pengumpulan Data.....	30
L3 Code <i>Pre-Processing</i>	31
L4 Code <i>Word Cloud</i>	32
L5 Code Pembobotan TF-IDF.....	32
L6 Code <i>Split Data</i>	33
L7 Code Klasifikasi.....	33
L8 Lembar Tindak Lanjut.....	34
L9 Surat Izin Penelitian.....	35
L10 Kartu Kendali Bimbingan.....	36

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Pemilihan umum (Pemilu) di Indonesia diselenggarakan oleh Komisi Pemilihan Umum (KPU), sebuah lembaga independen yang mengatur dan mengawasi proses pemilihan umum. KPU dan Badan Pengawas Pemilihan Umum (Bawaslu) memiliki peran penting dalam memastikan Pemilu berjalan secara langsung, umum, bebas, rahasia, jujur, dan adil (LUBER), serta terjamin kebebasannya dari intervensi kekuasaan lain untuk menjaga masa depan demokrasi negara (Hananto Widodo & Dicky Eko Prasetyo, 2021). Untuk memberikan gambaran awal cepat hasil pemilihan sebelum pengumuman resmi, *Quick Count* (Hitung Cepat) digunakan untuk memperkirakan hasil suara kandidat presiden pada sampel suara dari sejumlah TPS. Metode ini tidak menghitung seluruh suara, melainkan sebagian kecil sampel yang diperkirakan mencerminkan hasil akhir nasional yang sering menjadi topik pembicaraan Masyarakat dan media (Nugraha et al., 2019). Hasil *Quick Count* sering dianggap perkiraan awal yang cukup akurat dan penting bagi kepercayaan publik terhadap proses pemilihan, meski bukan hasil resmi, namun pengumuman hasil yang tidak sesuai harapan dapat memicu ketegangan politik, protes, atau kerusuhan (Rhima Indria Saraswati, 2020).

Media sosial memiliki dampak signifikan dalam kehidupan masyarakat dengan kemudahan berkomunikasi jarak jauh dan bersosialisasi secara virtual. Namun, media sosial juga memiliki konsekuensi seperti penyebaran bahasa sarkasme atau ungkapan kasar oleh para netizen yang menggunakan media sosial untuk mengekspresikan identitas dan eksistensi diri mereka, bahkan dengan cara yang melanggar norma kesantunan berbahasa (Andi Saadillah et al., 2023). Pada pemilihan presiden sebelumnya, penggunaan media sosial berperan penting dalam kampanye politik, di mana para kandidat menyampaikan visi dan misi mereka melalui platform media sosial (Ulfa et al., 2020). Dengan berkembangnya platform digital yang mampu mewedahi opini masyarakat, pro maupun kontra tidak hanya bermunculan di layar kaca melalui

liputan berita, tetapi juga banyak bermunculan di media sosial X Twitter dan platform digital (Zhafira et al., 2021). Media sosial seperti X Twitter telah menjadi platform penting bagi masyarakat untuk mengekspresikan pandangan dan opini terkait isu politik termasuk pemilihan presiden (Wulandari, 2024).

Urgensi penelitian terletak pada pentingnya mengklasifikasikan teks terkait *Quick Count* Pemilihan Presiden Indonesia 2024 di media sosial X Twitter untuk mengidentifikasi respons masyarakat terhadap perkembangan politik yang terjadi. Menurut artikel dari Tirto.id oleh Irfan Amin, pengguna media sosial di Indonesia sudah mencapai 167 juta orang pada tahun 2023 berdasarkan catatan dari *We Are Social*, dengan sebagian besar pengguna berasal dari kalangan muda yang mendominasi lebih dari 50 persen pemilih pada Pemilu. Ketertarikan pengguna media sosial terhadap informasi politik cukup tinggi, dengan 46,07 persen pengguna menunjukkan ketertarikan yang signifikan terhadap informasi politik (Amin, 2023).

Proses ekstraksi informasi teks di media sosial adalah *text mining*, melibatkan pemantauan tren, pola, dan pembobotan kata untuk mendapatkan informasi berharga dari teks yang luas (Sosiawan & Wibowo, 2020). Klasifikasi adalah teknik dalam *text mining* yang bertujuan mengelompokkan objek-objek dengan karakteristik serupa ke dalam beberapa kelas untuk menganalisis pendapat, penilaian, sikap, dan perasaan orang (Rizki et al., 2023). *Text mining* memiliki definisi menambang data berupa teks, bertujuan mencari kata-kata yang dapat mewakili isi dokumen sehingga dapat dilakukan analisa keterhubungan antar dokumen dengan menggunakan algoritma *machine learning*. Salah satu teknik dalam *text mining* adalah klasifikasi, yang bertujuan mengelompokkan objek-objek dengan karakteristik serupa ke dalam beberapa kelas, seperti mengkaji pendapat dan sikap terhadap teks (Nurmalasari & Ribut Yuliantoro, 2022). Metode *Naive Bayes* dan TF-IDF telah terbukti efektif dalam klasifikasi teks studi terhadap kenaikan harga bahan pokok di X Twitter, metode ini mencapai akurasi tinggi setelah melalui *preprocessing* menyeluruh (Muslimin & Lusiana, 2023).

Penelitian sebelumnya telah menerapkan pembobotan kata TF-IDF dan klasifikasi dengan algoritma *Naive Bayes classifier* pada ulasan aplikasi Ruang Guru. Hasil penelitian menunjukkan bahwa prediksi sentimen lebih dominan pada sentimen positif dengan nilai presisi 71%, recall 69%, F1-score 69%, dan akurasi 69% (Novitasari et al., 2022). Penelitian lainnya menggunakan *Naive Bayes* dan ekstraksi fitur TF-IDF, dengan hasil sentimen positif 70,20%, negatif 29,80%, dan akurasi 65,39% (Salim & Solichin, 2022). Penelitian terdahulu telah menerapkan kombinasi pembobotan TF-IDF dan klasifikasi *Naive Bayes* pada berbagai bidang. Namun, belum ada penelitian yang secara khusus menerapkan pendekatan ini untuk mengklasifikasi teks *Quick Count* Pemilihan Presiden Indonesia 2024 di X Twitter.

Penelitian ini mengadopsi serangkaian pendekatan, dimulai dengan dengan pengumpulan data melalui pencarian Twitter dan web crawling menggunakan Python. Setelah data terkumpul, dilakukan labeling data oleh ahli dalam pelabelan data, tahap *preprocessing* teks yang mencakup koreksi kata tidak standar dengan menggunakan kamus yang disusun oleh peneliti, serta penerapan pembobotan kata dengan metode *Term Frequency Inverse Document Frequency* (TF-IDF). Data yang telah diolah kemudian dibagi menjadi data pelatihan dan data pengujian. Klasifikasi teks dilakukan dengan *Naive Bayes classifier*. Evaluasi akhir dilakukan dengan mengukur akurasi yang dihasilkan. Melalui pendekatan ini, diharapkan klasifikasi teks *Quick Count* Pemilihan Presiden 2024 di X Twitter mampu membantu para pihak terkait seperti lembaga survei dan pemangku kepentingan lainnya untuk mengidentifikasi respons masyarakat terhadap perkembangan politik yang terjadi. Dengan demikian, penelitian ini memberikan kontribusi akademis dalam bidang klasifikasi teks dan memiliki dampak praktis dalam mengidentifikasi respons masyarakat di media sosial.

1.2 Rumusan Masalah

Rumusan masalah penelitian ini adalah seberapa akurat metode *Naive Bayes* dengan pembobotan TF-IDF dalam mengklasifikasikan teks *Quick Count* Pemilihan Presiden Indonesia 2024 di media sosial X Twitter?

1.3 Tujuan Penelitian

Tujuan penelitian ini adalah mengukur akurasi metode *Naive Bayes* dengan pembobotan TF-IDF dalam mengklasifikasikan teks *Quick Count* Pemilihan Presiden Indonesia 2024 di media sosial X Twitter.

1.4 Manfaat Penelitian

Manfaat Penelitian ini adalah:

1. Teoritis: Memberikan kontribusi dalam bidang text mining dan klasifikasi teks, khususnya dalam konteks politik di media sosial. Penelitian ini dapat memperkaya literatur tentang penggunaan metode TF-IDF dan Naive Bayes dalam klasifikasi teks.
2. Praktis: Membantu lembaga survei dan pemangku kepentingan lainnya mengidentifikasi respons masyarakat terkait teks *Quick Count* Pemilihan Presiden Indonesia 2024 di media sosial X Twitter, sehingga dapat merumuskan strategi komunikasi yang lebih efektif.
3. Metodologis: Menyediakan studi kasus yang dapat dijadikan referensi dalam penerapan metode TF-IDF dan Naive Bayes untuk klasifikasi teks di berbagai konteks lain, baik dalam ranah politik, pemasaran, maupun bidang lainnya yang membutuhkan analisis teks di media sosial.

BAB II METODE PENELITIAN

2.1 Objek Penelitian

Objek penelitian adalah data Twitter yang berisi cuitan *tweet* terkait *Quick Count* Pemilihan Presiden Indonesia tahun 2024. Data tersebut mencakup cuitan dari berbagai pengguna Twitter yang membahas, mengungkapkan opini, atau memberikan informasi seputar hasil sementara pemilihan presiden yang diumumkan melalui *Quick Count* oleh lembaga-lembaga survei. Data Twitter ini akan digunakan sebagai bahan utama untuk diklasifikasi lebih lanjut.

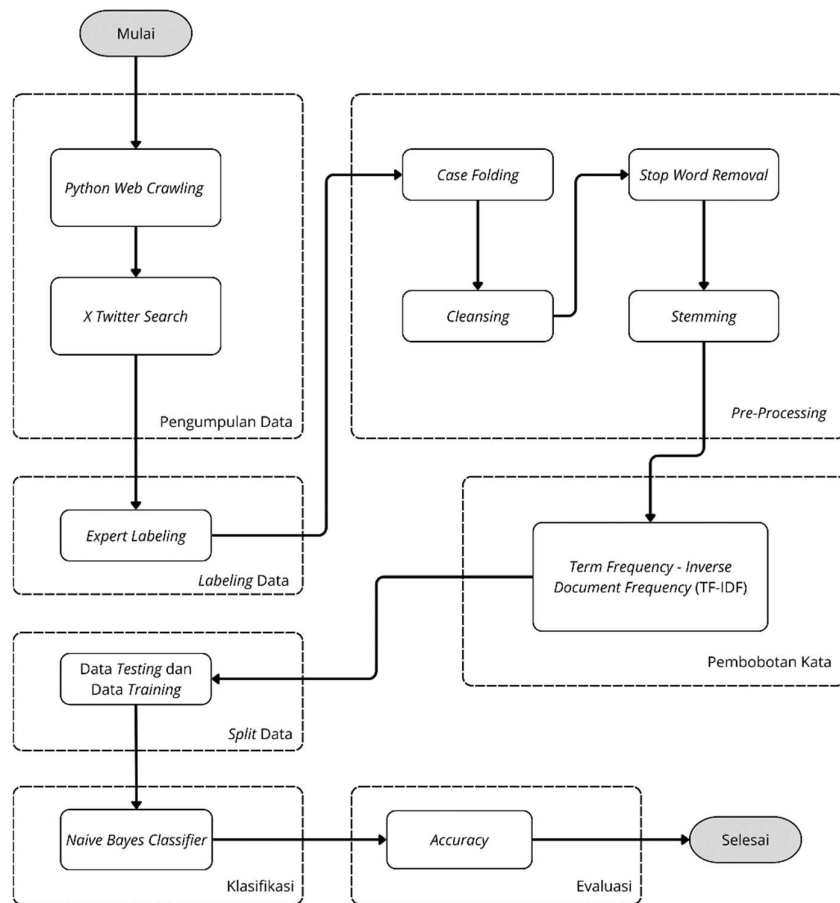
2.2 Alat dan Bahan

Dalam penelitian ini memanfaatkan berbagai alat perangkat lunak yang saling terintegrasi untuk melakukan klasifikasi teks. *Google Colab* versi 1.0.0 dipilih sebagai platform utama untuk menjalankan kode dan analisis, dengan *Python* versi 3.10.12 sebagai bahasa pemrograman dasarnya. Beberapa *library Python* yang digunakan dalam penelitian ini adalah (i) *Pandas* versi 2.0.3 untuk manipulasi dan analisis data setelah data diambil dan disimpan ke dalam file *CSV*, (ii) *NumPy* versi 1.25.2 untuk operasi numerik yang efisien, (iii) *Re* versi 2.2.1 untuk operasi pencocokan pola regex, (iv) *NLTK* versi 3.8.1 untuk pemrosesan bahasa alami, (v) *Scikit-learn* versi 1.2.2 untuk pembelajaran mesin dan analisis data, (vi) *Matplotlib* versi 3.7.1 untuk visualisasi data, (vii) *Sastrawi* versi 1.0.1 untuk stemming bahasa Indonesia. (viii) *Framework* tambahan yang digunakan adalah *Node.js* versi 14.16.0 untuk menjalankan alat *tweet-harvest*, serta *tweet-harvest* versi 2.6.0 untuk mengambil data *tweet* dari Twitter.

Bahan yang digunakan penelitian ini adalah dataset *tweet* terkait *Quick Count* pemilu di Indonesia. *Tweet* berbahasa Indonesia yang dikumpulkan pada periode 14-15 Februari 2024. Data tersebut menjadi sumber utama untuk analisis percakapan dan opini publik mengenai perhitungan cepat hasil pemilihan presiden di Indonesia.

2.3 Prosedur Penelitian

Penelitian ini mencakup tahapan dalam mengklasifikasi teks data X Twitter, dimulai dengan pengumpulan data melalui pencarian X Twitter dan *web crawling* menggunakan Python. Setelah data terkumpul, dilakukan *labeling* oleh ahli pelabelan data. Data yang telah di *labeling* kemudian melalui tahap *preprocessing* seperti *case folding*, *cleansing*, *stopwords removal*, dan *stemming*. Selanjutnya, fitur ekstraksi dilakukan dengan metode TF-IDF untuk pembobotan kata. Data yang telah diolah kemudian dibagi menjadi data pelatihan dan data pengujian. Klasifikasi teks dilakukan dengan *Naive Bayes classifier*. Kinerja model dievaluasi dengan mengukur akurasi yang dihasilkan. Gambar 2.1 menunjukkan alur penelitian secara keseluruhan.



Gambar 2.1 Alur Penelitian

2.3.1 Pengumpulan Data

Twitter menyediakan kunci antarmuka pemrograman aplikasi yang memungkinkan pengguna untuk mengakses data Twitter secara terprogram, namun dengan keterbatasan jumlah data yang dapat diambil. Pengumpulan data dilakukan menggunakan *Tweet Harvest*, sebuah alat yang memungkinkan pengguna untuk mengumpulkan lebih banyak data Twitter dapat diakses dan dijalankan melalui *Command Line Interface (CLI)* hanya dengan menggunakan *auth_token*. Proses *crawling* dilakukan dengan menentukan kata kunci pencarian, rentang tanggal pencarian, dan batas jumlah data yang akan diambil. Setelah proses *crawling* selesai, data yang berhasil diambil akan disimpan dalam format CSV untuk digunakan dalam klasifikasi selanjutnya (Darman, 2023). Tahapan pengumpulan data yang dilakukan dalam penelitian ini adalah sebagai berikut (Lampiran 2 Code Pengumpulan Data):

- a. Pada tahap pertama, token autentikasi Twitter disimpan dalam variabel *twitter_auth_token*. Token ini diperlukan untuk mengakses API Twitter dan mengambil data *tweet* yang relevan.
- b. Tahap kedua melibatkan instalasi paket *pandas* menggunakan *pip* dan *Node.js* melalui *apt-get*. Proses ini mencakup pembaruan paket, pengaturan repositori, dan verifikasi instalasi *Node.js*.
- c. Pada tahap terakhir, parameter untuk pengambilan data *tweet* ditentukan. Kata kunci pencarian yang digunakan adalah "*quick count*", dengan fokus pada *tweet* berbahasa Indonesia (*lang:id*) yang diunggah antara 14 Februari 2024 hingga 15 Februari 2024. *Tweet-harvest* kemudian dijalankan untuk mengambil data *tweet* sesuai parameter tersebut. Data yang dikumpulkan disimpan dalam format CSV, dengan jumlah maksimum *tweet* sesuai batas yang telah ditentukan.

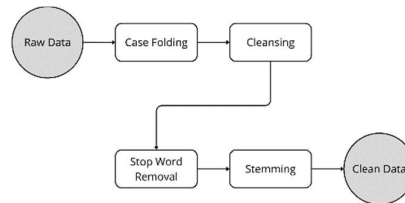
2.3.2 Labeling Data

Dalam proses melakukan Klasifikasi Teks pada data *tweet* sebagai bagian dari skripsi tugas akhir, peneliti mencari ahli dalam pelabelan data yang memiliki pengalaman dalam Labeling data dan memiliki pengetahuan mendalam tentang bahasa Indonesia. Setelah mengajukan permintaan pada website projects.co.id, peneliti berhasil mendapatkan tenaga ahli yang sesuai dengan kriteria, yaitu yang memiliki kualifikasi dan pengalaman yang relevan. Calon tenaga ahli telah memasukkan penawaran dengan mencantumkan pekerjaan saat ini, pengalaman yang relevan dalam pelabelan data, serta gelar akademik yang dimiliki. Pada lampiran CV dari ahli pelabelan data telah dicantumkan untuk memberikan informasi lebih lanjut mengenai kualifikasi dan pengalaman yang dimilikinya dalam melakukan *labeling* data. Dasar atau acuan yang digunakan oleh ahli pelabelan data dalam melakukan pelabelan pada data adalah sebagai berikut (Lampiran 1 CV *Expert Labeling*):

- a. Positif: *Quick count* telah terbukti memberikan hasil yang akurat dan dapat dipercaya dalam berbagai pemilihan. Metode ini membantu memberikan gambaran awal yang valid tentang hasil pemungutan suara, mendukung harapan masyarakat akan transparansi proses demokrasi. Banyak pihak menyambut baik pelaksanaan *Quick Count*, mengikuti perkembangannya dengan antusias, dan menggunakan hashtag *#QuickCount* untuk berbagi informasi terkini. Media pun turut menyajikan berita-berita objektif mengenai hasil *Quick Count*, membantu publik memahami situasi dengan lebih baik.
- b. Negatif: Di sisi lain, sejumlah pihak mengungkapkan keraguan terhadap metodologi dan akurasi *Quick Count*. Beberapa kritik muncul terkait potensi penggiringan opini publik atau manipulasi data yang menimbulkan perdebatan di media sosial. Sebagian masyarakat mengekspresikan kekecewaan atau kemarahan terhadap hasil *Quick Count* yang tidak sesuai ekspektasi mereka, bahkan ada yang menggunakan bahasa kasar atau sindiran untuk mengungkapkan ketidakpuasan. Keraguan juga muncul terkait objektivitas

lembaga pelaksana *Quick Count*, dengan tuduhan adanya kepentingan tertentu di balik penyelenggaraannya.

2.3.3 Pre-Processing



Gambar 2.2 *Pre-Processing*

Gambar 2.2 *Preprocessing* menunjukkan tahapan-tahapan yang diperlukan dalam proses *preprocessing* teks. *Preprocessing* adalah tahap yang krusial sebelum memulai sebuah penelitian. Text *preprocessing* dilakukan untuk memastikan bahwa data awal melewati serangkaian tahapan sehingga menjadi siap digunakan sepenuhnya dalam proses klasifikasi (Albab et al., 2023). Tahapan *preprocessing* yang dilakukan dalam penelitian ini adalah sebagai berikut (Lampiran 3 Code *Preprocessing*):

- a. Pada tahap pertama, *Case folding* merupakan tahap di mana semua huruf dalam teks diubah menjadi huruf kecil Python *string lower method*. Tujuannya adalah untuk memastikan konsistensi dalam format huruf sehingga semua teks diproses dalam bentuk huruf kecil.
- b. Pada tahap kedua, *Cleansing* pembersihan teks bertujuan untuk menghapus kata-kata atau karakter yang tidak diinginkan yang tidak relevan untuk klasifikasi teks. Ini termasuk penghapusan karakter khusus seperti URL, *emoji*, *username*, angka, dan simbol-simbol kecuali *underscore*, karena tidak memberikan kontribusi pada penilaian sentimen.
- c. Tahap ketiga adalah *Stopword Removal* tahap ini melibatkan penghapusan kata-kata umum yang tidak memberikan makna tambahan dalam klasifikasi teks. Kata-kata tersebut

dikenal sebagai *stopwords*. Penghapusan *stopwords* membantu dalam fokus pada kata-kata yang lebih informatif dan signifikan dalam menentukan sentimen.

- d. Pada tahap terakhir proses *Stemming* merupakan langkah selanjutnya yang bertujuan untuk mengubah kata-kata ke dalam bentuk dasarnya dengan menghapus imbuhan-imbuhan yang ada. Contohnya, kata-kata seperti "berlari," "berlari-lari," dan "lari" dapat diubah menjadi bentuk dasar "lari." Ini membantu dalam penyederhanaan kata-kata dengan akar yang sama untuk klasifikasi teks dengan lebih akurat.

2.3.4 Pembobotan Kata

Term Frequency-Inverse Document Frequency (TF-IDF) adalah proses penting dalam analisis data teks, di mana pembobotan dilakukan menggunakan algoritma TF-IDF untuk memberikan skor frekuensi setiap *term* dalam sebuah dokumen. *Term Frequency* (TF) menentukan seberapa sering *term* muncul dalam dokumen, sedangkan *Document Frequency* (DF) menunjukkan jumlah dokumen di mana *term* tersebut muncul (Hamka et al., 2022). Salah satu *transformer* atau metode untuk melakukan transformasi data teks menjadi representasi numerik adalah *TfidfVectorizer* yang mencakup ekstraksi fitur kata dan penghitungan frekuensi kemunculannya. Dengan demikian, *TfidfVectorizer* tidak hanya mengonversi teks tetapi juga menghitung bobot TF-IDF untuk setiap kata dalam dataset (Surya et al., 2024). Tahapan pembobotan kata yang dilakukan dalam penelitian ini adalah sebagai berikut (Lampiran 5 Code Pembobotan Kata):

- a. Pada tahap pertama, *library pandas*, *numpy*, dan *TfidfVectorizer* dari *scikit-learn* diimpor. Kemudian, file CSV hasil *preprocessing* dibaca menggunakan *pandas* dan disimpan dalam *DataFrame* *df*. Kolom '*stemming*' dari *DataFrame* *df* diambil dan disimpan dalam *list documents*, sedangkan kolom 'Sentimen' disimpan dalam *list sentiments*. Jumlah dokumen dihitung dan disimpan dalam variabel *N*.

- b. Tahap kedua, objek *TfidfVectorizer* diinisialisasi untuk melakukan pembobotan TF-IDF pada data teks. Kemudian, metode *fit_transform()* dipanggil pada objek *vectorizer* dengan menggunakan data teks *documents* sebagai input. Hasilnya disimpan dalam *tfidf_matrix*, yang merupakan matriks TF-IDF untuk setiap dokumen.
- c. Tahap ketiga melibatkan konversi *tfidf_matrix* menjadi *DataFrame pandas tfidf_df* dengan kolom berisi daftar *term* yang dihasilkan oleh *vectorizer.get_feature_names_out()*.
- d. Pada tahap keempat, hasil pembobotan TF-IDF ditampilkan. Kemudian, 40 elemen pertama dari matriks TF-IDF yang tidak nol ditampilkan, diikuti oleh tiga titik sebagai pemisah, dan terakhir 3 elemen terakhir dari matriks TF-IDF yang tidak nol ditampilkan.
- e. Tahap terakhir kode untuk menyimpan hasil TF-IDF ke dalam file Excel disediakan. *DataFrame tfidf_df* yang berisi hasil pembobotan TF-IDF disimpan ke dalam file Excel tanpa menyertakan indeks.

Nilai ekstraksi fitur dihitung menggunakan Persamaan (2.1) sebagai berikut:

$$TF - IDF(t, d) = TF(t, d) * IDF(t) \quad (2.1)$$

Keterangan:

- a. $TF(t, d)$ adalah *term frequency* dari kata t dalam dokumen d , yang dihitung sebagai berikut:

$$TF(t, d) = \frac{\text{jumlah kemunculan kata } t \text{ dalam dokumen } d}{\text{total jumlah kata dalam dokumen } d} \quad (2.2)$$

- b. $IDF(t)$ adalah *inverse document frequency* dari kata t , yang dihitung sebagai berikut:

$$IDF(t) = \log \frac{\text{total jumlah dokumen dalam koleksi}}{\text{jumlah dokumen yang mengandung kata } t + 1} \quad (2.3)$$

2.3.5 *Split Data*

Split Data adalah proses membagi dataset yang diperoleh menjadi Ada dua komponen utama, yakni *training* data dan *testing* data. Data pelatihan digunakan untuk melatih model klasifikasi *Naive Bayes* agar dapat mempelajari pola dan karakteristik data. Sedangkan data uji digunakan untuk menilai performa model yang sudah dilatih pada data baru yang belum pernah diproses sebelumnya (Putri et al., 2023). Dalam penelitian tentang klasifikasi komentar toxic menggunakan TF-IDF dan Naive Bayes, rasio 80:20 menghasilkan model dengan akurasi tertinggi. Rasio 80:20 dipilih karena memberikan keseimbangan yang optimal antara jumlah data yang cukup untuk melatih model dan data yang cukup untuk menguji model secara akurat. Hal ini menunjukkan bahwa rasio 80:20 memberikan hasil yang optimal dalam menjaga keseimbangan antara data pelatihan dan pengujian sehingga model dapat dievaluasi dengan baik pada data yang belum pernah dilihat sebelumnya (Sidiq et al., 2020). Tahapan *split* data yang dilakukan dalam penelitian ini adalah sebagai berikut (Lampiran 6 Code *Split Data*):

- a. Pada tahap pertama, fungsi-fungsi yang diperlukan diimpor dari *library scikit-learn* dan *matplotlib*. Fungsi *train_test_split* dari *scikit-learn* digunakan untuk membagi dataset menjadi data latih dan data uji, sementara *matplotlib* digunakan untuk membuat visualisasi grafik.
- b. Tahap kedua melibatkan pemrosesan data dan pembagian dataset, di mana data teks diubah menjadi representasi numerik dan kemudian dataset dibagi menjadi data latih dan data uji dengan rasio 80:20. *Random_state=42* digunakan untuk menetapkan *seed* generator angka acak, menjamin pembagian data yang konsisten dan hasil yang dapat direproduksi setiap kali kode dijalankan.
- c. Pada tahap ketiga, jumlah sampel untuk total dataset, data latih, dan data uji dihitung. Selanjutnya, sebuah visualisasi *bar chart* dibuat menggunakan *matplotlib* untuk menampilkan distribusi sampel dalam dataset.

- d. Tahap keempat melibatkan penyempurnaan visualisasi. Jumlah sampel ditambahkan di atas setiap *bar chart*, label sumbu y dan judul grafik ditambahkan, serta persentase untuk data latih dan data uji ditampilkan pada bar masing-masing.
- e. Pada tahap terakhir, *layout* grafik diatur menggunakan *plt.tight_layout()* untuk memastikan semua elemen grafik terlihat dengan jelas dan tidak tumpang tindih. Grafik ditampilkan menggunakan fungsi *plt.show()*.

2.3.6 Klasifikasi

Klasifikasi *Naive Bayes* adalah metode klasifikasi yang mengandalkan *Teorema Bayes*, yang awalnya diusulkan oleh ilmuwan Inggris bernama *Thomas Bayes*. Metode ini memanfaatkan pendekatan probabilitas dan statistik untuk memproyeksikan kemungkinan kejadian di masa depan berdasarkan pengalaman masa lalu, sehingga sering disebut sebagai *Teorema Bayes*. *Naive Bayes classifier* menyediakan kemudahan bagi penggunanya seperti prosesnya yang cepat, mudah diterapkan, cukup sederhana dalam strukturnya dan sangat efektif (Sriani, Suhardi, 2024). Salah satu pengembangan dari algoritma klasifikasi *Naive Bayes* adalah *Multinomial Naive Bayes* yaitu model dimana kelas tidak hanya ditentukan dengan kata yang muncul, tapi juga dengan jumlah kemunculannya sehingga cocok untuk klasifikasi teks atau dokumen (Husada & Toba, 2020). *Teorema Bayes* dirumuskan dalam Persamaan (2.4) sebagai berikut:

$$P(C|X) = \frac{P(X|C) \cdot P(C)}{P(X)} \quad (2.4)$$

Keterangan:

- a. X: Data yang kelasnya belum diketahui
- b. C: Hipotesis bahwa data X merupakan kelas tertentu
- c. P(C|X): Probabilitas hipotesis C benar jika diberikan data X (*probabilitas posterior*)
- d. P(C): Probabilitas awal hipotesis C sebelum melihat data X (*probabilitas prior*)

- e. $P(X|C)$: Probabilitas munculnya data X jika hipotesis C benar (*likelihood*)
- f. $P(X)$: Probabilitas kemunculan data X secara umum (*probabilitas evidence*)

$P(C|X)$ adalah probabilitas bahwa suatu data X termasuk ke dalam kelas C , yang dihitung pada probabilitas prior $P(C)$, likelihood $P(X|C)$, dan probabilitas evidence $P(X)$. Semakin tinggi nilai $P(C|X)$, semakin besar kemungkinan data X termasuk ke dalam kelas C .

Confusion matrix adalah alat yang digunakan untuk mengukur performa model klasifikasi. *Confusion matrix* memberikan gambaran jumlah prediksi yang akurat dan tidak akurat yang dibuat oleh model (Fikri et al., 2020). Tahapan evaluasi yang dilakukan dalam penelitian ini adalah sebagai berikut (Lampiran 7 Code Klasifikasi):

- a. Pada tahap pertama, fungsi-fungsi yang diperlukan diimpor dari *library scikit-learn*, *seaborn*, dan *matplotlib*. Ini mencakup metrik evaluasi seperti *classification_report*, *confusion_matrix*, dan *accuracy_score*, serta *MultinomialNB* untuk *Naive Bayes*.
- b. Tahap kedua melibatkan inisialisasi objek klasifikasi *Naive Bayes Multinomial* menggunakan *MultinomialNB()*.
- c. Pada tahap ketiga, metode *fit* digunakan untuk melatih model dengan data latih (*X_train*, *y_train*).
- d. Tahap keempat melibatkan penggunaan metode *predict* untuk melakukan prediksi pada data uji (*X_test*) menggunakan model yang telah dilatih sebelumnya.
- e. Pada tahap kelima, *confusion matrix* dihitung menggunakan fungsi *confusion_matrix*, dengan label sebenarnya (*y_test*) dan label prediksi sebagai input.
- f. Tahap terakhir melibatkan visualisasi *confusion matrix* menggunakan *seaborn. Heatmap* dibuat dengan label yang sesuai untuk sumbu x dan y , serta judul plot.

Struktur *Confusion matrix* untuk klasifikasi biner ditunjukkan pada Tabel 2.1 sebagai berikut:

Tabel 2.1 *Confusion matrix*

		<i>Predicted Class</i>	
		<i>Positive</i>	<i>Negative</i>
<i>Actual Class</i>	<i>Positive</i>	<i>True Positive (TP)</i>	<i>False Negative (FN)</i>
	<i>Negative</i>	<i>False Positive (FP)</i>	<i>True Negative (TN)</i>

Keterangan:

- True Positive (TP)*: Jumlah data positif yang diprediksi dengan akurat sebagai positif.
- False Positive (FP)*: Jumlah data negatif yang keliru diprediksi sebagai positif.
- False Negative (FN)*: Jumlah data positif yang keliru diprediksi sebagai negatif.
- True Negative (TN)*: Jumlah data negatif yang diprediksi dengan tepat sebagai negatif.

2.3.7 Evaluasi

Akurasi adalah cara untuk mengukur seberapa tepat suatu model dalam membuat prediksi. Metode ini membandingkan seberapa dekat hasil prediksi dengan nilai yang sebenarnya untuk keseluruhan data (Farokhah, 2020). Akurasi model dihitung dengan Persamaan (2.5) sebagai berikut:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (2.5)$$

BAB III HASIL ANALISIS DAN PEMBAHASAN

3.1 Pengumpulan Data

Proses pengumpulan data untuk penelitian ini menghasilkan dataset yang komprehensif terkait diskusi *Quick Count* Pemilihan Presiden Indonesia 2024 di platform X Twitter. Menggunakan *Tweet Harvest*, berhasil dikumpulkan total 2113 *tweet* yang relevan dalam rentang waktu 14-15 Februari 2024. Proses *crawling* data berlangsung selama 29 menit pada 21 April 2024, dari pukul 21:11 WITA hingga 21:40 WITA. Dari sampel data yang ditampilkan pada Tabel 3.1, terlihat keragaman sudut pandang dan topik diskusi seputar *Quick Count*, termasuk pembahasan hasil sementara dan pertanyaan tentang pengaruh Quick Count terhadap proses pemilu selanjutnya.

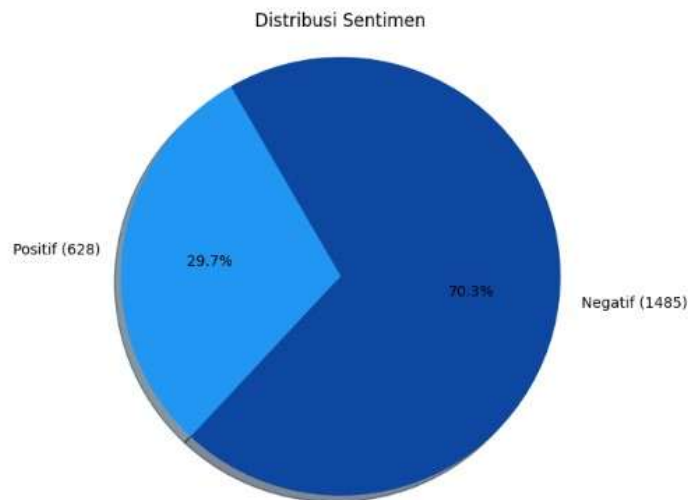
Tabel 3.1 Hasil Pengumpulan Data

No	created_at	favorite_count	full_text	...	username
1	Wed Feb 14 22:24:25 +0000 2024	0	@m4retha Bilas muka gosok gigi liat quick count lagi	...	irfanalawiii
2	Wed Feb 14 22:24:19 +0000 2024	0	@nvtamarsaoly @convomfs Bener bgt. Ga jauh2 hasil real count nanti sama quick count. Udah nder capee ngomong sama yang ga mau kalah sipaling dicurangi dan tersakiti	jisookimjichuu
3	Wed Feb 14 22:24:10 +0000 2024	1	Yg udh liat dirty vote masa percaya ma Quick Count .. Ayokk semangat para pejuang amin Real count masih berjalan	...	MantanKamu1993
...
2111	Wed Feb 14 16:29:27 +0000 2024	0	@andre_rosiade @prabowo @gibran_tweet Quick count hasil resmi kah? Lgsg lantik aja besok abis subuh	...	alexandersommy4
2112	Wed Feb 14 16:29:26 +0000 2024	1	@tanyarlfs Perasaan itu bukan pidato kemenangan deh tadi gw nonton fyi aja itu tuh emang udah direncanain kali acaranya buat ngelihat hasil quick count bareng timsesnya plis deh jangan kemakan judul berita minimal tonton sampe habis	...	ourrbee
2113	Wed Feb 14 16:29:25 +0000 2024	0	@m4retha Update 23.29: 02 unggul dengan suara diatas 50% pada 29 provinsi. Total quickcount di 38 provinsi + Luar Negeri: 01 Amin 32 50% 02 Pagi 51 21% 03 Gama 16 29%	...	msmrrleia

Data yang terkumpul sebanyak 2113 *tweet* dengan berbagai variasi konten yang berhubungan dengan *Quick Count*, di mana hanya bagian *full_text* yang diambil dari setiap *tweet*. Tahap selanjutnya data tersebut akan melalui proses *labeling* oleh seorang ahli pelabelan data yang memiliki kualifikasi dan pengalaman relevan.

3.2 Labeling Data

Dalam proses *labeling* data untuk klasifikasi teks *tweet*, *labeling* dilakukan oleh ahli yang memiliki kualifikasi dan pengalaman relevan. Ahli pelabelan menggunakan acuan yang jelas untuk mengkategorikan teks *tweet* ke dalam kelas-kelas yang telah ditentukan. Visualisasi distribusi sentimen pada Gambar 3.1 menunjukkan dataset terdiri dari 2113 *tweet* total, dengan 1485 *tweet* (70.3%) berlabel negatif dan 628 *tweet* (29.7%) berlabel positif. Distribusi ini mengindikasikan ketidakseimbangan kelas yang signifikan, dimana sentimen negatif jauh lebih dominan. Meskipun terdapat ketidakseimbangan besar, jumlah data ini masih cukup representatif untuk klasifikasi teks terkait topik Quick Count Pemilihan Presiden Indonesia 2024 pada platform X Twitter. Hasil proses *labeling* data dapat dilihat pada Gambar 3.1, yang menampilkan distribusi sentimen dalam bentuk pie chart.



Gambar 3.1 Tampilan Hasil Distribusi Sentimen

3.3 Pre-Processing

Preprocessing adalah tahap yang krusial dalam penelitian untuk memastikan data teks siap digunakan dalam proses klasifikasi teks. Berikut adalah masing-masing tahapan *preprocessing* yang telah dilakukan:

3.3.1 Case folding

Pada tahap ini, seluruh teks dalam kolom *full_text* diubah menjadi huruf kecil menggunakan fungsi *case_folding*. Hasilnya dapat dilihat pada kolom '*case_folding*' di output. Proses ini berhasil mengkonversi semua karakter menjadi huruf kecil, memastikan konsistensi format teks untuk tahapan selanjutnya. Berikut adalah tabel hasil dari tahap *case_folding* pada beberapa contoh teks:

Tabel 3.2 Hasil *Case folding*

No	full_text	case_folding
1	@m4retha Bilas muka gosok gigi liat quick count lagi	@m4retha bilas muka gosok gigi liat quick count lagi
2	@nvtamarsaoly @convomfs Bener bgt. Ga jauh2 hasil real count nanti sama quick count. Udah nder capee ngomong sama yang ga mau kalah sipaling dicurangi dan tersakiti .	@nvtamarsaoly @convomfs bener bgt. ga jauh2 hasil real count nanti sama quick count. udah nder capee ngomong sama yang ga mau kalah sipaling dicurangi dan tersakiti .
3	Yg udh liat dirty vote masa percaya ma Quick Count .. Ayokk semangat para pejuang amin Real count masih berjalan	yg udh liat dirty vote masa percaya ma quick count .. ayokk semangat para pejuang amin real count masih berjalan
...
2111	@andre_rosiade @prabowo @gibran_tweet Quick count hasil resmi kah? Lgsg lantik aja besok abis subuh	@andre_rosiade @prabowo @gibran_tweet quick count hasil resmi kah? lgsg lantik aja besok abis subuh
2112	@tanyarlfs Perasaan itu bukan pidato kemenangan deh tadi gw nonton fyi aja itu tuh emang udah direncanain kali acaranya buat ngelihat hasil quick count bareng timsesnya plis deh jangan kemakan judul berita minimal tonton sampe habis	@tanyarlfs perasaan itu bukan pidato kemenangan deh tadi gw nonton fyi aja itu tuh emang udah direncanain kali acaranya buat ngelihat hasil quick count bareng timsesnya plis deh jangan kemakan judul berita minimal tonton sampe habis
2113	@m4retha Update 23.29: 02 unggul dengan suara diatas 50% pada 29 provinsi. Total quickcount di 38 provinsi + Luar Negeri: 01 Amin 32 50% 02 Pagi 51 21% 03 Gama 16 29%	@m4retha update 23.29: 02 unggul dengan suara diatas 50% pada 29 provinsi. total quickcount di 38 provinsi + luar negeri: 01 amin 32 50% 02 pagi 51 21% 03 gama 16 29%

Proses case folding berhasil mengubah seluruh teks menjadi huruf kecil, yang penting untuk menyeragamkan format teks. Hal ini membantu mengurangi dimensi data dan meningkatkan konsistensi.

3.3.2 *Cleansing*

Tahap pembersihan teks dilakukan menggunakan fungsi *clean_text* yang menghapus berbagai elemen tidak penting. Hasil *cleansing* terlihat pada kolom '*cleansing*', di mana teks menjadi lebih bersih dan hanya berisi kata-kata yang relevan untuk proses analisis. Berikut adalah tabel hasil dari tahap *cleansing* pada beberapa contoh teks:

Tabel 3.3 Hasil *Cleansing*

No	case_folding	cleansing
1	@m4retha bilas muka gosok gigi liat quick count lagi	bilas muka gosok gigi liat quick count lagi
2	@nvtamarsaoly @convomfs bener bgt. ga jauh2 hasil real count nanti sama quick count. udah nder capee ngomong sama yang ga mau kalah sipaling dicurangi dan tersakiti . yg udh liat dirty vote masa percaya ma quick count .. ayokk semangat para pejuang amin real count masih berjalan	bener bgt ga jauh hasil real count nanti sama quick count udah nder capee ngomong sama yang ga mau kalah sipaling dicurangi dan tersakiti yg udh liat dirty vote masa percaya ma quick count ayokk semangat para pejuang amin real count masih berjalan
...
2111	@andre_rosiade @prabowo @gibrantweet quick count hasil resmi kah? lgsg lantik aja besok abis subuh	quick count hasil resmi kah lgsg lantik aja besok abis subuh
2112	@tanyarlfe perasaan itu bukan pidato kemenangan deh tadi gw nonton fyi aja itu tuh emang udah direncanain kali acaranya buat ngelihat hasil quick count bareng timsesnya plis deh jangan kemakan judul berita minimal tonton sampe habis	perasaan itu bukan pidato kemenangan deh tadi gw nonton fyi aja itu tuh emang udah direncanain kali acaranya buat ngelihat hasil quick count bareng timsesnya plis deh jangan kemakan judul berita minimal tonton sampe habis
2113	@m4retha update 23.29: 02 unggul dengan suara diatas 50% pada 29 provinsi. total quickcount di 38 provinsi + luar negeri: 01 amin 32 50% 02 pagi 51 21% 03 gama 16 29%	update unggul dengan suara diatas pada provinsi total quickcount di provinsi luar negeri amin pagi gama

Proses *cleansing* berhasil menghapus elemen-elemen yang tidak penting seperti username Twitter, tanda baca, dan karakter khusus. Ini membantu mengurangi noise dalam data dan memfokuskan analisis pada konten teks yang substansial.

3.3.3 Stopword Removal

Penghapusan *stopwords* dilakukan menggunakan fungsi *remove_stopwords*, yang menghilangkan kata-kata umum yang tidak signifikan dalam analisis. Daftar *stopwords* yang digunakan mencakup kata-kata baku bahasa Indonesia serta tambahan kata-kata informal. Hasil proses ini terlihat pada kolom '*stopword_removal*', di mana kata-kata yang tersisa adalah kata-kata yang lebih bermakna untuk analisis. Berikut adalah tabel hasil dari tahap *stopword removal* pada beberapa contoh teks:

Tabel 3.4 Hasil *Stopword Removal*

No	cleansing	stopword_removal
1	bilas muka gosok gigi liat quick count lagi	bilas muka gosok gigi liat quick count
2	bener bgt ga jauh hasil real count nanti sama quick count udah nder capee ngomong sama yang ga mau kalah sipaling dicurangi dan tersakiti	bener hasil real count quick count capee ngomong kalah sipaling dicurangi tersakiti
3	yg udh liat dirty vote masa percaya ma quick count ayokk semangat para pejuang amin real count masih berjalan	liat dirty vote percaya quick count ayokk semangat pejuang amin real count berjalan
...
2111	quick count hasil resmi kah lgsg lantik aja besok abis subuh	quick count hasil resmi lantik besok subuh
2112	perasaan itu bukan pidato kemenangan deh tadi gw nonton fyi aja itu tuh emang udah direncanain kali acaranya buat ngelihat hasil quick count bareng timsesnya plis deh jangan kemakan judul berita minimal tonton sampe habis	pidato kemenangan nonton emang direncanain acaranya hasil quick count bareng timsesnya kemakan judul berita minimal tonton sampe habis
2113	update unggul dengan suara diatas pada provinsi total quickcount di provinsi luar negeri amin pagi gama	update unggul suara diatas provinsi total quickcount provinsi negeri amin

Proses *stopword removal* berhasil menghilangkan kata-kata umum yang tidak signifikan dalam analisis, membantu mengurangi dimensi data dan memfokuskan analisis pada kata-kata yang lebih bermakna. Penggunaan daftar *stopwords* yang mencakup kata-kata baku dan informal bahasa Indonesia merupakan langkah yang baik. Namun, terdapat inkonsistensi dalam penghapusan *stopwords*. Beberapa kata informal atau slang tidak terdeteksi sebagai *stopwords*, contohnya seperti "capee" dan "sipaling" pada baris ke-2, serta "emang" pada baris ke-2112. Kata-kata ini merupakan bentuk informal yang umumnya dianggap sebagai *stopword* dan idealnya harus dihilangkan dalam proses *stopword removal*.

3.3.4 Stemming

Tahap terakhir melibatkan proses stemming menggunakan fungsi *stem_text* dengan bantuan *library Sastrawi*. Proses ini mengubah kata-kata menjadi bentuk dasarnya. Hasil stemming ditampilkan pada kolom '*stemming*', di mana kata-kata telah disederhanakan ke bentuk akarnya. Ini membantu mengurangi variasi kata dan meningkatkan efektivitas analisis. Berikut adalah tabel hasil dari tahap *stemming* pada beberapa contoh teks:

Tabel 3.5 Hasil *Stemming*

No	stopword_removal	stemming
1	bilas muka gosok gigi liat quick count bener hasil real count quick count	bilas muka gosok gigi liat quick count bener hasil real count quick count
2	capee ngomong kalah sipaling dicurangi tersakiti	capee ngomong kalah sipaling curang sakiti
3	liat dirty vote percaya quick count ayokk semangat pejuang amin real count berjalan	liat dirty vote percaya quick count ayokk semangat juang amin real count jalan
...
2111	quick count hasil resmi lantik besok subuh	quick count hasil resmi lantik besok subuh
2112	pidato kemenangan nonton emang direncanain acaranya hasil quick count bareng timsesnya kemakan judul berita minimal tonton sampe habis	pidato menang nonton emang direncanain acara hasil quick count bareng timsesnya makan judul berita minimal tonton sampe habis
2113	update unggul suara diatas provinsi total quickcount provinsi negeri amin	update unggul suara atas provinsi total quickcount provinsi negeri amin

Proses stemming dengan menggunakan library Sastrawi berhasil mengubah kata-kata menjadi bentuk dasarnya, membantu mengurangi variasi kata dan meningkatkan efektivitas analisis. Meskipun demikian, terdapat beberapa kekurangan dalam hasil stemming. Beberapa kata tidak di-stemming dengan benar, seperti "direncanain" pada baris ke-2112 yang seharusnya menjadi "rencana".

3.4 Visualisasi

Visualisasi *Word Cloud* yang ditampilkan pada Gambar 3.2 memberikan gambaran yang jelas tentang kata-kata dalam setiap kategori sentimen. Untuk sentimen negatif, kata-kata yang menonjol meliputi "curang", "real", "hasil", dan "kpu", mengindikasikan fokus pada isu-isu seputar kecurangan dan hasil penghitungan suara. Sementara itu, *Word Cloud* sentimen positif

menampilkan kata-kata seperti "kpu", "suara", "hasil", dan "menang" sebagai yang paling dominan, menunjukkan kecenderungan membahas kepercayaan terhadap proses pemilu dan hasil yang menguntungkan.



Gambar 3.2 Tampilan Hasil *Word Cloud*

3.5 Pembobotan Kata

Berdasarkan hasil pembobotan TF-IDF yang ditampilkan pada gambar 3.3, setiap *term* dalam dataset telah diubah menjadi vektor numerik. Total kosakata yang dianalisis adalah 4999 *term*. Pada *output* (0, 838) yang menunjukkan bahwa pada dokumen ke-0 terdapat *term* pada *index* ke-838 dengan nilai 0,0688. Untuk *output* (2112, 146) menunjukkan bahwa pada dokumen ke-2112 terdapat *term* pada *index* ke-146 dengan nilai 0,2451. Proses ini menghasilkan matriks TF-IDF yang mencakup 2112 dokumen dan 4999 *term*.

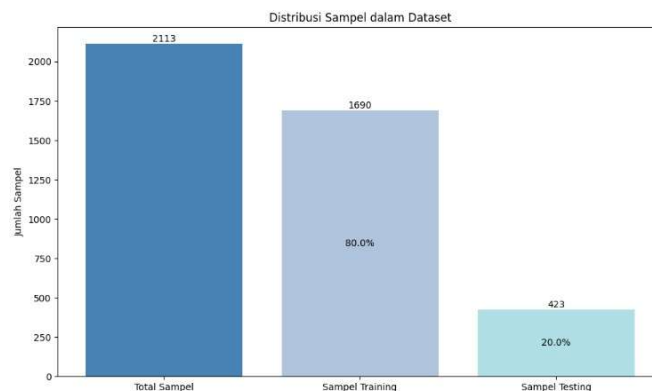
Total vocabulary: 4999			
Term	Frequency	Inverse Document	Frequency (TF-IDF)
(0, 838)	0,0688	99642091793	0,068899642091793
(0, 3669)	0,0688	99642091793	0,068899642091793
(0, 2466)	0,2385	2635668018155	0,23852635668018155
(0, 1458)	0,4843	5863847651	0,48435863847651
(0, 1494)	0,4843	5863847651	0,48435863847651
(0, 2868)	0,4516	208951228151	0,4516208951228151
(0, 536)	0,5103	4483378471	0,51034483378471
(1, 3851)	0,3700	909191252937	0,3700909191252937
(4, 880)	0,2222	209442708463	0,2222209442708463
(1, 4169)	0,4658	1517195286436	0,46581517195286436
(1, 2001)	0,2259	5508952625	0,22595508952625
(1, 3087)	0,3918	136726514	0,3918136726514
(1, 716)	0,4980	85734679603	0,498085734679603
(1, 3720)	0,1733	645454710436	0,1733645454710436
(1, 1583)	0,1315	426829419828	0,1315426829419828
(4, 449)	0,2854	768395740677	0,2854768395740677
(1, 838)	0,1324	6662346017476	0,13246662346017476
(1, 3669)	0,0662	33117900778	0,066233117900778
(2, 1841)	0,3223	36947282971	0,322336947282971
(2, 146)	0,2781	786628891291	0,2781786628891291
(2, 1950)	0,2936	368474378893	0,2936368474378893
(2, 4029)	0,2851	5569615561	0,28515569615561
(4, 225)	0,4585	15558958142	0,458515558958142
(2, 3411)	0,1955	6787121704372	0,19556787121704372
(2, 4820)	0,3676	298636995403	0,3676298636995403
(2, 1854)	0,3778	78082122224	0,377878082122224
(2, 3720)	0,1718	4915883145582	0,17184915883145582
(2, 838)	0,1313	807211303153	0,1313807211303153
(2, 3669)	0,0656	40686515765	0,065640686515765
(2, 2466)	0,2273	893219790304	0,2273893219790304
(3, 1129)	0,3293	85079960921	0,329385079960921
(3, 987)	0,3821	801212629606	0,3821801212629606
(3, 2414)	0,3624	646799218665	0,3624646799218665
(3, 4851)	0,3899	2952672523816	0,38992952672523816
(3, 3461)	0,2498	6216927890243	0,24986216927890243
(3, 358)	0,2239	499746235517	0,2239499746235517
(3, 1432)	0,3821	801212629606	0,3821801212629606
(3, 718)	0,2537	856486666287	0,2537856486666287
(3, 3463)	0,2822	1857274917	0,28221857274917
(3, 3411)	0,3871	986166074691	0,3871986166074691
...
(2112, 3678)	0,1995	55539791502	0,199555539791502
(2112, 4317)	0,1547	686489904022	0,1547686489904022
(2112, 146)	0,2451	273162975515	0,2451273162975515

Gambar 3.3 Tampilan Hasil Pembobotan Kata

Matriks TF-IDF yang dihasilkan menggambarkan bobot setiap term dalam setiap dokumen. Nilai-nilai dalam matriks ini mencerminkan tingkat kepentingan suatu term dalam dokumen tertentu relatif terhadap seluruh kumpulan dokumen. Visualisasi matriks ini memungkinkan untuk melihat penyebaran bobot term di seluruh kumpulan dokumen, membantu dalam identifikasi pola dan tema utama dalam kumpulan dokumen.

3.6 *Split Data*

Proses *split* data menghasilkan pembagian dataset menjadi dua bagian utama, dengan 1690 sampel (80%) digunakan sebagai data *training* dan 423 sampel (20%) sebagai data *testing* dari total 2113 sampel. Pembagian dengan rasio 80:20 ini dipilih untuk memberikan keseimbangan optimal antara jumlah data yang cukup untuk melatih model *Naive Bayes* dan data yang memadai untuk menguji performa model secara akurat. Visualisasi *bar chart* memperlihatkan distribusi sampel dengan jelas, menampilkan perbandingan antara total sampel, sampel *training*, dan sampel *testing* beserta persentasenya. Hasil pembagian data ini secara visual direpresentasikan dalam Gambar 3.4, yang menampilkan distribusi sampel dalam dataset.

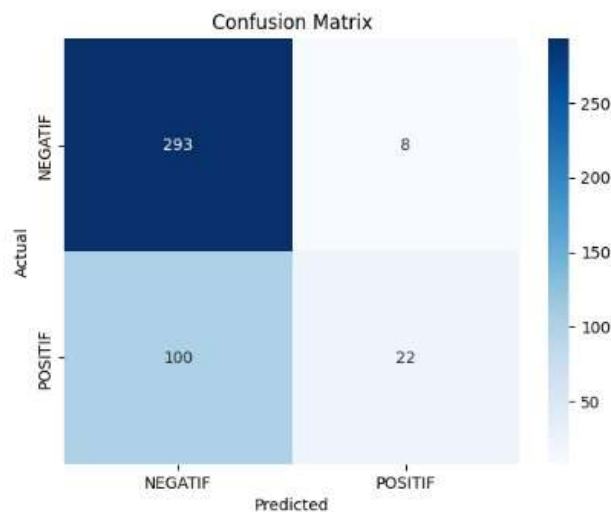


Gambar 3.4 Tampilan Hasil *Split Data*

3.7 Klasifikasi

Hasil analisis menggunakan model klasifikasi *Naive Bayes* juga ditampilkan dalam bentuk confusion matrix pada Gambar 3.5. Matriks ini memberikan gambaran detail tentang performa model dalam memprediksi sentimen. Matriks menunjukkan bahwa model berhasil

memprediksi 293 kasus sebagai *true negative* dan 22 kasus sebagai *true positive*. Namun, terdapat 100 kasus diklasifikasikan sebagai *false negative* dan 8 kasus diklasifikasikan sebagai *false positive*. Visualisasi ini membantu dalam memahami distribusi prediksi model secara lebih mendalam.



Gambar 3.5 Tampilan Hasil *Confusion matrix* Metode *Naive Bayes*

3.8 Evaluasi

Untuk memvalidasi hasil akurasi yang diperoleh dari model, dapat melakukan perhitungan manual menggunakan rumus akurasi berdasarkan data dari *confusion matrix*. Akurasi model dapat dihitung (3.1) sebagai berikut:

$$Accuracy = \frac{22 + 293}{22 + 100 + 8 + 293} = \frac{315}{423} = 0,7446 \quad (3.1)$$

Hasil akurasi berdasarkan data dari *confusion matrix*, yaitu 74,46%. Perhitungan manual ini memverifikasi keakuratan hasil yang diberikan oleh model klasifikasi *Naive Bayes*. Meskipun hasil ini menunjukkan performa yang cukup baik, masih ada ruang untuk peningkatan, misalnya dengan mengoptimalkan fitur, menambah data atau mencoba algoritma klasifikasi lainnya.

BAB IV

PENUTUP

4.1 Simpulan

Penelitian ini berhasil mencapai tujuannya untuk mengukur akurasi metode *Naive Bayes* dengan pembobotan TF-IDF dalam klasifikasi teks *Quick Count* Pemilihan Presiden Indonesia 2024 di X Twitter. Hasil analisis menunjukkan model mencapai akurasi 74,46% dalam mengklasifikasikan teks. Namun, performa model tidak seimbang antara kelas sentimen, dengan kemampuan yang lebih baik dalam mengidentifikasi sentimen negatif dibandingkan sentimen positif. Meskipun akurasi keseluruhan cukup baik, masih terdapat ruang untuk peningkatan, terutama dalam pengenalan sentimen positif. Temuan ini mengindikasikan bahwa metode *Naive Bayes* dengan pembobotan TF-IDF memiliki potensi yang baik dalam mengklasifikasi teks terkait *Quick Count*, namun memerlukan penyempurnaan lebih lanjut untuk meningkatkan keseimbangan performa antar kelas sentimen.

4.2 Saran

Berdasarkan hasil penelitian, beberapa saran dapat dipertimbangkan untuk meningkatkan kinerja model klasifikasi teks *Quick Count* Pemilihan Presiden Indonesia 2024 di X Twitter. Mengatasi ketidakseimbangan kelas melalui teknik *oversampling* atau *undersampling* mungkin bermanfaat. Penggunaan metode ekstraksi fitur alternatif seperti *word embeddings* bisa membantu menangkap makna kata lebih baik. Membandingkan performa dengan algoritma klasifikasi lain seperti *Gradient Boosting* atau *Random Forest* dapat memberikan wawasan baru. Peningkatan proses *preprocessing*, terutama untuk bahasa informal, juga bisa menjadi area perbaikan. Memperbesar dataset dan menerapkan *cross-validation* mungkin meningkatkan generalisasi dan akurasi estimasi performa model. Penerapan saran-saran ini diharapkan dapat meningkatkan akurasi dan keseimbangan model secara signifikan.

DAFTAR RUJUKAN

- Albab, M. U., Karuniawati, Y., & Fawaiq, M. N. (2023). Optimization of the Stemming Technique on Text preprocessing President 3 Periods Topic. *Jurnal TRANSFORMATIKA*, 20(2), 1–10. <https://journals.usm.ac.id/index.php/transformatika/page1>
- Amin, I. (2023). *Media Sosial antara Potensi & Bahaya dalam Kampanye Pemilu 2024*. Tirto.Id. <https://tirto.id/media-sosial-antara-potensi-bahaya-dalam-kampanye-pemilu-2024-gNwH>
- Andi Saadillah, Andi Haryudi, Muhammad Reskiawan, & Alam Ikhsanul Amanah. (2023). Penggunaan Bahasa Sarkasme Netizen di Media Sosial. *Jurnal Onoma: Pendidikan, Bahasa, Dan Sastra*, 9(2), 1437–1447. <https://doi.org/10.30605/onoma.v9i2.2367>
- Darman, R. (2023). ANALISIS SENTIMEN RESPONS TWITTER TERHADAP PERSYARATAN BADAN PENYELENGGARA JAMINAN SOSIAL (BPJS) DI KANTOR PERTANAHAN. *JURNAL WIDYA BHUMI ANALISIS*, 3(2), 113–136.
- Farokhah, L. (2020). Implementasi K-Nearest Neighbor Untuk Klasifikasi Bunga Dengan Ekstraksi Fitur Warna Rgb Implementation of K-Nearest Neighbor for Flower Classification With Extraction of Rgb Color Features. *Jurnal Teknologi Informasi Dan Ilmu Komputer (JTIK)*, 7(6), 1129–1136. <https://doi.org/10.25126/jtiik.202072608>
- Fikri, M. I., Sabrila, T. S., Azhar, Y., & Malang, U. M. (2020). Perbandingan Metode Naïve Bayes dan Support Vector Machine pada Analisis Sentimen Twitter. *SMATIKA Jurnal: STIKI Informatika Jurnal*, 10(2), 71–76.
- Hamka, M., Alfatari, N., & Ratna Sari, D. (2022). Analisis Sentimen Produk Kecantikan Jenis Serum Menggunakan Algoritma Naïve Bayes Classifier. *Jurnal Sistem Komputer Dan Informatika (JSON)*, 4(1), 64. <https://doi.org/10.30865/json.v4i1.4740>
- Hananto Widodo, & Dicky Eko Prasetyo. (2021). Penataan Kewenangan KPU dan Bawaslu dalam Melakukan Pengawasan dan Menangani Sengketa Proses Pemilu. *Perspektif Hukum*, 21(2), 17–38. <https://doi.org/10.30649/ph.v21i2.93>
- Husada, I. N., & Toba, H. (2020). Pengaruh Metode Penyeimbangan Kelas Terhadap Tingkat Akurasi Analisis Sentimen pada Tweets Berbahasa Indonesia. *Jurnal Teknik Informatika Dan Sistem Informasi*, 6(2), 400–413. <https://doi.org/10.28932/jutisi.v6i2.2743>
- Muslimin, M., & Lusiana, V. (2023). Analisis Sentimen Terhadap Kenaikan Harga Bahan Pokok Menggunakan Metode Naive Bayes Classifier. *Jurnal Media Informatika Budidarma*, 7(3), 1200–1209. <https://doi.org/10.30865/mib.v7i3.6418>
- Novitasari, I., Kurniawan, T. B., Dewi, D. A., & Misinem. (2022). Analisis sentimen masyarakat terhadap tweet ruang guru menggunakan algoritma naive bayes classifier (NBC) [Analysis of public sentiment towards ruang guru's tweets using the Naive Bayes Classifier (NBC) algorithm]. *Jurnal Mantik*, 6(3), 2685–4236.
- Nugraha, N. B., Suhaidi, M., & S, J. (2019). Aplikasi Quick Count Pada Pemilu Legislatif Dengan Metode Sainte Lague Berbasis Mobile. *INFORMATIKA : JURNAL INFORMATIKA, MANAJEMEN DAN KOMPUTER*, 11(1), 76. <https://doi.org/10.36723/juri.v11i1.158>
- Nurmalasari, D., & Ribut Yuliantoro, H. (2022). Implementasi Ekstraksi Fitur untuk Pengelompokan Dokumen Proposal Menggunakan Algoritma Naïve Bayes. *Jurnal Komputer Terapan*, 8(1), 194–203. <https://doi.org/10.35143/jkt.v8i1.5351>
- Putri, A., Hardiana, C. S., Novfuja, E., Siregar, F. T. P., Rahmaddeni, R., Fatma, Y., & Wahyuni, R. (2023). Komparasi Algoritma K-NN, Naive Bayes dan SVM untuk Prediksi Kelulusan Mahasiswa Tingkat Akhir. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 3(1), 20–26. <https://doi.org/10.57152/malcom.v3i1.610>

- Rhima Indria Saraswati, A. P. (2020). Kepercayaan Masyarakat Terhadap Hasil Quick Count Pada Pemilihan Presiden Dan Legislatif Tahun 2019. *Jurnal Kewarganegaraan*, 4(1), 25.
- Rizki, M., Fikri Hidayattullah, M., & Dwi Intan Af'idah. (2023). Klasifikasi Opini Publik di Twitter Terhadap Bakal Calon Presiden Indonesia Tahun 2024 Menggunakan LSTM Secara Realtime Berbasis Website. *Infotekmesin*, 14(2), 285–295. <https://doi.org/10.35970/infotekmesin.v14i2.1908>
- Salim, E., & Solichin, A. (2022). ANALISIS SENTIMEN PADA MEDIA SOSIAL TWITTER TERHADAP PELAYANAN DINAS KEPENDUDUKAN DAN PENCATATAN SIPIL MENGGUNAKAN ALGORITMA NAÏVE BAYES. In *INDONESIA JOURNAL INFORMATION SYSTEM (IDEALIS)* (Vol. 5, Issue 2). <http://jom.fti.budiluhur.ac.id/index.php/IDEALIS/indexEmilSalim%7Chttp://jom.fti.budiluhur.ac.id/index.php/IDEALIS/index%7C>
- Sidiq, R. P., Dermawan, B. A., & Umidah, Y. (2020). Sentimen Analisis Komentar Toxic pada Grup Facebook Game Online Menggunakan Klasifikasi Naïve Bayes. *Jurnal Informatika Universitas Pamulang*, 5(3), 356. <https://doi.org/10.32493/informatika.v5i3.6571>
- Sosiawan, E. A., & Wibowo, R. (2020). Kontestasi Berita Hoax Pemilu Presiden Tahun 2019 di Media Daring dan Media Sosial. *Jurnal Ilmu Komunikasi*, 17(2), 133. <https://doi.org/10.31315/jik.v17i2.3695>
- Sriani, Suhardi, L. S. Y. (2024). ANALISIS SENTIMEN PENGGUNA APLIKASI MOBILE JKN MENGGUNAKAN ALGORITMA NAÏVE BAYES. *Journal Of Science and Social Research*, 4307(May), 555–563.
- Surya, M. R. A., Martanto, & Hayati, U. (2024). Analisis Sentimen Ulasan Pengguna Ovo Menggunakan Algoritma Naive Bayes Pada Google Play Store. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(3), 2780–2786.
- Ulfa, K., Purnomo, E. P., & Kasiwi, A. N. (2020). The Campaign Strategy of 2019 Presidential and Vice-Presidential Elections on Social Media. *Society*, 8(2), 284–297. <https://doi.org/10.33019/society.v8i2.137>
- Wulandari, L. (2024). Framing and Sentiment Analysis of Gerindra Party Political Communication on Twitter (X) Towards the 2024 Elections Framing and Sentiment Analysis of Gerindra Party Political. *Jurnal Wacana Politik*, 8(January), 2–10. <https://doi.org/10.24198/jwp.v8i1.51967>
- Zhafira, D. F., Rahayudi, B., & Indriati, I. (2021). Analisis Sentimen Kebijakan Kampus Merdeka Menggunakan Naive Bayes dan Pembobotan TF-IDF Berdasarkan Komentar pada Youtube. *Jurnal Sistem Informasi, Teknologi Informasi, Dan Edukasi Sistem Informasi*, 2(1), 55–63. <https://doi.org/10.25126/justsi.v2i1.24>

RIWAYAT HIDUP



Aditya Pranata atau biasanya dipanggil Adi, lahir di Liang pada tanggal 01 Juli 2002 dari pasangan Bapak Taufik Hidayat dan Ibu Norsiah penulis merupakan anak ketiga dari 3 bersaudara. Penulis Berkebangsaan Indonesia dan beragama Islam. Penulis berasal dari Desa Liang dan bertempat tinggal di Jalan Awang Long Kecamatan Kota Bangun Kabupaten Kutai Kartanegara. Penulis menempuh pendidikan dari SD Negeri 021 Kota Bangun (2008-2012) lalu pindah pada SD Negeri 005 Kota Bangun (2012-2014), selanjutnya menempuh pendidikan pada sekolah pertama di SMP Negeri 2 Kota Bangun (2015-2017), kemudian untuk pendidikan sekolah menengah atas penulis menyelesaikannya di MAN 1 Kutai Kartanegara (2018-2020), dan untuk pendidikan perguruan tinggi penulis tempuh di Universitas Muhammadiyah Kalimantan Timur (UMKT) dari tahun 2020 sampai sekarang.

LAMPIRAN

Lampiran 1 CV *Expert Labeling*

Rizka Audina Sinaga

Jawa Tengah | rizkasinaga17@gmail.com



PERSONAL INFORMASI

Berpengalaman lebih dari 4 tahun dalam keterampilan monitoring serta analisis data media online dan sosial. Selain itu, saya telah memberikan rekomendasi yang berharga untuk pengambilan keputusan terkait strategi komunikasi dan kebijakan publik kepada klien. Keahlian saya tidak hanya terbatas pada aspek analitis, tetapi juga pada kemampuan menulis yang telah terbukti melalui kontribusi saya di berbagai media, baik online, sosial, maupun cetak.

PENDIDIKAN

Institut Agama Islam Negeri Surakarta (2015-2019)

Komunikasi dan Penyiaran Islam

KEMAMPUAN

- Software Dasar Microsoft Office
- Analisis Media
- Strategi Komunikasi
- Penulisan Berita dan Konten Kreatif
- Search Engine Optimization (SEO) dan Content Management System (CMS)

PENGALAMAN

PT Indonesia Indicator

Media Analyst (2020-2024)

- Mengumpulkan, menganalisis, dan menginterpretasi data dari berbagai sumber media untuk mengidentifikasi tren dan pola yang relevan.
- Menggunakan alat analisis data dan teknologi big data untuk mengolah informasi dan menghasilkan laporan yang komprehensif.
- Memantau dan mengevaluasi sentimen publik dan tren media sosial untuk memberikan wawasan strategis kepada perusahaan.
- Menyusun dan menyampaikan laporan analisis media secara berkala kepada manajemen dan pemangku kepentingan lainnya.
- Berkolaborasi dengan tim lintas fungsi untuk mengembangkan strategi komunikasi yang efektif berdasarkan hasil analisis data.
- Mengembangkan dan mengoptimalkan metode analisis media untuk meningkatkan akurasi dan efisiensi pemantauan.
- Mengidentifikasi peluang dan ancaman yang muncul dari data media dan memberikan rekomendasi yang berdasar untuk tindakan lanjut.
- Menjaga pengetahuan terkini tentang tren teknologi big data dan analisis media untuk memastikan metodologi dan alat yang digunakan selalu mutakhir.

Lampiran 2 Code Pengumpulan Data

```

✓ TOKEN AUTENTIKASI TWITTER
[ ] # @title TOKEN AUTENTIKASI TWITTER
    twitter_auth_token = 'd5753d38fc5d1baa12358939e1034e0d0df10b4'

✓ INSTALASI PANDAS DAN NODE.JS
[ ] # @title INSTALASI PANDAS DAN NODE.JS
    pip install pandas                # Menginstal library pandas
    sudo apt-get update                # Memperbarui daftar paket
    sudo apt-get install -y ca-certificates curl gnupg    # Menginstal sertifikat CA, curl, dan gnupg
    sudo mkdir -p /etc/apt/keyrings    # Membuat direktori untuk menyimpan kunci GPG
    curl -fsSL https://deb.nodesource.com/gpgkey/nodesource-repo.gpg.key | sudo gpg --dearmor -o /etc/apt/keyrings/nodesource.gpg # Mengunduh dan menyimpan kunci GPG Node
    # Menambahkan repositori NodeSource ke sources.list
    INODE_MAJOR=20 && echo "deb [signed-by=/etc/apt/keyrings/nodesource.gpg] https://deb.nodesource.com/node_${NODE_MAJOR}.x nodistro main" | sudo tee /etc/apt/sources.list.d
    sudo apt-get update                # Memperbarui daftar paket lagi
    sudo apt-get install nodejs -y     # Menginstal Node.js
    node -v                            # Menampilkan versi Node.js yang terinstal

✓ MENGUMPULKAN DATA TWEET MENGGUNAKAN TWEET-HARVEST
[ ] # @title MENGUMPULKAN DATA TWEET MENGGUNAKAN TWEET-HARVEST
    filename = 'Quick-Count.csv'        # Nama file untuk menyimpan hasil pengumpulan data
    search_keyword = 'quick count lang:id until:2024-02-15 since:2024-02-14' # Kata kunci pencarian dan filter waktu
    limit = 2100                        # Batas jumlah tweet yang akan dikumpulkan
    # Menjalankan tweet-harvest untuk mengumpulkan tweet
    lnpx -y tweet-harvest@2.6.0 -o "{filename}" -s "{search_keyword}" --tab "LATEST" -l {limit} --token {twitter_auth_token}

✓ MEMBACA DAN MENAMPILKAN DATA TWEET
[ ] # @title MEMBACA DAN MENAMPILKAN DATA TWEET
    import pandas as pd                # Mengimpor library pandas untuk manipulasi data
    file_path = f"tweets-data/{filename}" # Menentukan path file CSV
    df = pd.read_csv(file_path, delimiter=";") # Membaca file CSV ke dalam DataFrame pandas
    display(df)                        # Menampilkan DataFrame

✓ MENGHITUNG JUMLAH TWEET DALAM DATAFRAME
[ ] # @title MENGHITUNG JUMLAH TWEET DALAM DATAFRAME
    num_tweets = len(df)               # Menghitung jumlah total tweet dalam DataFrame
    print(f"Jumlah tweet dalam dataframe adalah {num_tweets}.") # Mencetak jumlah total tweet

```

Lampiran 3 Code Pre-Processing

```
LIBRARY
[] # @title LIBRARY
import re # Modul untuk regular expression
import pandas as pd # Manipulasi data tabular dengan Pandas
import numpy as np # Operasi numerik dengan Numpy

# Instalasi Sastrawi
!pip install Sastrawi # Paket Sastrawi untuk pemrosesan bahasa Indonesia
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory # Stemming bahasa Indonesia dari Sastrawi

import nltk # Pemrosesan teks dengan NLTK
from nltk.corpus import stopwords # Daftar stopwords dari NLTK
# Download data NLTK yang diperlukan
nltk.download('stopwords') # Daftar stopwords dari NLTK

MEMBACA DAN MENAMPILKAN INFORMASI DATA
[] # @title MEMBACA DAN MENAMPILKAN INFORMASI DATA
data = pd.read_csv('DATA-CSV.csv') # Membaca dan memuat data dari file CSV "Data-Quick Count.csv"
data.info() # Menampilkan informasi tentang struktur data

VISUALISASI DISTRIBUSI SENTIMEN
[] # @title VISUALISASI DISTRIBUSI SENTIMEN
import matplotlib.pyplot as plt # Menginstall library matplotlib

sentiment_counts = data['Sentimen'].value_counts() # Menghitung jumlah setiap sentimen
positif = sentiment_counts.get('Positif', 0) # Mengambil jumlah sentimen positif
negatif = sentiment_counts.get('Negatif', 0) # Mengambil jumlah sentimen negatif

labels = ['Positif (positif)', 'Negatif (negatif)'] # Membuat label untuk pie chart
sizes = [positif, negatif] # Menentukan ukuran setiap bagian pie
colors = ['#219653', '#9C27B0'] # Menentukan warna untuk setiap bagian pie

plt.figure(figsize=(8, 6)) # Membuat figure dengan ukuran 8x6
plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', shadow=True, startangle=120) # Membuat pie chart
plt.axis('equal') # Memastikan pie berbentuk lingkaran
plt.title('Distribusi Sentimen') # Menambahkan judul grafik
plt.show() # Menampilkan grafik

MENAMPILKAN DATA FRAME
[] # @title MENAMPILKAN DATA FRAME
columns = ['full_text', 'Sentimen'] # Mendefinisikan kolom yang akan digunakan
df = pd.DataFrame(data, columns=columns).iloc[0:2115] # Membuat DataFrame dengan 549 baris pertama dari data
df.head(5) # Menampilkan 5 baris pertama dari DataFrame

CASE FOLDING
[] # @title CASE FOLDING
def case_folding(text):
    if isinstance(text, str):
        lowercase_text = text.lower() # Menonaktifkan apakah input adalah string
        lowercase_text = lowercase_text # Mengubah teks menjadi huruf kecil
        lowercase_text = lowercase_text # Mengembalikan teks yang sudah diubah
    else:
        return text # Mengembalikan input asli jika bukan string
df['case_folding'] = df['full_text'].apply(case_folding) # Menerapkan fungsi case_folding pada kolom 'full_text'
df.head(5) # Menampilkan 5 baris pertama dari DataFrame

CLEANSING
[] # @title CLEANSING
def clean_text(tweet):
    tweet = re.sub(r'https://[a-zA-Z0-9.]*', '', tweet) # Menghapus URL
    tweet = re.sub(r'<.*>', '', tweet) # Menghapus tag HTML
    emoji_pattern = re.compile("["
        u'\U0001F600-\U0001F64F' # emoticons
        u'\U0001F300-\U0001F3FF' # simbol & pictograf
        u'\U0001F1E0-\U0001F1FF' # simbol transportasi & peta
        u'\U0001F170-\U0001F1FF' # bendera (IO)
        u'\U00002700-\U000027BF'
        u'\U00002600-\U000026FF"
    ]) # flaggero UNICODE
    tweet = emoji_pattern.sub('', tweet) # Menghapus emoji
    tweet = re.sub(r'@.*', '', tweet) # Menghapus username
    tweet = re.sub(r'#.*', '', tweet) # Menghapus angka
    tweet = re.sub(r'[\s]', ' ', tweet) # Normalisasi karakter berulang
    tweet = re.sub(r'[_]', '', tweet) # Menghapus simbol kecuali underscore
    tweet = re.sub(r' ', '', tweet.strip()) # Menghapus spasi berlebih
    return tweet # Mengembalikan teks yang sudah dibersihkan
df['cleaning'] = df['case_folding'].apply(clean_text) # Aplikasi fungsi clean_text ke kolom 'case_folding'
df.head(5) # Tampilkan 5 baris pertama dari DataFrame

STOPWORD REMOVAL
[] # @title STOPWORD REMOVAL
stop_words = stopwords.words('Indonesian') # Mengambil daftar stopwords bahasa Indonesia dari NLTK
additional_stopwords = [
    'ya', 'bye', 'aja', 'ga', 'nyal', 'ya', 'sah', 'saya', 'kamu', 'dia', 'mencari',
    'kita', 'saya', 'kalian', 'itu', 'sini', 'tuh', 'sana', 'dan', 'dong', 'kalo', 'lgaa',
    'kan', 'pau', 'ah', 'kok', 'tina', 'mah', 'bin', 'tadi', 'di', 'di', 'tp', 'ngelihat',
    'luc', 'ters', 'dgn', 'sy', 'lg', 'jg', 'kn', 'sah', 'ad', 'gk', 'sm', 'bgt', 'nden',
    'ki', 'knt', 'aj', 'eng', 'amp', 'dim', 'ga', 'lu', 'kay', 'to', 'n', 'abis', 'na',
    'sana', 'lag', 'mah', 'kay', 'kay', 'ngp', 'td', 'mah', 'tm', 'dat', 'perasaan',
    'nti', 'gini', 'kn', 'yuk', 'sannya', 'krng', 'skg', 'sy', 'gmn', 'fy', 'udah',
    'ong', 'jdi', 'atr', 'gitu', 'sport', 'ann', 'oto', 'tobat', 'tuh', 'keli', 'piis',
    'ban', 'hig', 'krng', 'luc', 'udah', 'kkn', 'dini', 'mg', 'mlm', 'pdhl', 'pagi', 'gmn',
    'treak', 'kita', 'yuk', 'ah', 'yah', 'wah', 'jgn', 'lg', 'ah', 'tgi', 'n', 'ah', 'tpi', 'ba']
stop_words.extend(additional_stopwords) # Menambahkan stopwords tambahan ke dalam daftar
def remove_stopwords(text):
    words = text.split() # Membagi teks menjadi kata-kata
    cleaned_text = ' '.join(word for word in words if word not in stop_words) # Menghapus stopwords dan menggabungkan kata-kata kembali
    return cleaned_text
df['stopword_removal'] = df['cleaning'].apply(remove_stopwords) # Menerapkan fungsi remove_stopwords pada kolom 'cleaning'
df.head(5) # Menampilkan 5 baris pertama dari DataFrame

STEMMING
[] # @title STEMMING
factory = StemmerFactory() # Membuat objek StemmerFactory
stemmer = factory.create_stemmer() # Membuat objek stemmer dari factory

def stem_text(text):
    stemmed_text = ' '.join([stemmer.stem(word) for word in text.split()]) # Melakukan stemming pada setiap kata dan menggabungkannya kembali
    return stemmed_text
df['stemming'] = df['stopword_removal'].apply(stem_text) # Menerapkan fungsi stem_text pada kolom 'stopword_removal'
df.head(5) # Menampilkan 5 baris pertama dari DataFrame

DOWNLOAD CSV HASIL PRE-PROCESSING
[] # @title DOWNLOAD CSV HASIL PRE-PROCESSING
df.to_csv('HASIL-PRE.csv', encoding='utf8', index=False)
```

Lampiran 4 Code *Word Cloud*

```

VISUALISASI WORD CLOUD

[] # @title VISUALISASI WORD CLOUD
import matplotlib.pyplot as plt # Import plotting library
from wordcloud import WordCloud # Import WordCloud for visualization
from nltk.corpus import stopwords, wordnet # Import NLTK corpus tools
import pandas as pd # Import pandas for data manipulation
import nltk # Import NLTK library
nltk.download('stopwords') # Download stopwords dataset
nltk.download('wordnet') # Download WordNet dataset

kunci_utama = ['quick', 'count', 'quick count'] # Define main keywords to remove

def get_synonyms(word): # Function to get synonyms
    synonyms = set()
    for syn in wordnet.synsets(word):
        for lemma in syn.lemmas():
            synonyms.add(lemma.name().replace('_', ' '))
    return synonyms

all_stopwords = set(stopwords.words('indonesian')) # Get Indonesian stopwords
for key in kunci_utama: # Add synonyms of main keywords
    all_stopwords.update(get_synonyms(key))
all_stopwords.update(kunci_utama) # Add main keywords to stopwords

def clean_text(text): # Function to clean text
    words = text.split()
    return ' '.join(word for word in words if word.lower() not in all_stopwords)

positif_text = ' '.join(df[df['Sentimen'] == 'Positif']['stemming']) # Concatenate positive texts
negatif_text = ' '.join(df[df['Sentimen'] == 'Negatif']['stemming']) # Concatenate negative texts

positif_text_clean = clean_text(positif_text) # Clean positive text
negatif_text_clean = clean_text(negatif_text) # Clean negative text

def generate_wordcloud(text): # Function to generate word cloud
    return WordCloud(width=800, height=400, background_color='white', stopwords=all_stopwords).generate(text)
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(20, 10)) # Create subplot

wordcloud_positif = generate_wordcloud(positif_text_clean) # Generate positive word cloud
ax1.imshow(wordcloud_positif, interpolation='bilinear') # Display positive word cloud
ax1.axis('off') # Turn off axis
ax1.set_title('Word Cloud Sentimen Positif') # Set title for positive

wordcloud_negatif = generate_wordcloud(negatif_text_clean) # Generate negative word cloud
ax2.imshow(wordcloud_negatif, interpolation='bilinear') # Display negative word cloud
ax2.axis('off') # Turn off axis
ax2.set_title('Word Cloud Sentimen Negatif') # Set title for negative

plt.tight_layout() # Adjust layout
plt.show() # Display the plot

```

Lampiran 5 Code Pembobotan TF-IDF

```

PEMBOBOTAN TF-IDF

[] # @title PEMBOBOTAN TF-IDF
import pandas as pd # Mengimpor library pandas
import numpy as np # Mengimpor library numpy
from sklearn.feature_extraction.text import TfidfVectorizer # Mengimpor TfidfVectorizer dari sklearn

df = pd.read_csv('HASIL-PRE.csv') # Membaca file CSV
documents = df['stemming'].tolist() # Mengambil data stemming sebagai list
sentiments = df['Sentimen'].tolist() # Mengambil data sentimen sebagai list
N = len(documents) # Menghitung jumlah dokumen

vectorizer = TfidfVectorizer() # Membuat objek TfidfVectorizer
tfidf_matrix = vectorizer.fit_transform(documents) # Melakukan fit dan transform pada dokumen
tfidf_df = pd.DataFrame(tfidf_matrix.toarray(), columns=vectorizer.get_feature_names_out()) # Membuat DataFrame dari matriks TF-IDF

total_vocabulary = len(vectorizer.get_feature_names_out()) # Menghitung total kosakata unik
print(f" Total Vocabulary: {total_vocabulary}") # Mencetak total kosakata

print("\n Term Frequency-Inverse Document Frequency (TF-IDF):") # Mencetak judul hasil
for i, (row, col) in enumerate(zip(tfidf_matrix.nonzero()[0][:40], tfidf_matrix.nonzero()[1][:40])): # Loop untuk 40 elemen pertama
    print(f"({row}, {col})\t{tfidf_matrix[row, col]}") # Mencetak nilai TF-IDF
    print(f" ... ..") # Mencetak pemisah
for i, (row, col) in enumerate(zip(tfidf_matrix.nonzero()[0][-3:], tfidf_matrix.nonzero()[1][-3:])): # Loop untuk 3 elemen terakhir
    print(f"({row}, {col})\t{tfidf_matrix[row, col]}") # Mencetak nilai TF-IDF

DOWNLOAD EXCEL HASIL TF-IDF

[] # @title DOWNLOAD EXCEL HASIL TF-IDF
import pandas as pd # Mengimpor library pandas
import numpy as np # Mengimpor library numpy
from sklearn.feature_extraction.text import TfidfVectorizer # Mengimpor TfidfVectorizer dari sklearn

df = pd.read_csv('HASIL-PRE.csv') # Membaca file CSV hasil preprocessing
documents = df['stemming'].tolist() # Mengambil data stemming sebagai list
sentiments = df['Sentimen'].tolist() # Mengambil data sentimen sebagai list
N = len(documents) # Menghitung jumlah dokumen

vectorizer = TfidfVectorizer() # Membuat objek TfidfVectorizer
tfidf_matrix = vectorizer.fit_transform(documents) # Melakukan fit dan transform pada dokumen
tfidf_df = pd.DataFrame(tfidf_matrix.toarray(), columns=vectorizer.get_feature_names_out()) # Mengubah hasil menjadi DataFrame
tfidf_df.to_excel('HASIL-TFIDF.xlsx', index=False) # Menyimpan DataFrame ke file Excel

```

Lampiran 6 Code Split Data

```

v SPLIT DATA

[ ] # @title SPLIT DATA
from sklearn.feature_extraction.text import TfidfVectorizer # Mengimpor TfidfVectorizer dari sklearn
from sklearn.model_selection import train_test_split # Mengimpor train_test_split dari sklearn
import matplotlib.pyplot as plt # Mengimpor library matplotlib

tfidf_vectorizer = TfidfVectorizer() # Membuat objek TfidfVectorizer
X = tfidf_vectorizer.fit_transform(df['stemming']) # Melakukan fit dan transform pada kolom 'stemming'
y = df['Sentimen'] # Mengambil kolom 'Sentimen' sebagai target

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42) # Membagi data menjadi set training dan testing
total_samples = X.shape[0] # Menghitung total sampel
train_samples = X_train.shape[0] # Menghitung jumlah sampel training
test_samples = X_test.shape[0] # Menghitung jumlah sampel testing
fig, ax = plt.subplots(figsize=(10, 6)) # Membuat figure dan axes

bars = ax.bar(['Total Sampel', 'Sampel Training', 'Sampel Testing'], # Membuat bar chart
             [total_samples, train_samples, test_samples], # Data untuk setiap bar
             color=['#4682B4', '#80C4DE', '#80E0E6']) # Warna untuk setiap bar

for bar in bars:
    height = bar.get_height() # Loop untuk setiap bar
    ax.text(bar.get_x() + bar.get_width()/2., height, # Mendapatkan tinggi bar
           f'{height}', # Menambahkan teks di atas bar
           ha='center', va='bottom') # Teks yang ditambahkan (jumlah sampel)
                                   # Pengaturan posisi teks

ax.set_ylabel('Jumlah Sampel') # Menambahkan label sumbu y
ax.set_title('Distribusi Sampel dalam Dataset') # Menambahkan judul grafik

ax.text(1, train_samples/2, f'{train_samples/total_samples:.1%}', # Menambahkan persentase pada bar training
       ha='center', va='center') # Pengaturan posisi teks
ax.text(2, test_samples/2, f'{test_samples/total_samples:.1%}', # Menambahkan persentase pada bar testing
       ha='center', va='center') # Pengaturan posisi teks
plt.tight_layout() # Mengatur layout agar tidak tumpang tindih
plt.show() # Menampilkan grafik

```

Lampiran 7 Code Klasifikasi

```

v EVALUASI MODEL NAIVE BAYES DAN VISUALISASI CONFUSION MATRIX

[ ] # @title EVALUASI MODEL NAIVE BAYES DAN VISUALISASI CONFUSION MATRIX
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score # Mengimpor metrik evaluasi
from sklearn.naive_bayes import MultinomialNB # Mengimpor MultinomialNB classifier
import seaborn as sns # Mengimpor library Seaborn untuk visualisasi data
import matplotlib.pyplot as plt # Mengimpor library Matplotlib untuk plotting

nb_classifier = MultinomialNB() # Inisialisasi classifier Naive Bayes
nb_classifier.fit(X_train, y_train) # Melatih classifier dengan data training
y_pred = nb_classifier.predict(X_test) # Melakukan prediksi pada data testing
conf_matrix = confusion_matrix(y_test, y_pred) # Menghitung confusion matrix

sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', # Membuat heatmap dari confusion matrix
           xticklabels=['NEGATIF', 'POSITIF'],
           yticklabels=['NEGATIF', 'POSITIF']) # Memberikan label pada sumbu x
plt.xlabel('Predicted') # Memberikan label pada sumbu y
plt.ylabel('Actual') # Memberikan label pada sumbu y
plt.title('Confusion Matrix') # Memberikan judul pada plot
plt.show() # Menampilkan plot
accuracy = accuracy_score(y_test, y_pred) # Menghitung akurasi model
print("\nAccuracy:", accuracy) # Menampilkan akurasi
print("\nClassification Report:") # Menampilkan judul laporan klasifikasi
print(classification_report(y_test, y_pred, zero_division=1)) # Menampilkan laporan klasifikasi lengkap

```

Lampiran 8 Lembar Tindak Lanjut

Dokumen Revisi dan Tindak Lanjut Peserta Ujian Sidang Hasil

Nama: Aditya Pranata

NIM: 2011102441052

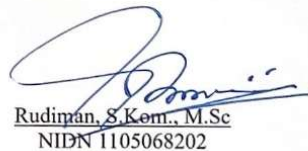
No	Nama Dosen	Permintaan Revisi	Tindak lanjut	Halaman
1	Naufal Azmi Verdikha, S.Kom., M.Eng	Judul perlu direvisi, karena metode penelitian yang dimaksud bukan untuk mendeteksi atau mencari hasil quick count	Telah di rubah dengan judul baru sesuai dengan penelitian	i-iii
2	Naufal Azmi Verdikha, S.Kom., M.Eng	Penulisan belum sesuai dengan pedoman	Telah di rubah sesuai dengan pedoman	1-3
3	Naufal Azmi Verdikha, S.Kom., M.Eng	Belum ada deskripsi urgensi penelitian. Urgensi ditulis untuk melihat fokus penelitian dan pentingnya penelitian tersebut dilakukan.	Sudah ditambahkan urgensi dari penelitian yang diminta	1-4
4	Naufal Azmi Verdikha, S.Kom., M.Eng	Rumusan masalah masih di tingkat D3, belum sampai di tahap analisis. Rumusan masalah perlu melibatkan unsur metode, objek, dan tolak ukur penelitian	Telah dirubah dan diperbaiki pada rumusan masalah	2
5	Naufal Azmi Verdikha, S.Kom., M.Eng	Setiap Tahapan penelitian perlu dijelaskan perlangkah-langkah dari awal hingga menghasilkan tahapan yang dituju.	Sudah ditambahkan setiap tahapan penelitian pada bab 2	4-10
6	Naufal Azmi Verdikha, S.Kom., M.Eng	Penentuan rasio data split perlu ada landasan kuat	Sudah ditambahkan referensi landasan kuat pada penentuan rasio split data	8
8	Rudinman, S.Kom., M.Sc	Naskah harus dibuat minimal sampai BAB 3	Telah dibuat naskah sampai dengan BAB 3	12-18

Menyetujui


Penguji 1


Naufal Azmi Verdikha, S.Kom., M.Eng
NIDN 1114048801

Penguji 2


Rudinman, S.Kom., M.Sc
NIDN 1105068202

Lampiran 9 Surat Izin Penelitian

 UMKT Program Studi Teknik Informatika Fakultas Sains dan Teknologi	Telp. 0541-748511 Fax. 0541-766832 Website http://informatika.umkt.ac.id email: informatika@umkt.ac.id
--	--

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Nomor : 055-002/KET/FST.1/A/2024
Lampiran : -
Perihal : **Keterangan Pengambilan Data Sekunder**

Assalamu'alaikum Warrahmatullahi Wabarrakatuh

Puji Syukur kepada Allah Subhanahu wa ta'ala yang senantiasa melimpahkan Rahmat-Nya kepada kita sekalian. Amin.

Dengan surat ini, kami menerangkan bahwa mahasiswa berikut:

No	Nama	NIM
1	Ricky Albin Pranata	2011102441017
2	Aditya Pranata	2011102441052


Melakukan penelitian dengan pengambilan data sekunder di Media Sosial Twitter data yang diambil yaitu sentimen mengenai quick count.


Demikian hal ini disampaikan, atas kerjasamanya kami ucapkan terima kasih.

Wassalamu'alaikum Warrahmatullahi Wabarrakatuh

Samarinda, 19 Dzulhijjah 1445 H
26 Juni 2024 M

Program Studi S1 Teknik Informatika


Hausyah, S.Kom., M.TI
N. 1118019203



Kampus 1 : Jl. Ir. H. Juanda, No.15, Samarinda
Kampus 2 : Jl. Pelita Pesona Mahakam, Samarinda

Lampiran 10 Kartu Kendali Bimbingan

KARTU KENDALI BIMBINGAN LAPORAN KARYA ILMIAH

Nama : Aditya Pranata
 NIM : 2011102441052
 Nama Dosen Pembimbing : Rudiman, S.Kom., M.Sc
 Judul Penelitian : Metode Pembobotan TF-IDF untuk Klasifikasi Teks Quick Count Pemilihan Presiden Indonesia 2024 pada X Twitter Menggunakan Metode Naive Bayes

No	Tanggal	Uraian Pembimbingan	Paraf Dosen
1	07/02/2024	Persetujuan bimbingan dengan topik penelitian	h
2	14/02/2024	Mencari permasalahan topik yang diteliti dan objek	h
3	22/02/2024	Malutuban tahap pencarian data di python googlecolab	h
4	29/02/2024	Menentukan judul dan mulai perancangan latar belakang yang ditetapkan oleh dosen pembimbing	h
5	09/03/2024	Pengajuan latar belakang harus mengikuti urutan yang telah ditetapkan oleh dosen pembimbing	h
6	13/03/2024	Malutuban revisi yang diarahkan dosen pembimbing	h
7	18/03/2024	Dosen pembimbing membantukan beberapa saran terhadap konsep pengajuan judul ke prodi	h
8	27/03/2024	membantukan beberapa revisi terhadap latar belakang, rumusan masalah, tujuan, dan manfaat	h
9	05/04/2024	Masuk ke bab 2 menggunakan strategi yang telah dibentukan dosen pembimbing	h
10	24/04/2024	membantukan masukan dan revisi pada Bab 2	h
11	26/06/2024	Dosen pembimbing memberi masukan Bab 3	h
12	02/07/2024	beberapa saran dan masukan dari dosen pembimbing terkait Seminar hasil	h
13	19/07/2024	membantukan revisi terkait jurnal yang akan di submit	h
14	15/07/2024	Malutuban revisi Bab 3 yang diarahkan dosen pembimbing	h

Dosen Pembimbing

(Rudiman, S.Kom., M.Sc.)



(Rudiman, S.Kom., M.Sc.)

SKRIPSI ADITYA PRANATA

by S1 Teknik Informatika Universitas Muhammadiyah Kalimantan Timur



Submission date: 16-Jul-2024 02:26PM (UTC+0800)

Submission ID: 2417653969

File name: SKRIPSI-ADITYA_PRANATA_-_ADITYA_PRANATA.docx (948.62K)

Word count: 6186

Character count: 39062

SKRIPSI ADITYA PRANATA

ORIGINALITY REPORT

13% SIMILARITY INDEX	10% INTERNET SOURCES	7% PUBLICATIONS	5% STUDENT PAPERS
--------------------------------	--------------------------------	---------------------------	-----------------------------

PRIMARY SOURCES

1	Untung Surapati, Ali Yafi Zulkarnain. "Implementasi Metode Naïve Bayes Untuk Mendeteksi Hate Speech Pada Twitter", INTECOMS: Journal of Information Technology and Computer Science, 2023 Publication	1%
2	Submitted to Sriwijaya University Student Paper	1%
3	123dok.com Internet Source	1%
4	ejournal.ust.ac.id Internet Source	1%
5	dspace.umkt.ac.id Internet Source	1%
6	media.neliti.com Internet Source	<1%
7	journal.maranatha.edu Internet Source	<1%
8	Submitted to Academic Library Consortium Student Paper	