KLASIFIKASI SENTIMEN PERSEPSI PENGGUNA APLIKASI TRANS SEMARANG MENGGUNAKAN *NAIVE BAYES CLASSIFIER* BERBASIS DATA ULASAN ONLINE

TUGAS AKHIR

Diajukan untuk Memenuhi Sebagian Syarat Guna
Memperoleh Gelar Sarjana Program Strata Satu (S-1)
dalam Ilmu Teknologi Informasi

Oleh:

ILMA SALSABILA

NIM: 2108096029

PROGRAM STUDI S-1 TEKNOLOGI INFORMASI

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS ISLAM NEGERI WALISONGO SEMARANG

2025

PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini:

Nama : **Ilma Salsabila**
NIM : 2108096029
Jurusan : Teknologi Informasi

Menyatakan bahwa skripsi yang berjudul :

Klasifikasi Sentimen Persepsi Pengguna Aplikasi Trans Semarang Menggunakan *Naive Bayes Classifier* Berbasis Data Ulasan Online

Secara keseluruhan adalah hasil penelitian/karya saya sendiri, kecuali bagian tertentu yang dirujuk sumbernya.

Semarang, 20 Juni 2025

Pembuat Pernyataan,

materai tempel
Rp. 10.000

Ilma Salsabila
NIM: 2108096029

KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI WALISONGO
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Prof. Dr. Hamka Ngaliyan Semarang
Telp.024-7601295 Fax.7615387

PENGESAHAN

Naskah skripsi berikut ini:

Judul : Klasifikasi Sentimen Persepsi Pengguna Aplikasi Trans Semarang Menggunakan Naïve Bayes Classifier Berbasis Data Ulasan Online
Penulis : **Ilma Salsabila**
NIM : 2108096029
Jurusan : Teknologi Informasi

Telah diujikan dalam sidang tugas akhir oleh Dewan Penguji Fakultas Sains dan Teknologi UIN Walisongo dan dapat diterima sebagai salah satu syarat memperoleh gelar sarjana dalam Ilmu Teknologi Informasi.

Semarang, 03 Juli 2025

DEWANPENGUJI

Penguji I,

Dr. Khotibul Umam, M.Kom
NIP. 197908272011011007

Penguji II,

Dr. Wenty Dwi Yuniarti, S.Pd., M.Kom
NIP. 197706222006042005

Penguji III,

Mokhamad Iklil Mustofa, M.Kom
NIP. 198808072019031010

Penguji IV,

Nur Cahyo Hendra Wibowo, S.T., M.Kom
NIP. 197312222006041001

Pembimbing I,

Dr. Wenty Dwi Yuniarti, S.Pd., M.Kom
NIP. 197706222006042005

Pembimbing II,

Masy Ari Ulinuha, M.T
NIP. 198108122011011007

NOTA DINAS

Semarang, 19 Juni 2025

Kepada
Yth. Ketua Program Studi Teknologi Informasi
Fakultas Sains dan Teknologi
UIN Walisongo Semarang

Assalamu'alaikum. wr. wb.

Dengan ini diberitahukan bahwa saya telah melakukan bimbingan, arahan dan koreksi naskah skripsi dengan:

Judul : KLASIFIKASI SENTIMEN PERSEPSI PENGGUNA APLIKASI TRANS SEMARANG MENGGUNAKAN NAIVE BAYES CLASSIFIER BERBASIS DATA ULASAN ONLINE
Penulis : **Ilma Salsabila**
NIM : 2108096029
Jurusan : Teknologi Informasi

Saya memandang bahwa naskah skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo untuk diujikan dalam Sidang Munaqosyah.

Wassalamu'alaikum. wr. wb.

Pembimbing I,

Dr. Wenty Dwi Yuniarti, S.Pd., M.Kom
NIP : 197706222006042005

NOTA DINAS

Semarang, 19 Juni 2025

Kepada
Yth. Ketua Program Studi Teknologi Informasi
Fakultas Sains dan Teknologi
UIN Walisongo Semarang

Assalamu'alaikum. wr. wb.

Dengan ini diberitahukan bahwa saya telah melakukan bimbingan, arahan dan koreksi naskah skripsi dengan:

Judul : KLASIFIKASI SENTIMEN PERSEPSI PENGGUNA APLIKASI TRANS SEMARANG MENGGUNAKAN NAIVE BAYES CLASSIFIER BERBASIS DATA ULASAN ONLINE
Penulis : **Ilma Salsabila**
NIM : 2108096029
Jurusan : Teknologi Informasi

Saya memandang bahwa naskah skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo untuk diujikan dalam Sidang Munaqosyah.

Wassalamu'alaikum. wr. wb.

ACC
23/2025
6

Pembimbing II,

Masy Ari Ulinuha, M.T
NIP : 198108122011011007

v

MOTTO DAN PERSEMBAHAN

MOTTO:

"Tidak ada mimpi yang terlalu tinggi. Tak ada mimpi yang patut untuk diremehkan. Lambungkan setinggi yang kau inginkan dan gapailah dengan selayaknya yang kau harapkan"

-Maudy Ayunda-

"Orang lain ga akan bisa faham *struggle* dan masa sulit nya kita yang mereka ingin tahu hanya bagian *success stories*. Berjuanglah untuk diri sendiri walaupun tidak ada yang tepuk tangan. Kelak diri kita dimasa depan akan sangat bangga dengan apa yang kita perjuangkan hari ini"

"Segala sesuatu yang telah diawali, maka harus diakhiri"

-Rizka Maryaningish-

PERSEMBAHAN:

Laporan skripsi ini saya dedikasikan dengan penuh cinta dan rasa syukur kepada kedua orang tua tercinta, Bapak Slamet Muthohirin dan Ibu Khoiriyah, atas doa yang tak pernah henti dan putus serta semangat dan dukungan yang tak ternilai. Juga untuk orang terdekatku yang tersayang, yang selalu memberi motivasi dan dukungan penuh. Terima kasih telah menjadi bagian dari proses panjang ini hingga akhirnya skripsi ini dapat diselesaikan dengan baik.

KLASIFIKASI SENTIMEN PERSEPSI PENGGUNA APLIKASI TRANS SEMARANG MENGGUNAKAN NAÏVE BAYES CLASSIFIER BERBASIS DATA ULASAN ONLINE

Oleh:
Ilma Salsabila
NIM. 2108096029

ABSTRAK

Aplikasi Trans Semarang merupakan layanan digital yang dirancang untuk memudahkan Masyarakat kota Semarang dalam mengakses informasi terkait transportasi umum Bus Rapid Transit (BRT). Namun, banyaknya ulasan yang terus bertambah dari para pengguna aplikasi menjadi tantangan tersendiri dalam memahami persepsi pengguna jika dilakukan secara manual. Pada penelitian ini bertujuan untuk mengklasifikasikan sentimen persepsi pengguna terhadap aplikasi Trans Semarang berdasarkan ulasan yang diperoleh dari *Google Play Store* ke dalam dua kategori sentimen, positif dan negatif. Data yang digunakan dalam penelitian ini dikumpulkan menggunakan teknik *web scrapping* pada ulasan apliaksi Trans Semarang, kemudian dilakukan tahapan pelabelan ulasan secara manual dan *preprocessing teks*. Dilanjutkan dengan pembobotan kata menggunakan metode TF-IDF. Selanjutnya proses klasifikasi dilakukan menggunakan metode *Naïve Bayes Classifier* dengan dua kelas sentimen. Pengujian pada penelitian ini dilakukan dengan membagi data *training* dan data *testing* dengan rasio 80:20. Hasil pengujian menunjukkan bahwa algoritma *Naïve Bayes Classifier* menghasilkan performa model yang baik, dengan nilai akurasi sebesar 82%, *precission* sebesar 81%, *recall* sebesar 82%, dan *f1-score* sebesar 81%.

Kata kunci: *Klasifikasi Sentimen, Trans Semarang, Naïve Bayes Classifier, Google Play Store*

KATA PENGANTAR

Alhamdulilah segala puja dan puji syukur dipanjatkan kepada Allah SWT atas segala berkat, rahmat, hidayah dan karunia-Nya sehingga penulis dapat menyelesaikan penyusunan laporan skripsi dengan judul **"Klasifikasi Sentimen Persepsi Pengguna Aplikasi Trans Semarang Menggunakan *Naive Bayes Classifier* Berbasis Data Ulasan Online"**. Penulis menyadari bahwa penyusunan skripsi ini tidak akan terwujud tanpa bantuan, bimbingan serta dukungan dari berbagai pihak, tidaklah mungkin untuk menyelesaikan skripsi ini dengan baik. Oleh karena itu, penulis mengucapkan banyak terima kasih kepada semua pihak yang telah membantu diantaranya:

1. Bapak Prof. Dr. Nizar, M.Ag, selaku Rektor Universitas Islam Negeri Walisongo Semarang.
2. Bapak Dr. Khotibul Umam, S.T., M.Kom, selaku Ketua Program Studi Teknologi Informasi Universitas Islam Negeri Walisongo Semarang.
3. Ibu Dr. Wenty Dwi Yuniarti, S.Pd., M.Kom, selaku dosen pembimbing skripsi I penulis yang telah memberikan bimbingan, arahan, saran, dan dukungan dalam proses penyusunan skripsi ini.
4. Bapak Dr. Masy Ulinuha, ST., M.T, selaku dosen pembimbing skripsi II penulis yang juga telah membimbing dan memberi arahan selama proses penyusunan skripsi.
5. Kedua orang tua tercinta, Bapak Slamet Muthohirin dan Ibu Khoiriyah yang selalu memberikan perhatian, nasihat, dukungan, motivasi, dan do'a yang tiada henti kepada penulis hingga terwujudnya laporan skripsi ini. Serta kakak tercinta penulis Ade Izyuddin S.Kom yang selalu mendukung penuh dan memberikan support kepada penulis, serta segenap keluarga tercinta yang selalu memberikan semangat dan dukungannya kepada penulis.
6. Teman-teman sahabat seperjuangan yang selalu

menemani, membantu hingga memberikan dorongan, semangat, motivasi dan support kepada penulis dalam menyelesaikan proses skripsi ini.
7. Teman-teman satu angkatan prodi Teknologi Informasi 2021 yang telah memberikan semangat dan dukungan selama menempuh studi di Universitas Islam Negeri Walisongo Semarang.
8. Serta semua pihak yang telah terlibat dan tidak dapat disebutkan satu persatu oleh penulis dalam penyusunan laporan skripsi ini hingga dapat terselesaikan dengan baik.

Penulis menyadari bahwa penyusunan skripsi ini masih jauh dari kata sempurna. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi penyempurnaan penelitian ini ditahap selanjutnya. Semoga skripsi ini dapat bermanfaat bagi penulis dan seluruh pihak yang membacanya.

Semarang, 20 Juni 2025
Penulis,

Ilma Salsabila
Nim. 210809602

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I
PENDAHULUAN

A. Latar Belakang

Perkembangan teknologi informasi yang semakin pesat telah mendorong berbagai sektor untuk memanfaatkan teknologi digital, termasuk dalam bidang transportasi publik (Muttaqin & Kharisudin, 2021). Baik angkutan pribadi maupun angkutan umum, transportasi tak terlepas dari bagian kehidupan sehari-hari. Diera sekarang beberapa masyarakat lebih memilih menggunakan kendaraan pribadi seperti kendaraan roda dua atau roda empat untuk melakukan aktivitasnya. Namun, peningkatan jumlah pengguna kendaraan menyebabkan kemacetan yang semakin parah pada kota-kota besar di indonesia. Oleh karena itu, pemerintah sekarang mendorong masyarakat agar memilih menggunakan transportasi umum (Hasanah & Sari, 2024). Hal tersebut menekan polusi udara, mengurangi konsumsi minyak dan energi, serta membantu mengurangi kemacetan lalu lintas yang sekarang menjadi masalah di kota-kota besar (Chintia, 2018).

Transportasi memiliki peran penting dalam mendukung mobilitas barang dan manusia dari satu tempat ke tempat lain. Di Kota Semarang, penggunaan transportasi umum telah menjadi solusi utama untuk mengurangi kemacetan lalu lintas (Fachreza & Handoko, 2024). Bus Rapid Transit (BRT) atau

dikenal sebagai Trans Semarang, pertama kali diluncurkan tanggal 18 September 2009. Kini, bus Trans Semarang telah menjadi opsi transportasi umum utama di Kota Semarang (Nursalim Nursalim & Agus Windu Sancono, 2023). Awalnya, pengelolaan Trans Semarang dilakukan oleh Pemerintah Kota Semarang yang bekerja sama dengan PT Trans Semarang. Namun, sejak tahun 2017, pengelolaan tersebut beralih ke BLU (Badan Layanan Umum) UPTD (Unit Pelaksana Teknis Daerah) Trans Semarang (Ardiana & Erawan, 2019).

Semenjak pengelolaan dipegang oleh badan layanan umum (BLU) UPTD Trans Semarang, Inovasi transportasi pelayanan publik pada sistem transportasi bus Trans Semarang atau BRT, dilengkapi dengan adanya fitur e-ticketing dan sebuah aplikasi Trans Semarang berbasis playstore yang dapat diinstal di smartphone untuk mengetahui informasi mengenai jadwal kedatangan bus serta berfungsi sebagai platform bagi masyarakat untuk menyampaikan kritik dan saran nya terkait layanan Trans Semarang. Aplikasi tersebut diluncurkan bersamaan dengan perayaan HUT ke 472 Kota Semarang pada tanggal 2 Mei 2019 di halaman kantor Balaikota Semarang (Chintia, 2018).

Aplikasi Trans Semarang adalah salah satu inovasi baru dalam layanan transportasi yang dirancang untuk memberikan kemudahan bagi pengguna transportasi umum di Kota Semarang dalam mengetahui informasi mengenai rute

berdasarkan koridor, posisi bus dan shelter, hingga kedatangan bus saat penumpang sedang menunggu dishelter serta layanan-layanan lainnya. Namun, keberhasilan sebuah aplikasi tidak hanya ditentukan oleh aspek fungsionalitasnya saja, tetapi juga berdasarkan tingkat kepuasan dan persepsi pengguna nya terhadap aplikasi tersebut.

Data ulasan online akan diambil dari sebuah platform di Google Play Store. Setiap pengguna di Google Play Store, memiliki kesempatan untuk memberikan ulasan dan penilaiannya terhadap sebuah aplikasi. Ulasan-ulasan yang ada di platform tersebut diberikan oleh pengguna untuk mengukur tingkat kepuasan dan ketidakpuasan terhadap layanan aplikasi. Ulasan ini dapat memberikan gambaran yang jelas untuk pengembang tentang bagaimana persepsi pengguna terhadap layanan aplikasi ini dengan mengetahui apakah layanan aplikasi telah memenuhi harapan pengguna, atau justru masih terdapat kekurangan yang harus diperbaiki. (Febriyanti, 2020).

Saat ini, aplikasi Trans Semarang memiliki rating 3,3 dari skala 5 bintang, dan beberapa pengguna masih menyampaikan kritik terkait fitur dan kinerja aplikasi. Di sisi lain, banyak juga ulasan yang menunjukkan komentar positif terhadap layanan aplikasi ini. Ulasan-ulasan tersebut memberikan informasi penting yang dapat digunakan pengembang untuk memperbaiki dan meningkatkan layanan aplikasi sesuai

dengan kritik dan saran yang disampaikan oleh para pengguna nya (Jiana & Hartono, 2024).

Namun, seiring dengan peningkatan jumlah pengguna aplikasi Trans Semarang juga menyebabkan meningkatnya volume ulasan yang masuk ke platform Google Play. Jumlah ulasan yang sangat banyak menjadi tantangan tersendiri bagi pengembang aplikasi karena tidak memungkinkan untuk diklasifikasi secara manual oleh pengembang secara efisien. Selain itu, ulasan yang tersebar secara acak, tidak terkelompok berdasarkan kategori sentimen, semakin menyulitkan dalam memahami persepsi pengguna secara keseluruhan. Akibatnya, pengembang aplikasi dapat kesulitan dalam mengetahui bagian mana yang perlu diperbaiki atau dipertahankan. Maka perlunya klasifikasi yang dapat mengelompokkan persepsi pengguna menjadi dua kategori sentimen.

Untuk mengatasi permasalahan ini, pendekatan *Text Mining* dapat diterapkan, khususnya melalui teknik klasifikasi sentimen. *Text Mining* ialah teknik untuk mencari informasi dari data berbasis teks. Salah satu pendekatan *text mining* yaitu klasifikasi sentimen, yang bertujuan untuk mengklasifikasikan opini atau perasaan menjadi beberapa kategori (Febriyanti, 2020). Klasifikasi sentimen memungkinkan ulasan pengguna dikelompokkan secara otomatis ke dalam kategori sentimen seperti positif atau negatif, sehingga pengembang dapat mengetahui pola umum persepsi pengguna terhadap layanan

aplikasi. Dengan cara ini, informasi dari ulasan yang tidak terstruktur dapat diolah menjadi data yang berguna dalam pengembangan aplikasi.

Penelitian ini dilakukan untuk mengklasifikasikan sentimen persepsi umum pengguna aplikasi Trans Semarang berdasarkan ulasan-ulasan di Google Play yang dikategorikan menjadi kategori positif dan negatif. Melalui pengelompokan ini, diharapkan dapat diketahui persepsi umum pengguna, aspek layanan yang dinilai baik, serta bagian mana yang masih membutuhkan perbaikan. Tujuan dari pengklasifikasian ini untuk mengelompokkan pola persepsi pengguna terhadap layanan aplikasi Trans Semarang. Dengan memahami persepsi dan penilaian pengguna aplikasi, data yang diperoleh dapat dimanfaatkan untuk mengidentifikasi kelebihan dan kekurangan layanan. Hal ini memungkinkan perusahaan untuk mengevaluasi fitur-fitur yang dinilai positif oleh pengguna, serta menentukan area yang memerlukan perbaikan untuk meningkatkan kualitas aplikasi (Junianto et al., 2024). Hal tersebut sesuai dengan Al-Qur'an surah Al-Furqan ayat 75 yang berbunyi:

أُولَٰئِكَ يُجْزَوْنَ الْغُرْفَةَ بِمَا صَبَرُوْا وَيُلَقَّوْنَ فِيْهَا تَحِيَّةً وَّسَلٰمًاۙ

Artinya:

"Mereka itulah orang yang dibalasi dengan martabat yang tinggi (dalam surga) karena kesabaran mereka dan mereka disambut dengan penghormatan dan ucapan selamat di dalamnya." (QS. Al-Furqan: Ayat 75).

Dari terjemahan diatas menunjukkan bahwa kesabaran akan dibalas dengan balasan yang tinggi, berupa kehormatan dan tempat yang mulia di surga. Sama seperti ayat tersebut dalam klasifikasi sentimen, perusahaan yang mampu mendengarkan dan memperbaiki kekurangan berdasarkan persepsi dan kritik dengan kesabaran juga akan dihargai dalam bentuk peningkatan kepuasaan pengguna. Dalam ayat ini, kesabaran dapat diartikan sebagai upaya gigih perusahaan untuk terus meningkatkan layanan sesuai dengan persepsi yang diterima.

Dalam pengklasifikasian sentimen, penulis menggunakan algoritma *naïve bayes classifier* (NBC) dengan bantuan bahasa pemrograman *python*. *Naive bayes classifier* (NBC) ialah teknik pembelajaran mesin berbasis probabilistik. Meskipun operasinya sederhana, algoritma ini menghasilkan tingkat akurasi dan performa yang tinggi dalam klasifikasi teks (LING et al., 2014). Salah satu ciri metode *Naïve Bayes* adalah asumsi yang sangat kuat (naif) dimana setiap kondisi atau kejadian dianggap saling independen satu sama lain (Yuniarti et al., 2020).

Berdasarkan uraian yang telah dijabarkan, penulis akan melakukan klasifikasi sentimen presepsi pengguna terhadap aplikasi Trans Semarang menggunakan algoritma *Naïve Bayes Classifier* berdasarkan data ulasan online yang dikumpulkan dari Google Play Store. Dari adanya penelitian ini, diharapkan

dapat diperoleh gambaran umum tentang persepsi pengguna, serta membantu mengidentifikasi kelebihan dan kekurangan layanan aplikasi, dan menjadi acuan bagi pengembang aplikasi dalam meningkatkan kualitas layanan yang sesuai persepsi pada kebutuhan dan harapan pengguna.

B. Identifikasi Masalah

Berdasarkan latar belakang yang telah diuraikan maka masalah yang diidentifikasi dalam penelitian ini adalah sebagai berikut:

1. Banyaknya ulasan dari pengguna membuat proses pemahaman persepsi pengguna menjadi sulit jika dilakukan secara manual
2. Ulasan yang tersebar secara acak di Google Play Store menyulitkan pengelompokan opini pengguna secara sistematis berdasarkan kategori sentimen

C. Rumusan Masalah

Berdasarkan uraian latar belakang yang telah dijabarkan maka pokok masalah pada penelitian ini dapat dirumuskan sebagai berikut:

1. Bagaimana *Naïve Bayes Classifier* digunakan untuk mengidentifikasi sentimen pengguna terhadap aplikasi Trans Semarang?

2. Bagaimana tingkat akurasi *Naive bayes Classifier* dalam mengklasifikasikan sentimen persepsi pengguna aplikasi Trans Semarang?

D. Pembatasan Masalah

Berdasarkan perumusan masalah yang telah dijelaskan batasan masalah dalam penelitian ini adalah:

1. Data ulasan online yang akan diklasifikasi berasal dari data ulasan yang ada di Google Play Store.
2. Jumlah Data yang diambil dari Google Play Store dibatasi sampai dengan tanggal 17 Juni 2025.
3. Hasil klasifikasi sentimen dengan menggunakan metode *naïve bayes classifier* akan menghasilkan 2 kategori sentimen positif dan negatif.
4. Klasifikasi dilakukan dengan menerapkan bahasa pemrograman *python* yang dijalankan di Google *Colab*.

E. Tujuan Penelitian

Tujuan dari adanya penelitian ini adalah sebagai berikut:

1. Menerapkan *Naïve Bayes Classifier* dalam mengidentifikasi sentimen pengguna aplikasi Trans Semarang ke dalam kategori positif dan negatif berdasarkan ulasan di Google Play Store.

2. Mengukur tingkat akurasi *Naïve Bayes Classifier* dalam mengklasifikasikan sentimen persepsi pengguna aplikasi Trans Semarang.

F. Manfaat Penelitian

Dalam penelitian ini, penulis berharap agar hasilnya dapat memberikan manfaat bagi para pembaca, yang dapat berupa manfaat teoritis dan manfaat praktis.

1. **Manfaat Teoritis:**
   a. Mengetahui persepsi umum pengguna aplikasi Trans Semarang berdasarkan data ulasan di Google Play.
   b. Membantu mengklasifikasikan tanggapan pengguna berupa positif dan negatif.
   c. Mengetahui tingkat akurasi pada algoritma *Naïve Bayes Classifier* dalam mengklasifikasikan sentimen pengguna Trans Semarang.
   d. Hasil penelitian dapat dijadikan acuan bagi penelitian-penelitian selanjutnya yang berkaitan dengan klasifikasi sentimen atau analisis ulasan online.
2. **Manfaat Praktis:**
   a. Bagi Masyarakat Pengguna Aplikasi:
      Dapat memberikan informasi yang lebih transparan mengenai persepsi pengguna terhadap aplikasi Trans Semarang.

b. Bagi Perusahaan/Pengembang Aplikasi Trans Semarang:

Memberikan wawasan yang berguna bagi pengembang aplikasi Trans Semarang dalam memahami kebutuhan dan harapan pengguna sehingga dapat membantu dalam evaluasi dan perbaikan layanan aplikasi.

c. Bagi perusahaan Transportasi Sejenis:

Dapat menjadi refrensi bagi perusahaan transportasi lain yang ingin meningkatkan kualitas layanan mereka melalui analisis ulasan pengguna.

G. Sistematika Penulisan

Laporan penelitian ini disusun dalam beberapa bab agar pembaca dapat dengan mudah memahami keseluruhan isi laporan. Berikut adalah urutan penulisan yang digunakan dalam penelitian ini:

BAB I PENDAHULUAN:

Pada bab pendahuluan dijelaskan mengenai latar belakang dilakukannya penelitian ini, serta mencakup rumusan masalah penelitian, batasan masalah, manfaat penelitian, dan juga sistematika penulisan laporan.

BAB II LANDASAN PUSTAKA

Pada bab landasan pustaka memuat tentang kajian teori yang mendukung pelaksanaan penelitian ini, serta menguraikan teori-teori dasar yang relevan dengan

penelitian, terutama yang berkaitan dengan klasifikasi sentimen menggunakan *Naive Bayes Classifier*.

BAB III METODOLOGI PENELITIAN

Pada bab metodologi penelitian memuat tentang bagaimana peneliti mengumpulkan data, bagaimana dataset yang didapatkan pada penelitian ini, kemudian menguraikan alat dan perangkat yang digunakan dalam penelitian ini, serta membahas tahapan alur pelaksanaan penelitian dan gambaran umum dari metode yang digunakan.

BAB IV HASIL DAN PEMBAHASAN

Pada hasil dan pembahasan memuat tentang hasil penelitian yang telah diperoleh, diikuti dengan analisis dan pembahasan terkait temuan yang dihasilkan dari penelitian.

BAB V KESIMPULAN DAN SARAN

Pada bab terakhir ini menyajikan kesimpulan hasil penelitian serta memberikan saran yang relevan bagi penelitian selanjutnya atau pengembangan lebih lanjut.

BAB II
LANDASAN PUSTAKA

A. Kajian Teori

1. Klasifikasi Sentimen

Klasifikasi sentimen adalah bagian dari analisis semtimen yang merupakan suatu pendapat seseorang pada sebuah isu, topik, layanan, atau golongan tertentu yang diekspresikan dalam sebuah teks. Teks tersebut dapat dikategorikan menjadi beberapa kelas yaitu kelas positif, negatif, atau netral (Kevin et al., 2024). Bentuk teks yang dianalisis dalam klasifikasi sentimen mulai dari keseluruhan dokumen, paragraf, kalimat, atau teks yang lebih pendek (Indransyah et al., 2022). Klasifikasi sering kali menggunakan metode pembelajaran mesin (*machine learning*) dan teknik pemrosesan bahasa alami (NLP) untuk mendeteksi emosi yang terkait dalam suatu teks.

2. Analisis Sentimen

Analisis sentimen ialah teknik mengekstrak data berbentuk teks guna memperoleh informasi tentang sentimen yang bernilai positif, negatif, atau netral. Tujuan dari teknik ini agar setiap orang

dapat memberikan penilaian terhadap sebuah aplikasi atau postingan pada sebuah media sosisal (Hasanah & Sari, 2024). Menurut Liu 2012, dalam (M. N. Akbar & Nirwana Samrin, 2023), analisis sentimen adalah salah satu bidang dalam text mining dalam penerapan analisis teks dan pemrosesan bahasa alami (*natural language processing*) yang mempelajari pendapat, sentimen, penilaian, sikap, serta emosi seseorang terhadap suatu hal yang diungkapkan melalui teks tertulis.

Analisis sentimen atau dikenal sebagai *opinion mining* yaitu proses pengumpulan, pengolahan, dan analisis terhadap suatu ulasan pengguna tentang suatu produk atau topik tertentu yang disampaikan melalui postingan blog, komentar, ulasan, atau tweet (Chairunnisa et al., 2022).

3. Online Review

Keluhan dan saran yang disampaikan melalui berbagai platform media, seperti kuesioner tertulis atau ulasan online, adalah salah satu dari empat cara yang digunakan untuk menilai tingkat kepuasan pengguna. Pada penelitian ini, ulasan online didefinisikan sebagai ulasan atau komentar yang dibuat oleh pengguna tentang persepsi dan

evaluasi pada sebuah aplikasi yang dilakukan melalui perangkat yang terhubung dengan internet seperti komputer (Febriyanti, 2020).

Dalam penelitian ini, data *review* pengguna dalam menilai ulasan aplikasi Trans Semarang dikumpulkan melalui situs Google Play Store. Komentar pengguna terdiri 2 bagian yaitu Nilai peringkat dan komentar teks:

a. Nilai peringkat menunjukkan peringkat numerik dari pengalaman pengguna secara keseluruhan
b. Komentar teks dapat memberikan pemahaman lebih lanjut tentang situasi (M. N. Akbar et al., 2022).

4. Trans Semarang

Transportasi umum Trans Semarang merupakan sistem transportasi bus yang menyediakan layanan murah dan cepat di Kota Semarang. Program ini merupakan implementasi dari Bus Rapid Transit (BRT) yang diinisiasi oleh Departemen Perhubungan. Trans Semarang pertama kali diuji coba pada 2 Mei 2009, bertepatan dengan ulang tahun kota Semarang yang ke-462. Setelah diuji coba tersebut, kabar mengenai

peluncuran resmi Trans Semarang sempat tidak terdengar. Namun, setelah beberapa waktu, Trans Semarang akhirnya resmi diluncurkan pada 18 September 2009. Selain itu, Transportasi bus Trans Semarang juga dilengkapi dengan hadirnya sebuah aplikasi yaitu aplikasi "Trans Semarang". Aplikasi tersebut juga diluncurkan bersamaan dengan HUT kota semarang yang ke 472. Aplikasi Trans Semarang resmi dirilis diplay store pada tanggal 23 April 2019 (Nursalim Nursalim & Agus Windu Sancono, 2023).

Gambar 2. 1 *Logo Aplikasi Trans Semarang*
(Sumber: https://id.wikipedia.org/wiki/)

Trans Semarang memiliki 7 jalur atau rute yang meliputi Koridor I (Terminal Mangkang sampai Terminal Penggaron), Koridor II (Terminal Terboyo sampai Terminal Sisemut), Koridor III (Pelabuhan Tanjung Emas – Rumah Sakit Elisabeth

– Akademi Kepolisian), Koridor IV (Terminal Cangkiran – Bandara – Stasiun Tawang), Koridor V (Meteseh – PRPP), Koridor VI (UNNES – UNDIP), serta Koridor VII (Genuk – Balaikota) (Ardiana & Erawan, 2019). Meskipun ada lebih dari 200 halte dan titik berhenti di setiap koridor, banyak pengguna Trans Semarang yang tidak tahu cara berpindah antar koridor dan rute yang dilalui sebelum aplikasi tersebut dirilis. Namun setelah adanya aplikasi Trans Semarang, membuat masyarakat lebih mudah dalam mengetahui informasi jalur tersebut.

5. Google Play Store

Google Play Store yaitu platform resmi yang memberikan akses kepada pengguna di seluruh dunia untuk mengunduh berbagai aplikasi, game, musik, film, dan buku pada perangkat Android. Google Play berfungsi sebagai toko aplikasi resmi untuk semua perangkat yang menjalankan sistem operasi Android. Pengguna dapat melihat serta menginstall aplikasi yang dikembangkan dengan Android SDK atau perangkat lunak pengembangan yang disediakan oleh Google melalui situs webnya atau mengunduh konten langsung dari perangkat

Android. (Hakim, 2021).

Google Play dirilis pertama kalinya pada 22 Oktober 2008 sebagai *Android market*. Pada Maret 2012, Google Play dirilis sebagai pengganti Android Market dan layanan musik Google, serta menjadi platform dengan jumlah aplikasi terbanyak. Google Play memiliki banyak fitur yang memungkinkan pengguna meninggalkan komentar tentang aplikasi yang mereka gunakan. (Hakim, 2021).

6. Web Scrapping

Web scraping merupakan metode untuk mengambil atau mengekstraksi data secara otomatis dari situs web yang membuat pencarian data atau informasi yang diperlukan menjadi lebih mudah di internet. Data yang digunakan didapatkan dari ulasan Google Play, yang mencakup teks ulasan dan rating. Proses *web scraping* melibatkan pengambilan data dari situs *web* dengan bantuan alat kemudian mengekstrak data tersebut dan menyimpannya dalam format *csv* atau file *Excel* yang terpisah (Muhammadiyah et al., 2024).

Web Scraping menurut (Zhao, 2020), adalah metode untuk mengekstraksi data dari *World Wide*

Web (WWW) dan menyimpannya ke dalam sistem file atau basis data untuk diambil atau dianalisis lebih lanjut. Data web umumnya diambil melalui *Hypertext Transfer Protokol* (HTTP) atau browser web. Proses ini dapat dilakukan secara manual atau otomatis menggunakan bot atau *web* crawler. Sedangkan menurut (Turland, 2010 dalam (Tiara et al., 2021)), w*eb scrapping* adalah proses pengambilan dokumen semi-terstruktur dari internet, yang biasanya berupa halaman *web* dalam format bahasa markup seperti *HTML* atau *XHTML* untuk menganalisis dokumen tersebut dan mengekstrak data spesifik dari halaman tersebut untuk keperluan lainnya.

7. Text Mining

Text mining adalah metode ekstraksi data yang digunakan untuk menganalisis dan mengklasifikasikan dokumen atau berbentuk teks dengan tujuan untuk mengidentifikasi pola serta informasi yang relevan untuk tujuan tertentu, sehingga memungkinkan analisis hubungan antar dokumen tersebut (Giovani et al., 2020). *Text mining* merupakan proses mengubah teks yang tidak terstruktur menjadi teks yang terstruktur,

serta mengidentifikasi pola untuk menemukan informasi atau wawasan baru yang sebelumnya belum diketahui dari data teks (Atimi & Enda Esyudha Pratama, 2022).

Dalam (Zanini & Dhawan, 2015), menyebutkan bahwa *Text Mining* didefinisikan sebagai sekumpulan ilmu komputer dan teknik statistik yang dirancang khusus untuk menganalisis data teks. Tujuan utama dari *text mining* untuk mengekstrak indeks numerik yang relevan dari teks dan mengolah informasi tidak terstruktur yang terkandung dalam teks, sehingga teks tersebut dapat diakses oleh algoritma statistik data mining.

Dalam proses *text mining*, dokumen text yang akan digunakan harus dipersiapkan sebelum memulai ke tahap utama. Tahapan persiapan dokumen teks atau dataset mentah dikenal dengan *text preprocessing* (Irsyad & Geralda, 2023). Secara umum, *preprocessing* data dilakukan dengan menghilangkan data yang tidak penting agar data tersebut menjadi lebih terstruktur sehingga dapat diolah lebih lanjut (Pratmanto et al., 2023). Berikut beberapa proses tahapan yang dilakukan dalam *text preprocessing*:

a. Cleansing

   Tahapan *cleansing* dilakukan untuk memperbaiki dan meningkatkan kualitas data dengan menghapus noise atau elemen yang tidak penting dari dokumen (Soraya & Indriyanti, 2024).

b. Case Folding

   Tahapan *case folding* dilakukan untuk mengubah semua huruf besar/kapital menjadi huruf kecil (Soraya & Indriyanti, 2024).

c. Remove Duplicate

   Tahapan *remove duplicate* dilakukan untuk menghilangkan data yang memiliki duplikat atau ganda yang dinilai sama.

d. Tokenizing

   Tahapan *tokenizing* dilakukan untuk membagi rangkaian karakter menjadi bagian lebih kecil (kata) yang dikenal sebagai token, dengan tujuan untuk mengumpulkan fitur dari setiap dokumen setelah diberi nilai (Soraya & Indriyanti, 2024).

e. Normalisasi Kata

   Tahapan *normalisasi* dilakukan untuk mengubah atau menggantikan kata-kata yang salah eja atau disingkat yang sesuai dengan

aturan tata bahasa Indonesia atau KBBI. Tujuannya adalah untuk mengurangi jumlah dimensi kata yang berbeda yang dapat muncul sebagai akibat dari kesalahan ejaan atau singkatan yang tidak diperbaiki. Dalam matriks, kata-kata ini dianggap sebagai entitas yang berbeda meskipun memiliki arti yang sama jika tidak dinormalisasi (Soraya & Indriyanti, 2024).

f. Stopword Removal

Tahapan proses *stopword removal* digunakan untuk mengeliminasi kata-kata yang tidak memberikan dampak signifikan pada proses klasifikasi, seperti kata depan dan kata sambung. Proses ini menghasilkan wordlist yang hanya mengandung kata-kata penting yang relevan dalam klasifikasi (Mustofa & Mahfudh, 2019).

g. Stemming

Stemming adalah teknik yang digunakan mengubah kata-kata dalam dokumen teks menjadi bentuk dasarnya. Tahapan stemming dilakukan dengan tujuan untuk mengurangi jumlah total entitas berbeda yang ada di dataset dan mengelompokkan kata-kata dasar

yang memiliki makna serupa, meskipun memiliki bentuk yang berbeda karena adanya imbuhan. Dalam setiap bahasa, penggunaan kata berimbuhan memiliki aturan berbeda. Dalam bahasa Indonesia, kompleksitas stemming mengacu pada variasi imbuhan. Hal ini menjadi penting untuk membangun kata dasar (Mustofa & Mahfudh, 2019).

8. Klasifikasi

Prasetyo (2012, dalam (Afifi, 2022)), menyatakan bahwa klasifikasi adalah proses menilai objek data sehingga dapat dimasukkan dalam kelas tertentu dari jumlah kelas yang tersedia. Menurut Han & Kamber (2006, dalam (Marisa et al., 2021)), Klasifikasi yaitu teknik analisis data yang digunakan untuk membuat model prediksi yang berfungsi untuk mendeskripsikan label atau kelas data.

Klasifikasi merupakan sebuah proses menemukan model untuk menjelaskan atau membedakan ide atau kelas data dari objek yang labelnya belum diketahui dengan menggunakan aturan atau fungsi tertentu. Model ini bisa berupa pohon keputusan, aturan “jika-maka”, atau rumus

matematis (Atimi & Enda Esyudha Pratama, 2022). Dalam penelitian ini, pengklasifikasian dengan menggunakan 3 kelas yaitu kelas positif, negatif, dan netral.

9. Splitting Data

Splitting data yaitu proses membagi data menjadi dua bagian, yaitu dataset latih (training data) dan dataset uji (testing data). Data latih digunakan untuk melatih model dan membantu model mempelajari fitur atau pola tersembunyi dalam data, sedangkan data uji digunakan untuk mengevaluasi kinerja model setelah proses pelatihan selesai (Ernianti Hasibuan & Elmo Allistair Heriyanto, 2022). Pada penelitian yang akan dilakukan, penulis akan membagi data dengan rasio 80:20, Dimana 80% digunakan sebagai data latih dan 20% sisanya digunakan sebagai data uji. Penelitian dengan rasio tersebut juga pernah dilakukan sebelumnya oleh (Rieuwpassa et al., 2024) dan menghasilkan jumlah nilai akurasi yang baik.

10. Pembobotan Fitur TFIDF

Pembobotan kata merupakan proses

menghitung skor untuk setiap kata dalam suatu dokumen berdasarkan frekuensinya. Pembobotan ini dilakukan dengan menggunakan TF-IDF (Hamka et al., 2022). TF-IDF (*Term Frequency-Inverse Document Frequency*) yaitu teknik pembobotan yang tersusun dari dua nilai yang berasal dari dua algoritma yang berbeda yaitu TF dan IDF (M. A. Akbar & Solichin, 2024). TF ialah jumlah kata yang muncul dalam dokumen dibagi dengan total jumlah kata dalam dokumen secara keseluruhan, sementara IDF ialah teknik yang digunakan untuk mengurangi bobot suatu kata atau *term* jika kata tersebut tersebar di seluruh dokumen (Yolanda et al., 2023)

TF-IDF adalah sebuah proses yang menghasilkan transformasi data teks menjadi data numerik. Proses tersebut berfungsi memberikan bobot pada setiap kata dan fitur dan menunjukkan seberapa penting setiap kata dalam dokumen. Hasil tersebut merupakan bentuk dari perkalian antara TF dan IDF. Singkatnya, semakin besar bobot sebuah kata, semakin sering kata tersebut muncul dalam suatu dokumen. Sebaliknya, jika bobotnya semakin kecil, kata tersebut cenderung muncul di banyak dokumen (Hasanah & Sari, 2024).

Perhitungan pembobotan fitur TF-IDF dapat ditulis pada persamaan berikut:

Keterangan:

d : dokumen

t : kata

TF(t,d) : Jumlah kata tiap dokumen

$$TF - IDF(t,d) = TF(t,d).IDF(t) \qquad (2.1)$$

a. TF

Term Frequency (TF) ialah jumlah kemunculan kata pada tiap dokumen. Rumus TF dapat dinyatakan pada persamaan berikut (Mustofa & Mahfudh, 2019):

$$Tf = \frac{Jumlah\ frekuensi\ kata\ terpilih}{Jumlah\ kata} \qquad (2.2)$$

b. IDF

IDF yaitu banyaknya jumlah dokumen dalam suatu kata tersebut. Rumus IDF dapat dinyatakan pada persamaan berikut (Mustofa & Mahfudh, 2019):

$$IDF(t) = \log\left(\frac{N}{DFt}\right) + 1 \qquad (2.3)$$

Keterangan:

IDF(t,D): Banyaknya dokumen yang memuat t,D

N : Jumlah keseluruhan dokumen

DFt : Jumlah dokumen yang memuat kata t

11. Naïve Bayes Classifier

Naïve Bayes *Classifier* (NBC) merupakan salah satu metode machine learning yang bertujuan untuk menggunakan probabilitas atau peluang dari data untuk mengelompokkannya pada beberapa kelas. (Yolanda et al., 2023). Menurut (Rish, 2001), *Naïve Bayes Classifier* dapat didefinisikan sebagai sebuah *classifier Bayesian* yang mengasumsikan bahwa fitur-fitur independen satu sama lain diberikan kepada kelas tertentu. *Naïve Bayes Classifier* ini memperkirakan kelas yang paling mungkin dengan menghitung probabilitas posteriror dan kesederhaan model ini memungkinkan efesiensi dan keprkatisan yang tinggi diberbagai domain.

Naïve Bayes *Classifier* (NBC) memprediksi probabilitas suatu kejadian di masa depan berdasarkan data pada masa sebelumnya, yang sering dikenal sebagai penerapan *Teorema Bayes* (Febriyanti, 2020). Naive Bayes *Classifier* didasarkan pada konsep dasar dari *Teorema Bayes*,

yang pertama kali diperkenalkan oleh Thomas Bayes (Ramadhani & Suryono, 2024). Algoritma Naive *Bayes Classifier* memiliki tujuan untuk mengklasifikan data ke dalam kategori yang spesifik. Kelebihan dari Algoritma ini adalah operasinya yang sederhana namun mampu menghasilkan tingkat akurasi yang tinggi saat diaplikasikan dalam database yang besar (Mahfudh & Mustofa, 2019). Pada tahap penelitian ini, dataset diuji menggunakan klasifikasi Naïve Bayes *Classifier.*

Perhitungan *Teorema Bayes* yang digunakan memiliki bentuk yang sederhana, sehingga dapat menyederhanakan kompleksitas komputasi menjadi operasi perkalian probabilitas. Selain itu, algoritma ini juga dapat menangani data dengan berbagai atribut. Rumus naïve bayes dapat dilihat pada persamaan berikut (Hakim, 2021) :

$$P(H|X) = \frac{P(X|H).P(H)}{P(X)} \qquad (2.4)$$

Keterangan:

X = Data dengan kelas yang belum diketahui

H = Hipotesis bahwa data X termasuk ke dalam kelas spesifik

P(H|X) = Probabilitas hipotesis H berdasarkan kondisi X (*probabilitas posterior*)

P(H) = Probabilitas hipotesis H (*probabilitas prior*)

P(X|H) = Probabilitas X berdasarkan kondisi tersebut

P(X) = Probabilitas X

12. Evaluasi Model

Proses evaluasi dilakukan untuk mengukur performasi pada sebuah model yang sedang dibangun. Untuk melakukan pengujian terhadap sebuah model, perhitungan pada akurasi, *presisi, recall* dan *F-Score* dalam mengklasifikasikan sentimen dievaluasi dengan menggunakan *confusion matrix* (Soraya & Indriyanti, 2024). *Confusion matrix* adalah metode yang digunakan untuk menghitung atau mengukur tingkat akurasi dari hasil sistem yang telah dibangun (Fatmawati et al., 2024). Pada penelitian ini, menggunakan *confusion matrix* 2x2 untuk hasil sentimen yang diklasifikasi dua kelas, yaitu kelas positif dan negatif. Tabel *confusion matrix* dapat dilihat pada tabel 2.1 berikut (Astuti et al., 2024).

Tabel 2. 1 *Confusion Matrix 2x2*

		Predicted Class	
		Negatif	*Positif*
True Class	*Negatif*	T Neg	F Pos
	Positif	F Neg	T Pos

Terdapat beberapa nilai dalam *confusion matrix* yang dapat digunakan sebagai referensi untuk perhitungan. Nilai-nilai ini adalah sebagai berikut (Harahap, Eva Darwisah Kurniawan, 2024):

1. *True Positif* (T Pos), terjadi ketika jumlah data aktual adalah positif dan diprediksi sebagai positif atau sesuai.
2. *True Negatif* (T Neg), terjadi ketika jumlah data aktual adalah negatif dan diprediksi sebagai negatif atau sesuai.
3. *False Positif* (F Pos), terjadi ketika jumlah data aktual adalah negatif, tetapi diprediksi sebagai positif.
4. *False Negatif* (F Neg), terjadi ketika jumlah data aktual positif, tetapi diprediksi sebagai negatif.

Berdasarkan Tabel 2.1 tersebut, dapat dilakukan perhitungan lebih lanjut untuk

mengukur nilai tingkat akurasi, *recall, precision*, serta *f1-Score* pada confusion matrix 2x2 (Astuti et al., 2024).

a. Akurasi

Akurasi (*Accuracy*) merupakan tingkat keakuratan model dalam mengklasifikasikan data dengan benar. Persamaan 2.5 berikut menunjukkan perhitungan akurasi (Rahmadhani, 2021).

$$Accuracy = \frac{TPos+TNeg}{TPos+TNeg+FPos+FNeg} \quad (2.5)$$

b. Precision

Precision menggambarkan seberapa tepat model dalam memberikan hasil prediksi yang sesuai dengan data yang diminta. Nilai precision yang tinggi menunjukkan bahwa model memiliki tingkat *False Positive* (FP) yang rendah. Persamaan 2.6 berikut menunjukkan perhitungan *precision*.

$$Precision = \frac{TP}{TP + FP} \quad (2.6)$$

c. Recall

Recall merupakan perbandingan antara

jumlah pengamatan positif yang terprediksi dengan benar terhadap jumlah total pengamatan dikelas positif yang sebenarnya. Tingginya nilai recall menunjukkan rendahnya tingkat *False Negative*. Persamaan 2.7 berikut menunjukkan perhitungan *recall.*

$$Recall = \frac{TP}{TP + FN} \quad (2.7)$$

d. F1-Score

F1-Score merupakan matrik tunggal yang mengkombinasikan antara *precision* dan *recall* dihitung sebagai rata-rata harmonik dari keduanya. Persamaan 2.8 berikut menunjukkan perhitungan *FI-Score*.

$$F1 - Score = 2 . \frac{Precission * Recall}{Precission + Recall} \quad (2.8)$$

13. Visualisasi

Proses visualisasi hasil klasifikasi sentimen dilakukan setelah model Naïve Bayes dilatih. Visualisasi ini diterapkan pada setiap kelas sentimen bertujuan untuk mengumpulkan data informasi tentang topik-topik yang paling sering

dibicarakan dan diulas oleh pengguna. Oleh karena itu, informasi penting dapat ditemukan dari teks secara keseluruhan (Hakim, 2021).

Dalam penelitian ini, akan menampilkan hasil klasifikasi ulasan pengguna yang akan divisualisasikan menggunakan *wordcloud* dan diagram (Agustiranti et al., 2024). *Wordcloud* yaitu representasi visual yang menampilkan kata-kata paling sering muncul dalam suatu dokumen atau teks, dimana ukuran kata menunjukkan frekuensi kemunculannya. Semakin besar ukuran kata dalam *wordcloud*, semakin tinggi frekuensi kata kemunculannya, dan sebaliknya jika semakin kecil ukurannya, semakin rendah juga frekuensinya (Santoso & Wibowo, 2022). Pada penelitian ini, hasil klasifikasi nantinya akan divisualisasikan dalam bentuk *wordcloud* dan diagram yang disajikan melalui tampilan sederhana berbasis *web* (Utami et al., 2024).

B. Kajian Penelitian Yang Relevan

Dalam bagian ini, akan dibahas beberapa penelitian sebelumnya yang relevan, yang bertujuan untuk memperkuat kajian teori serta mendukung pelaksanaan penelitian ini, antara lain:

1. Penelitian oleh (Nugraha & Gustian, 2023) membandingkan sentimen penggunaan aplikasi transportasi online (Gojek, Grab, Maxim). Penelitian tersebut dilakukan menggunakan algoritma *Naïve Bayes Classifier* berdasarkan komentar para pengguna aplikasi di google play store. Data review yang diambil pada masing-masing aplikasi tersebut berjumlah 3000 data ulasan yang akan diklasifikasikan menjadi sikap positif dan negatif menggunakan Excel. Hasil akhir yang diperoleh pada algoritma NBC menunjukkan aplikasi terbaik dari ketiga aplikasi tersebut di posisi pertama yaitu Maxim dengan akurasi sebesar 93% disusul oleh Grab dengan tingkat akurasi sebesar 87% dan Gojek diposisi terakhir dengan akurasi 86%.

2. Penelitian oleh (Agustiranti et al., 2024) menerapkan *Naïve Bayes* untuk menganalisis tanggapan pengguna terhadap transportasi umum kereta cepat Jakarta-Bandung (Whoosh) di media sosial twitter atau X. Penelitian ini dilakukan untuk mengeksplorasi penerapan *Naïve Bayes* dalam sentimen pengguna kereta cepat Jakarta-Bandung dan memahami pandangan masyarakat terhadap

layanan tersebut dengan mengelompokkan 714 komentar menjadi sentimen positif, negatif, dan netral menggunakan pembobotan fitur *TF-IDF*. Pemrosesan pelabelan data dilakukan dengan teknik manual, dan setelah melalui pemrosesan dengan teknik *vadersentimen* menghasilkan 64 sentimen negatif, 63 sentimen positif, dan 572 sentimen netral. Metode klasifikasi *Naïve Bayes* menunjukkan performa yang baik, dengan akurasi mencapai 88%, presisi 82%, dan *recall* 88%. Maka, dapat disimpulkan bahwa secara umum tanggapan masyarakat pada layanan transportasi ini dinilai positif.

3. Penelitian oleh (Fachreza & Handoko, 2024)
   terkait transportasi umum Bus Rapid Transit yaitu menganalisis tanggapan masyarakat terhadap kinerja Bus Trans Semarang dengan pengklasifikasian metode *Random Forest*. Data ulasan dikumpulkan melalui media sosial X dengan kata kunci "Trans Semarang" dan dikelompokkan ke dalam kategori positif, negatif, dan netral. Dari 136 data uji, hasil menunjukkan bahwa 47,43% ulasan positif, 32,28% ulasan netral, dan 15,29% ulasan negatif. Algoritma *Random Forest* memiliki

akurasi 81%, presisi 80%, *recall* 80%, dan *f-measure* 80%, menunjukkan performa yang baik dalam pengklasifikasian sentimen.

4. (Fatmawati et al., 2024) melalukan penelitian pada aplikasi transportasi biro perjalanan online yaitu RedBus dengan metode *Naïve Bayes*. Data ulasan yang diambil 600 ulasan diperoleh dari Google Play Store kemudian diklasifikasikan ke dalam sentimen positif serta negatif. Penelitian ini membandingkan pembobotan kata dengan metode *TF-RF* dan *TF-IDF*, kemudian diujikan dengan *confusion matrix* serta *k-fold cross validation*. Hasil menunjukkan akurasi terbaik pada rasio 90% data latih dan 10% data uji, dengan 93,8% menggunakan pembobotan *TF-RF* dan 94,6% dengan pembobotan *TF-IDF*. Akurasi klasifikasi dengan pembobotan *TF-IDF* menujukkan performa lebih unggul dengan akurasi 93,56%, presisi 93,97%, *spectificty* 94,68%, dan *f-measure* 93,53%, hal ini menunjukkan bahwa metode pembobotan terbaik pada penelitian ini adalah *TF-IDF*.

5. (Verawati & Jaelani, 2024) melakukan penelitian menggunakan Multinomial Naïve Bayes terhadap

sentimen masyarakat pada transportasi bus listrik diambil dari platform media sosial X, dengan pelabelan sentimen positif, negatif, dan netral berbasis *lexicon base*. Dari 2412 tweet, sentimen positif mendominasi sebesar 77,32% sementara sentimen negatif 22,69%. Dilakukan 4 kali percobaan rasio perbandingan yaitu 9:1, 8:2, 7:3, dan 6:4. Akurasi Model bervariasi tergantung rasio data latih dan uji, dengan peningkatan setelah *hyperparameter tuning*. Tanpa optimalisasi, akurasi tertinggi adalah 75,6% pada rasio 9:1 dan 8:2, tetapi setelah *tuning* dengan alpha 0.2, akurasi mencapai 78% pada rasio 6:4. Optimalisasi pada model juga membawa risiko *overfitting* dan rasio pembagian data dapat mempengaruhi tingkat akurasi yang diperoleh.

Perbandingan antara penelitian-penelitian yang relevan dengan penelitian yang sedang dilakukan ini dapat disajikan dalam tabel kajian penelitian relevan, yang ditampilkan pada Tabel 2.2 berikut.

Tabel 2. 2 *Kajian Penelitian Relevan*

No	Peneliti	Topik	Metode	Klasifikasi	Hasil
1	(Nugraha & Gustian, 2023)	Sentimen pengguna terhadap penggunaan aplikasi transportasi online (Gojek, Grab, Maxim) (Google Play Store)	Naïve Bayes Classifier	Positif dan negatif	Penelitian dengan menggunakan 3000 data ulasan pada masing-masing aplikasi menunjukkan bahwa model naïve bayes mampu memberikan hasil yang baik. Nilai akurasi tertinggi diperoleh pada aplikasi Maxim sebesar 93%, diikuti oleh Grab sebesar 87%, dan Gojek sebesar 86%.
2	(Agustiranti et al., 2024)	Tanggapan pengguna terhadap transportasi umum kereta cepat Jakarta-Bandung (Whoosh) (Twitter 'X')	Naïve Bayes	Positif, negatif, dan netral	Penelitian yang menggunakan 714 data komentar dengan pelabelan secara manual serta pembagian data raiso 80:20 menunjukkan bahwa model naïve bayes menghasilkan akurasi yang cukup tinggi sebesar 88%, presisi 82%, dan recall 88%.
3	(Fachreza & Handoko, 2024)	Tanggapan masyarakat terhadap transportasi umum Bus Rapid Transit	Random Forest	Positif, negatif, dan netral	Penelitian yang menggunakan 955 data post dari pengguna terhadap layanan kinerja Trans Semarang

		terhadap kinerja Trans Semarang (Twitter 'X')			dilakukan dengan pelabelan manual serta pembagian data dengan rasio 90:10. Hasil klasifikasi pada algoritma ini menunjukkan akurasi sebesar 81%, presisi 81%, recall 80%. Dan f-*measure* 80%.
4	(Fatmawati et al., 2024)	Sentimen pengguna terhadap transportasi biro perjalanan online bus antar provinsi (RedBus) (Google Play Store)	Naïve Bayes	Positif dan Negatif	Penelitian yang menggunakan 600 data ulasan membandingkan dua metode pembobotan kata, yaitu TF-RF dan TF-IDF dengan pembagian data rasio 90:10. Hasil menunjukkan bahwa klasifikasi dengan pembobotan TF-IDF memberikan performa lebih baik dengan akurasi 93,56%, presisi 93,97%, *specificity* 94,68%, dan f-*measure* 93,53%.
5	(Verawati & Jaelani, 2024)	Tanggapan masyarakat terhadap transportasi bus listrik (Twitter 'X')	Naïve Bayes	Positif dan Negatif	Penelitian yang menggunakan 2585 tweet dilakukan pelabelan otomatis dengan metode lexicon-based dan menggunakan empat rasio

					pembagian data: 9:1, 8:2, 7:3, dan 6:4. Hasil tertinggi diperoleh pada rasio 8:2 dengan akurasi sebesar 75,6%, presisi 71,8%, recall 92,6%, dan f1-score 80,9%. Pada rasio 9:1 akurasi juga sebesar 75,6%, rasio 7:3 sebesar 74,6%, dan rasio 6:4 sebesar 74,4%.
6	Yang sedang dilakukan		Naïve Bayes Classifier	Positif dan Negatif	

Berdasarkan penelitian yang telah dilakukan oleh peneliti sebelumnya, penelitian tersebut dapat dijadikan sebagai referensi untuk penelitian ini. Penulis tertarik untuk mengkaji klasifikasi sentimen pengguna terhadap transportasi umum Bus Rapid Transit (Trans Semarang) melalui ulasan pada google play store menggunakan Naïve Bayes Classifier yang bertujuan mengelompokkan sentimen pengguna ke dalam dua kategori sentimen kategori positif dan negatif guna memahami persepsi mereka terhadap layanan aplikasi tersebut.

Perbedaan utama dengan penelitian yang telah dilakukan sebelumnya yaitu lebih berfokus mengklasifikasi pada aplikasi transportasi Bus Rapid Transit (BRT) Semarang,

sementara penelitian terkait aplikasi Trans Semarang dengan metode yang dilakukan dan objek di Google Play belum pernah dilakukan oleh peneliti-peneliti sebelumnya. Sementara itu, pengklasifikasian terkait layanan Bus Rapid Transit masih jarang diteliti oleh para penelitian sebelumnya.

BAB III
METODOLOGI PENELITIAN

A. Metode Pengumpulan Data

1. Studi Pustaka

   Penulis melakukan studi pustaka untuk mempelajari konsep, analisis, dan masalah yang relevan dengan penelitian ini. Penulis melakukan ini dengan memanfaatkan beragam sumber, termasuk buku, jurnal, dan skripsi. Selain itu, penulis juga mencari data yang dilakukan secara online melalui situs web pada internet yang terkait dengan klasifikasi sentimen, machine learning, text mining, dan algoritma *Naïve Bayes* classifier.

2. Studi Lapangan

   Penulis melaksanakan observasi secara langsung terhadap aktivitas serta pandangan pengguna aplikasi Trans Semarang dengan mengumpulkan ulasan-ulasan yang terdapat di Google Play. Melalui pengamatan ini, penulis berusaha memahami lebih dalam bagaimana persepsi serta tanggapan pengguna terhadap fitur layanan aplikasi tersebut. Dengan melihat ulasan tersebut, penulis berharap dapat memperoleh gambaran yang lebih rinci tentang kepuasan pengguna berdasarkan masukan langsung dari pengguna aplikasi.

Penulis juga melakukan pencarian dan pengumpulan data secara online melalui situs web Google Play pada aplikasi Trans Semarang, menggunakan teknik web *scrapping* yang diakses melalui *extensions* Google Chrome. Dalam pengambilan data ini, penulis mengambil data ulasan dengan kata pencarian menggunakan ID aplikasi yaitu "ngi.brtsemarang.apppublic" pada alamat situs aplikasi Trans Semarang.

B. Dataset Penelitian

Jumlah data ulasan pengguna aplikasi Trans Semarang yang ada di Google Play Store tercatat sebanyak 1.109 data ulasan. Namun, dari jumlah tersebut, hanya 501 ulasan saja yang berupa teks yang dapat diambil sebagai dataset. Sisanya tidak dapat diambil karena berupa ulasan tanpa teks yang hanya berisi ulasan berupa rating score bintang. Data yang diambil mencakup ulasan teks pengguna aplikasi Trans Semarang dari rentang waktu 25 April hingga 17 Juni 2025 yang berjumlah 500 data. Hal ini disebabkan oleh keterbatasan jumlah ulasan berbasis teks, karena kebanyakan hanya berupa pemberian rating tanpa disertai komentar.

C. Alat dan perangkat Penelitian

Untuk mengimplementasikan sistem, beberapa perangkat keras (*hardware*) dan perangkat lunak (*software*) harus dipenuhi agar prosesnya dapat berjalan dengan optimal. Berikut adalah kebutuhan yang digunakan dalam penelitian ini:

1. Perangkat Keras (hardware) yang dibutuhkan:

Tabel 3. 1 *Spesifikasi Perangkat Keras*

No	Hardware	Spesifikasi
1.	Perangkat	HP Laptop 14s-dk0024AD
2.	Processor	AMD A9-9425 Radeon R5, 5 COMPUTE CORES 2C+3G 3.10 GHz
3.	Memori (RAM)	4,00 GB
4.	Monitor	14 inch

2. Perangkat Lunak (software) yang dibutuhkan:

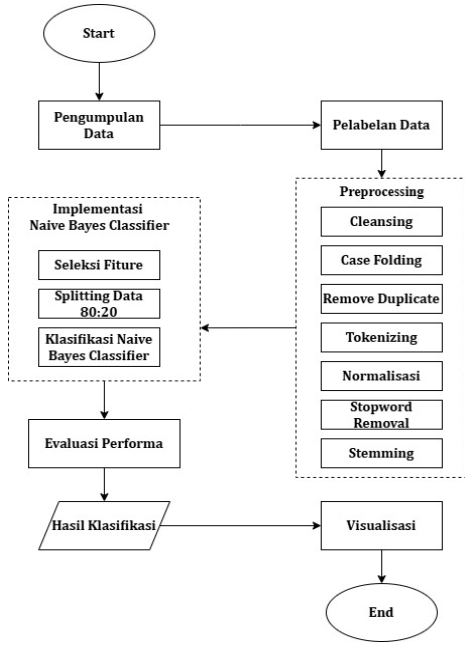
Tabel 3. 2 *Spesifikasi Perangkat Lunak*

No	Software	Spesifikasi
1.	Sistem Operasi	Windows 11 64-bit
2.	Bahasa Pemrograman	*Python* dan *Streamlit*
3.	Browser	Microsoft Edge
4.	Google Drive	Google Colab
5.	Ms.Office	Ms.Word, Ms Excel 2010

D. Tahapan Alur Penelitian

Pada tahap bagian alur penelitian akan dijelaskan gambaran umum serta penjabaran lebih mendalam

mengenai tahapan yang akan dilakukan penulis untuk memberikan pemahaman menyeluruh dari proses tahapan awal hingga akhir agar mencapai hasil yang efektif dan optimal. Berikut ditunjukkan *flowchart* alur penelitian yang dapat dilihat pada gambar 3.1 dibawah ini.



Gambar 3. 1 *Flowchart Tahapan Alur Penelitian*

Berdasarkan *flowchart* alur penelitian diatas, berikut akan dijelaskan penjabaran langkah-langkah dari pengerjaan alur penelitian tersebut diantaranya:

1. Pengumpulan Data

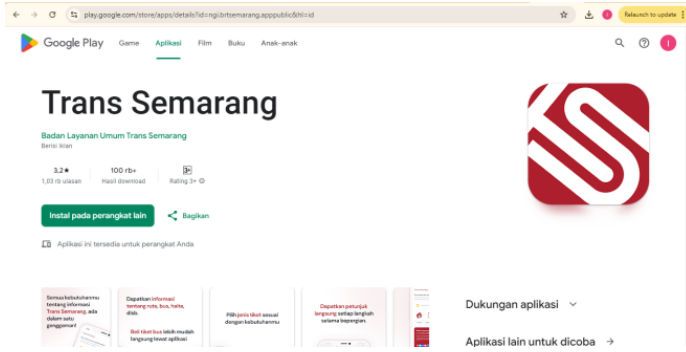
   Pengumpulan data dalam penelitian dilakukan

melalui teknik web *scrapping* dengan bahasa python menggunakan beberapa pustaka library di Google Colab. Data yang akan diambil yaitu data ulasan review para pengguna aplikasi trans semarang dalam bentuk score nilai dan komentar teks, dengan filter diambil ulasan yang berbahasa indonesia. Data ini diambil secara online dari situs Google Play pada aplikasi trans semarang menggunakan ID aplikasi melalui alamat link berikut:
https://play.google.com/store/apps/details?id=ngi.brtsemarang.apppublic&hl=id

Proses pengambilan data menggunakan pustaka library "google-play-scrapper" untuk mendapatkan ulasan secara langsung dari aplikasi, serta pustaka library "pandas" untuk mengelola dan menyusun data dalam bentuk tabel, sementara "numpy" digunakan untuk operasi perhitungan data yang lebih efisien. Selain itu, data yang akan dipakai dalam penelitian mencakup kolom username, score, tanggal (kolom *at*), dan konten ulasan (kolom *content*).

Hasil pengambilan data diperoleh sebanyak 501 data ulasan yang telah dikumpulkan dari hasil scrapping pada google play. Kemudian data difilter mulai rentang waktu mulai 25 April sampai 17 Juni 2025 yang berjumlah 500 ulasan. Jumlah dataset kecil

karena sebagian besar para pengguna hanya memberikan review berupa rating tanpa komentar teks. Hasil proses web *scrapping* tersebut disimpan dalam file berformat CSV yang dapat diunduh untuk keperluan klasifikasi lebih lanjut. Dibawah ini merupakan tampilan dari halaman situs web google play dan hasil *web scrapping* dapat diperhatikan pada gambar berikut.



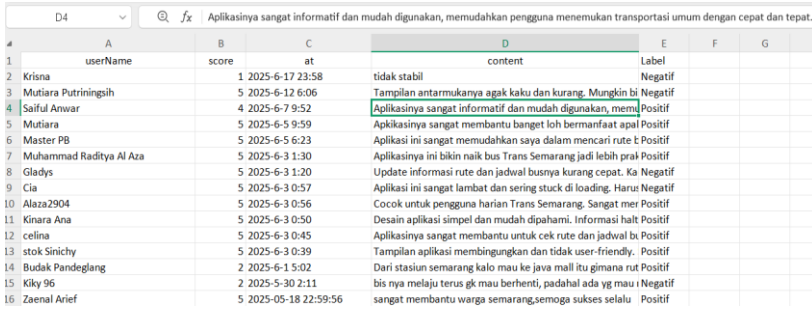
Gambar 3. 2 *Tampilan Situs Aplikasi Trans Semarang*

	userName	score	at	content
2	Krisna	1	2025-6-17 23:58	tidak stabil
3	Mutiara Putriningsih	5	2025-6-12 6:06	Tampilan antarmukanya agak kaku dan kurang. Mungkin bisa dibuat lebih user-friendly agar lebih menarik digunakan.
4	Saiful Anwar	4	2025-6-7 9:52	Aplikasinya sangat informatif dan mudah digunakan, memudahkan pengguna menemukan transportasi umum dengan cepat dan tepat.
5	Mutiara	5	2025-6-5 9:59	Apkikasinya sangat membantu banget loh bermanfaat apalagi buat anak sekolah yg biasa naik transportasi gini
6	Master PB	5	2025-6-5 6:23	Aplikasi ini sangat memudahkan saya dalam mencari rute bus. Informasinya lengkap dan jelas
7	Muhammad Raditya Al Aza	5	2025-6-3 1:30	Aplikasinya ini bikin naik bus Trans Semarang jadi lebih praktis. Tidak perlu bingung lagi harus nunggu dimana
8	Gladys	5	2025-6-3 1:20	Update informasi rute dan jadwal busnya kurang cepat. Kadang sudah ada perubahan di lapangan tapi belum tercermin di aplikasi
9	Cia	5	2025-6-3 0:57	Aplikasi ini sangat lambat dan sering stuck di loading. Harusnya bisa jadi solusi, malah jadi bikin repot
10	Alaza2904	5	2025-6-3 0:56	Cocok untuk pengguna harian Trans Semarang. Sangat membantu sekali, Lanjutkan inovasinya!
11	Kinara Ana	5	2025-6-3 0:50	Desain aplikasi simpel dan mudah dipahami. Informasi halte dan estimasi waktu kedatangan cukup akurat. Semoga terus dikembangkan
12	celina	5	2025-6-3 0:45	Aplikasinya sangat membantu untuk cek rute dan jadwal bus. Sekarang nggak bingung lagi mau naik koridor berapa. Terima kasih Trans Semarang!
13	stok Sinichy	5	2025-6-3 0:39	Tampilan aplikasi membingungkan dan tidak user-friendly. Susah cari halte terdekat dan peta sering tidak muncul
14	Budak Pandeglang	2	2025-6-1 5:02	Dari stasiun semarang kalo mau ke java mall itu gimana rutenya?

Gambar 3. 3 *Hasil Scraping Web*

2. Pelabelan Data

Setelah didapatkan hasil berbentuk file csv dari hasil *scrapping* data, dilanjutkan dengan melakukan pelabelan atau memberi label yang dikelompokkan menjadi dua sentimen pada data tersebut secara manual. Hasil dari *scrapping web* yang sudah diberi label ditunjukkan pada gambar 3.4.



D4 | Aplikasinya sangat informatif dan mudah digunakan, memudahkan pengguna menemukan transportasi umum dengan cepat dan tepat.

	A	B	C	D	E	F	G
1	userName	score	at	content	Label		
2	Krisna	1	2025-6-17 23:58	tidak stabil	Negatif		
3	Mutiara Putriningsih	5	2025-6-12 6:06	Tampilan antarmukanya agak kaku dan kurang. Mungkin bi	Negatif		
4	Saiful Anwar	4	2025-6-7 9:52	Aplikasinya sangat informatif dan mudah digunakan, memu	Positif		
5	Mutiara	5	2025-6-5 9:59	Apkikasinya sangat membantu banget loh bermanfaat apal	Positif		
6	Master PB	5	2025-6-5 6:23	Aplikasi ini sangat memudahkan saya dalam mencari rute k	Positif		
7	Muhammad Raditya Al Aza	5	2025-6-3 1:30	Aplikasinya ini bikin naik bus Trans Semarang jadi lebih prak	Positif		
8	Gladys	5	2025-6-3 1:20	Update informasi rute dan jadwal busnya kurang cepat. Ka	Negatif		
9	Cia	5	2025-6-3 0:57	Aplikasi ini sangat lambat dan sering stuck di loading. Harus	Negatif		
10	Alaza2904	5	2025-6-3 0:56	Cocok untuk pengguna harian Trans Semarang. Sangat mer	Positif		
11	Kinara Ana	5	2025-6-3 0:50	Desain aplikasi simpel dan mudah dipahami. Informasi halt	Positif		
12	celina	5	2025-6-3 0:45	Aplikasinya sangat membantu untuk cek rute dan jadwal bu	Positif		
13	stok Sinichy	5	2025-6-3 0:39	Tampilan aplikasi membingungkan dan tidak user-friendly.	Positif		
14	Budak Pandeglang	2	2025-6-1 5:02	Dari stasiun semarang kalo mau ke java mall itu gimana rut	Positif		
15	Kiky 96	2	2025-5-30 2:11	bis nya melaju terus gk mau berhenti, padahal ada yg mau n	Negatif		
16	Zaenal Arief	5	2025-05-18 22:59:56	sangat membantu warga semarang,semoga sukses selalu	Positif		

Gambar 3. 4 *Hasil Pelabelan*

Proses pelabelan data dalam penelitian ini akan diberikan label positif dan negatif. Proses pelabelan data dilakukan dengan memahami ulasan pengguna berdasarkan emosi dan perasaan yang diungkapkan dalam kata-kata yang digunakan. Berikut contoh dari labelling data ditunjukkan pada tabel 3.3.

Tabel 3. 3 *Contoh Labelling Data*

Ulasan	Kategori Label
(Intan) Jelek banget, bis ga terdeteksi, lemot, ga update	Negatif
(Rendy Romadhon) Aplikasi nya bagus kalian harus download	Positif

3. Preprocessing Data

Tahap *Preprocessing* dilakukan dimana data dibersihkan dan diperbaiki, kemudian dilakukan cleansing. *Preprocessing* data dilakukan untuk mengolah pembersihan dataset mentah agar siap dilakukan klasifikasi. Ketika data tersebut dikumpulkan, biasanya data tersebut bersifat tidak tersruktur dan mengandung berbagai karakter. Tujuan *preprocessing* ini untuk menghilangkan noise. Selanjutnya pengolahan data harus melalui beberapa tahapan *preprocessing* diantaranya:

a. Cleansing

Pada tahap ini dokumen akan dibersihkan dari kata-kata yang dinilai tidak penting dan simbol tanda baca seperti titik (.), koma (,), garis miring (/), angka, dan emoticon (~!@%$^&*:";,<>{} [] -, _=+'). Proses ini dilakukan untuk menghilangkan noise dan data yang tidak penting. Berikut contoh dari ulasan hasil *cleansing* data ditunjukkan pada tabel 3.4.

Tabel 3. 4 *Contoh Tahap Cleansing Data*

Text Input	Text Output
(Intan) Jelek banget, bis ga terdeteksi, lemot, ga update	Jelek banget bis ga terdeteksi lemot ga update

b. Case Folding

Case folding digunakan mengonversi semua huruf kapital dalam korpus menjadi huruf kecil (*lowercase*). Hanya huruf dari a hingga z saja yang diterima, sedangkan karakter lainnya dianggap sebagai pemisah. Berikut contoh dari tahapan hasil *case folding* ditunjukkan pada tabel 3.5.

Tabel 3. 5 *Contoh Tahap Case Folding*

Text Input	Text Output
(Intan) Jelek banget, bis ga terdeteksi, lemot, ga update	jelek banget bis ga terdeteksi lemot ga update

c. Remove Duplicate

Pengguna aplikasi di google play store terkadang memberikan ulasan yang sama secara berulang. Oleh karena itu, proses *remove duplicate* perlu digunakan dengan tujuan untuk menghapus ulasan yang memiliki kesamaan sehingga hanya satu ulasan yang dipertahankan untuk mempermudah

dan menyederhanakan tahap *preprocessing*. Pada tahap ini, dalam dataset aplikasi Trans Semarang, tidak adanya ulasan yang memiliki duplicate atau sama dengan ulasan yang lain. Sehingga, pada proses ini jumlah ulasan data masih tetap. Berikut contoh dari tahapan hasil *remove duplicate* ditunjukkan pada gambar 3.5.

	userName	score	at	content	Label	text_clean
0	Krisna	1	2025-06-17 23:58:24	tidak stabil	Negatif	tidak stabil
1	Mutiara Putriningsih	5	2025-06-12 06:06:54	Tampilan antarmukanya agak kaku dan kurang. Mu...	Negatif	tampilan antarmukanya agak kaku dan kurang mun...
2	Saiful Anwar	4	2025-06-07 09:52:51	Aplikasinya sangat informatif dan mudah diguna...	Positif	aplikasinya sangat informatif dan mudah diguna...
3	Mutiara	5	2025-06-05 09:59:06	Apkikasinya sangat membantu banget loh bermanf...	Positif	apkikasinya sangat membantu banget loh bermanf...
4	Master PB	5	2025-06-05 06:23:32	Aplikasi ini sangat memudahkan saya dalam menc...	Positif	aplikasi ini sangat memudahkan saya dalam menc...
...	...	...	...	...	...	...
495	Pengguna Google	5	2019-05-01 13:36:48	Sangat membantu mengetahui bus terdekat. Jadi ...	Positif	sangat membantu mengetahui bus terdekat jadi b...
496	Pengguna Google	4	2019-05-01 05:16:47	Karena saya tidak tinggal di Semarang saya bel...	Positif	karena saya tidak tinggal di semarang saya bel...
497	Pengguna Google	5	2019-04-26 03:50:29	amazing apk	Positif	amazing apk
498	Pengguna Google	5	2019-04-25 08:05:24	mantapppp	Positif	mantapppp
499	Pengguna Google	5	2019-04-25 07:48:10	mantab banget nih aplikasi	Positif	mantab banget nih aplikasi

500 rows × 6 columns

Gambar 3. 5 *Hasil Tahap Remove Duplicate*

d. Tokenizing

Proses *tokenizing* diperlukan untuk memisahkan kata-kata dalam suatu kalimat untuk melanjutkan proses analisis. Tahapan memisahkan teks menjadi potongan-potongan kata yang tidak independen disebut juga token (Syam et al., 2023). Token dapat berupa kata, angka, symbol, atau tanda baca yang ada dalam teks. Berikut contoh dari tahapan proses *tokenizing* ditunjukkan pada tabel 3.6.

Tabel 3. 6 *Contoh Tahap Proses Tokenizing*

Text Input	Text Output
jelek banget bis ga terdeteksi lemot ga update	['jelek', 'banget', 'bis', 'ga', 'terdeteksi', 'lemot', 'ga', 'update']

e. Normalisasi Kata

Ulasan yang diungkapkan para pengguna aplikasi di google play seringkali mengandung kata-kata informal dan singkatan serta bahasa gaul yang sudah popular dimasa sekarang. Normalisasi kata pada ulasan tersebut bertujuan untuk mengonversi kata-kata yang dinilai tidak baku atau singkatan ke dalam bentuk standar yang sesuai dengan KBBI. Proses ini membantu menyamakan kata-kata tersebut dengan standar bahasa yang baku sehingga analisis teks menjadi lebih akurat dan terstruktur. Berikut contoh dari tahapan normalisasi kata ditunjukkan pada tabel 3.7.

Tabel 3. 7 *Contoh Tahap Normalisasi Kata*

Text Input	Text Output
['jelek', 'banget', 'bis', 'ga', 'terdeteksi', 'lemot', 'ga', 'update']	['jelek', 'banget', 'bis', 'tidak', 'terdeteksi', 'lemot', 'tidak', 'update']

f. Stopword Removal

Tahap proses *stopword removal* adalah proses yang bertujuan untuk mengambil kata-kata penting dan yang tidak relevan dari kalimat tanpa mengubah maknanya. Proses ini dilakukan secara otomatis dengan menggunakan library "nltk.corpus stopwords" dimana kamus *stopword* yang digunakan adalah khusus untuk bahasa Indonesia. Berikut contoh dari tahapan *stopword removal* ditunjukkan pada tabel 3.8.

Tabel 3. 8 *Contoh Tahap Stopword Removal*

Text Input	Text Output
['jelek', 'banget', 'bis', 'tidak', 'terdeteksi', 'lemot', 'tidak', 'update']	['jelek', 'banget', 'bis', 'terdeteksi', 'lemot', 'update']

g. Stemming

Tahap *stemming* dilakukan untuk mengubah kata berimbuhan menjadi bentuk kata dasar atau mengembalikannya ke kata akar. Proses ini dilakukan dengan menghilangkan imbuhan dari kata tersebut. *Stemming* dilakukan secara otomatis menggunakan library Sastrawi untuk memproses setiap kata dalam teks menjadi

bentuk kata dasarnya. Berikut contoh dari tahapan *stemming* ditunjukkan pada tabel 3.9.

Tabel 3. 9 *Contoh Tahap Proses Stemming*

Text Input	Text Output
['jelek', 'banget', 'bis', 'terdeteksi', 'lemot', 'update']	jelek banget bis deteksi lemot update

4. Seleksi Fitur Kata

Data yang telah bersih dari hasil preprocessing harus diubah menjadi bentuk numerik, sehingga dilakukan proses ekstraksi atau seleksi fitur. Dalam penelitian yang dilakukan, penulis menggunakan metode TFIDF (*Term Frequency Invers Document Frequency*) untuk menentukan pembobotan fitur pada setiap kata. TFIDF digunakan untuk menghitung bobot kata berdasarkan frekuensi kemunculannya dalam suatu dokumen. TFIDF mengonversi kata menjadi representasi numerik berdasarkan frekuensi keumunculan dokumen dan seberapa jarang kata tersebut muncul dalam dataset, sehingga model dapat lebih akurat mengidentifikasi pola sentimen. Pembobotan kata dilakukan secara otomatis dengan menggunakan bahasa pemrograman python melalui library TfidfVectorizer dari sklearn (Verawati & Jaelani, 2024). Pembobotan juga bisa dilakukan secara

manual menggunakan perhitungan sesuai dengan rumus yang dapat dilihat pada bab sebelumnya. Selain itu, metode TFIDF juga dapat meningkatkan akurasi proses klasifikasi.

5. Pembuatan Data Training dan Testing Model

Data training dan testing dibentuk untuk membagi data dari hasil *preprocessing* ke dalam perbandingan tertentu. Pembagian ini dilakukan menggunakan metode *splitting* data pada rasio 80:20, dimana 80% data digunakan sebagai data latih untuk melatih model, dan 20% sisanya sebagai data uji untuk menguji performa model. Data training dilakukan untuk membangun model klasifikasi, sementara data testing untuk mengukur performa model yang dihasilkan (Hakim, 2021). Selanjutnya, data training dan testing diimplementasikan dalam proses klasifikasi menggunakan algoritma naïve bayes.

6. Klasifikasi Naïve Bayes Classifier

Setelah melalui proses training dan testing, pemodelan dilakukan dengan proses klasifikasi algoritma naïve bayes classifier untuk mengukur performa algoritma tersebut. Naïve bayes mampu menghasilkan akurasi yang baik meskipun hanya

dengan sedikit data training. Pada penelitian ini, sentimen akan diklasifikasikan ke dalam dua kategori yaitu positif dan negatif. Penerapan klasifikasi naïve bayes classifier biasanya dilakukan setelah melalui tahap *preprocessing* dan pembobotan kata TFIDF.

7. Evaluasi Performa Model

Evaluasi merupakan tahap yang bertujuan untuk mengetahui tingkat kualitas performa algoritma klasifikasi yang diterapkan dalam penelitian ini. Proses evaluasi model dilakukan menggunakan confusion matrix, yaitu sebuah table yang digunakan untuk mengevaluasi kinerja dari model klasifikasi atau metode yang dipakai untuk membandingkan hasil prediksi sistem dengan data aktual, dengan menggunakan dua kelas sentimen. Confusion matrix membantu evaluasi menyeluruh dari performa model (Fatmawati et al., 2024). Melalui evaluasi ini, diperoleh matrik seperti akurasi, presisi, *recall*, dan *F1-score* yang menggambarkan seberapa efektivitas model dalam melakukan klasifikasi data dengan benar.

8. Visualisasi

Visualisasi tahap terakhir yang dilakukan bertujuan untuk mengidentifikasi informasi penting dalam

keseluruhan teks ulasan berupa topik yang paling sering diulas oleh para pengguna aplikasi trans semarang. Pada tahap ini, hasil klasifikasi akan divisualisasikan dalam bentuk *wordcloud* serta diagram untuk memudahkan pengguna memahamai pola dan kata-kata pada topik yang sering diulas. Hasil visualisasi ini juga akan ditampilkan melalui tampilan sederhana berbasis *web*, untuk menampilkan hasil visualisasi data dari sentimen pengguna secara jelas sehingga lebih mudah dipahami oleh para pengguna nya.

BAB IV
HASIL DAN PEMBAHASAN

Dalam bab ini menyajikan hasil dan pembahasan terkait proses klasifikasi yang telah dirancang. Penelitian ini menggunakan data ulasan pengguna aplikasi Trans Semarang yang dikumpulkan dari platform Google Play Store sebagai sumber data utama. Untuk mengevaluasi kinerja sistem, dilakukan serangkaian proses pengujian yang mencakup pengambilan data ulasan (*scrapping*), pengolahan data (*preprocessing*), klasifikasi metode Naive Bayes, perhitungan tingkat akurasi, serta visualisasi hasil klasifikasi data.

A. Pengumpulan Data Ulasan

Pengumpulan data pada penelitian ini dilakukan dengan cara teknik *scrapping* menggunakan bahasa pemrograman *python* dengan bantuan Google *Colaboratory.* Berikut beberapa tahap dalam proses scrapping:

1. Mengunduh library Google play Scrapper

Tahap awal yang dilakukan adalah mengunduh dan menginstall sebuah paket library *google play scrapper* untuk menarik ulasan pengguna aplikasi. Code dapat dilihat pada tabel 4.1 berikut ini.

Tabel 4. 1 *Code Install Library Google Play Scrapper*

```
!pip install google-play-scraper
```

2. Mengimport Library Yang Digunakan

Dalam proses *scrapping* harus mengimport beberapa modul library yang nantinya akan digunakan. Pada penelitian ini akan mengimport modul import pandas as pd dan import numpy as np. Library pandas yaitu pustaka dalam python yang digunakan untuk analisis data termasuk manipulasi, persiapan, dan pembersihan data. Sedangkan library Numpy yaitu pustaka yang juga disediakan oleh *pyhton* lebih berfokus pada komputasi dan numerik. Import library ditunjukkan pada code tabel 4.2 berikut.

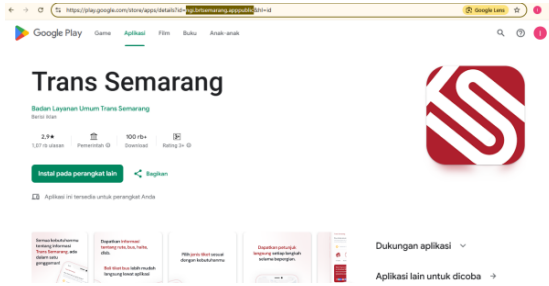
Tabel 4. 2 *Code Import Library Proses Scrapping*

```
from google_play_scraper import app
import pandas as pd
import numpy as np
```

3. Masukkan ID Aplikasi Trans Semarang

Memasukkan dan salin alamat ID pada aplikasi Trans Semarang melalui website Google Play Store untuk dimasukkan ke dalam code pada tahap proses *scrapping*. Gambar ID aplikasi yang

digunakan ditunjukkan pada gambar 4.1 berikut.



Gambar 4. 1 *Tampilan ID Aplikasi Trans Semarang*

4. Proses Scrapping Data Aplikasi Trans Semarang

Setelah didapatkan ID aplikasi Trans Semarang dari Google Playstore pada tahap sebelumnya dilanjutkan dengan memasukkan ID tersebut ke dalam code untuk proses *scrapping* data selanjutnya ditunjukkan tabel 4.3 berikut.

Tabel 4. 3 *Code Proses Scrapping Data*

```
from google_play_scraper import Sort, reviews

result, continuation_token = reviews(
'ngi.brtsemarang.apppublic',
lang='id' ,
sort=Sort.MOST_RELEVANT,
count=600,
filter_score_with=None
)
```

Berdasarkan source code diatas, pustaka library "google_play_scrapper" digunakan untuk melakukan pengambilan data ulasan aplikasi yang

berasal dari Google Playstore dengan menggunakan paket ID **'ngi.brtsemarang.apppublic'**. Parameter lang='id' digunakan untuk menentukan bahwa ulasan yang dikumpulkan hanya dalam bahasa Indonesia, sedangkan parameter sort=Sort.MOST_RELEVANT berfungsi untuk mengurutkan ulasan sesuai relevansi tertinggi, sehingga ulasan yang paling populer akan ditampilkan terlebih dahulu. Selanjutnya, nilai 'count=600' mengindikasikan bahwa jumlah maksimum ulasan yang akan dikumpulkan sebanyak 600 ulasan. Parameter 'filter_score_with=None' berarti semua ulasan dengan berbagai penilaian (skor) akan disertakan tanpa penyaringan. Hasil pemanggilan fungsi 'reviews()' akan disimpan dalam variabel 'result' yang berisi data ulasan yang telah dikumpulkan. Jika jumlah ulasan melebihi batas yang ditentukan, penggunaan 'continuation_token' memungkinkan untuk pengambilan ulasan tambahan guna melanjutkan proses pengumpulan data.

5. Membuat DataFrame Dari Hasil *Scrapping*

Dari proses *scrapping* yang dilakukan, hasil ulasan yang diperoleh masih dalam bentuk *array* yang

kurang terstruktur dan sulit dipahami oleh pengguna awam. Oleh karena itu, diperlukan proses pengubahan data tersebut menjadi DataFrame menggunakan *library* pandas dan numpy, sehingga akan lebih mudah dipahami, dianalisis, dan diolah untuk keperluan lebih lanjut.

Tabel 4. 4 *Code Mengubah Array Menjadi DataFrame*

```
Df_busu=pd.DataFrame(np.array(result),columns=[
'review'])
df_busu=df_busu.join(pd.DataFrame(df_busu.pop('
review').tolist()))
df_busu.head()
```

Hasil ulasan dikonversi menjadi DataFrame dengan memindahkan kolom 'review' ke dalam DataFrame yang baru. Fungsi tolist() digunakan untuk mengubah isi kolom tersebut menjadi daftar, kemudian daftar tersebut dikonversi kembali kedalam bentuk DataFrame menggunakan pd.DataFrame(), sehingga data lebih rapi dan mudah diproses.

	reviewId	userName	userImage	content	score	thumbsUpCount	reviewCreatedVersion	at	replyContent	repliedAt	appVersion
0	64185673-71c6-4ad3-b1e1-2e1841eee392	baskoro wijanarko	https://play-lh.googleusercontent.com/a/ACg8oc...	aplikasinya bagus, tapinya agak rumit bagi ora...	5	15	2.3.3	2025-03-24 02:21:14	None	None	2.3.3
1	8303422b-0fd2-4d70-9ece-5a69b54d1fc5	Saiful Anwar	https://play-lh.googleusercontent.com/a/ACg8oc...	Aplikasinya sangat informatif dan mudah diguna...	4	0	2.3.3	2025-06-07 09:52:51	None	None	2.3.3
2	024e9cb6-5931-44d1-8cb6-5fc35a42e42b	Mutiara Putriningsih	https://play-lh.googleusercontent.com/a/ACg8oc...	Tampilan antarmukanya agak kaku dan kurang. Mu...	5	0	2.3.3	2025-06-12 06:06:54	None	None	2.3.3
3	816595ff-a8dd-4866-86f9-9380cc8fddaf	Khasaras Dara Arinta	https://play-lh.googleusercontent.com/a/ACg8oc...	BRT Semarang jarang sekali berhenti di halte h...	1	13	2.3.3	2025-05-05 01:10:21	None	None	2.3.3
4	c5c9275a-e7b7-4e32-bc4b-d45742230014	Kiky 96	https://play-lh.googleusercontent.com/a-/ALV-U...	bis nya melaju terus gk mau berhenti, padahal ...	2	1	2.3.3	2025-05-30 02:11:58	None	None	2.3.3

Gambar 4. 2 *Hasil DataFrame*

6. Mendapatkan Jumlah Data Ulasan

Jumlah data ulasan yang didapatkan dalam proses scrapping data pada aplikasi Trans Semarang sebanyak 501 data ulasan.

```
[15] len(df_busu.index) #Jumlah data yang didapatkan dari hasil scrapping
```

```
501
```

```
df_busu[['userName', 'score', 'at', 'content']].head() #Memfilter beberapa kolom tertentu
```

	userName	score	at	content
0	baskoro wijanarko	5	2025-03-24 02:21:14	aplikasinya bagus, tapinya agak rumit bagi ora...
1	Saiful Anwar	4	2025-06-07 09:52:51	Aplikasinya sangat informatif dan mudah diguna...
2	Mutiara Putriningsih	5	2025-06-12 06:06:54	Tampilan antarmukanya agak kaku dan kurang. Mu...
3	Khasaras Dara Arinta	1	2025-05-05 01:10:21	BRT Semarang jarang sekali berhenti di halte h...
4	Kiky 96	2	2025-05-30 02:11:58	bis nya melaju terus gk mau berhenti, padahal ...

Gambar 4. 3 *Jumlah Data Ulasan Aplikasi Trans Semarang*

Dilanjutkan melakukan penyaringan atau seleksi terhadap hasil ulasan yang telah diperoleh. Dari DataFrame yang tersedia, sehingga hanya beberapa kolom tertentu yang akan digunakan, yaitu kolom (username, score, at, dan content).

7. **Mengurutkan Data Berdasarkan Tanggal Ulasan**

Setelah data berhasil difilter dari beberapa kolom tertentu, selanjutnya akan diurutkan berdasarkan tanggal ulasan untuk menyusun data lebih terstruktur, code ditunjukkan gambar 4.5 berikut.

Tabel 4. 5 *Code Data Urut Sesuai Tanggal Ulasan*

```
new_df = df_busu[['userName', 'score','at',
'content']]
sorted_df = new_df.sort_values(by='at',
ascending=False)
sorted_df.head()
```

Kemudian kolom data yang telah difilter akan disimpan ke dalam DataFrame baru seperti ditunjukkan gambar berikut 4.4.

	userName	score	at	content
431	riena handayani	1	2025-06-19 00:54:25	ribet
430	Krisna	1	2025-06-17 23:58:24	tidak stabil
2	Mutiara Putriningsih	5	2025-06-12 06:06:54	Tampilan antarmukanya agak kaku dan kurang. Mu...
1	Saiful Anwar	4	2025-06-07 09:52:51	Aplikasinya sangat informatif dan mudah diguna...
478	Mutiara	5	2025-06-05 09:59:06	Apkikasinya sangat membantu banget loh bermanf...

Gambar 4. 4 *Code DataFrame Setelah Proses Filter*

8. **Menyimpan Hasil Proses Scrapping Data Dalam Format CSV**

Selanjutnya proses hasil *scrapping* data akan disimpan dalam bentuk file csv untuk dilakukan pengolahan proses klasifikasi sentimen pada tahap

selanjutnya.

Tabel 4. 6 *Code Simpan Hasil Scrapping Data*

```
my_df.to_csv("ScrappingDataTS.csv", index
= False)
```

Hasil *scrapping* yang telah disimpan dapat diunduh dengan file berformat csv seperti gambar tersebut.

D6 | Apkikasinya sangat membantu banget loh bermanfaat apalagi buat anak sekolah yg biasa naik transportasi gini

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	userName	score	at	content											
2	riena handayani	1	2025-6-19 0:54	ribet											
3	Krisna	1	2025-6-17 23:58	tidak stabil											
4	Mutiara Putriningsih	5	2025-6-12 6:06	Tampilan antarmukanya agak kaku dan kurang. Mungkin bisa dibuat lebih user-friendly agar lebih menarik digunakan.											
5	Saiful Anwar	4	2025-6-7 9:52	Aplikasinya sangat informatif dan mudah digunakan, memudahkan pengguna menemukan transportasi umum dengan cepat dan tepat.											
6	Mutiara	5	2025-6-5 9:59	Apkikasinya sangat membantu banget loh bermanfaat apalagi buat anak sekolah yg biasa naik transportasi gini											
7	Master PB	5	2025-6-5 6:23	Aplikasi ini sangat memudahkan saya dalam mencari rute bus. Informasinya lengkap dan jelas											
8	Muhammad Raditya Al A	5	2025-6-3 1:30	Aplikasinya ini bikin naik bus Trans Semarang jadi lebih praktis. Tidak perlu bingung lagi harus nunggu dimana											
9	Gladys	5	2025-6-3 1:20	Update informasi rute dan jadwal busnya kurang cepat. Kadang sudah ada perubahan di lapangan tapi belum tercermin di aplikasi											
10	Cia	5	2025-6-3 0:57	Aplikasi ini sangat lambat dan sering stuck di loading. Harusnya bisa jadi solusi, malah jadi bikin repot											
11	Alaza2904	5	2025-6-3 0:56	Cocok untuk pengguna harian Trans Semarang. Sangat membantu sekali, Lanjutkan inovasinya!											
12	Kinara Ana	5	2025-6-3 0:50	Desain aplikasi simpel dan mudah dipahami. Informasi halte dan estimasi waktu kedatangan cukup akurat. Semoga terus dikembangkan											
13	celina	5	2025-6-3 0:45	Aplikasinya sangat membantu untuk cek rute dan jadwal bus. Sekarang nggak bingung lagi mau naik koridor berapa. Terima kasih Trans Semarang!											
14	stok Sinichy	5	2025-6-3 0:39	Tampilan aplikasi membingungkan dan tidak user-friendly. Susah cari halte terdekat dan peta sering tidak muncul											
15	Budak Pandeglang	2	2025-6-1 5:02	Dari stasiun semarang kalo mau ke java mall itu gimana rutenya?											
16	Kiky 96	2	2025-5-30 2:11	bis nya melaju terus gk mau berhenti, padahal ada yg mau naik, kondektur nya jg judes, padahal bisnya jg kosong											
17	Zaenal Arief	5	2025-5-18 22:59	sangat membantu warga semarang,semoga sukses selalu											

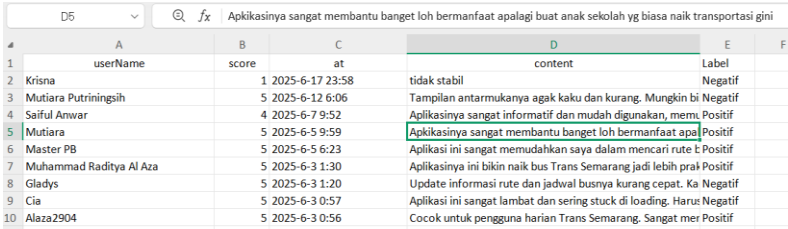
Gambar 4. 5 *Tampilan Hasil Scrapping Data*

B. Pelabelan Data

Hasil file dengan format csv dari proses *scrapping* yang telah berhasil didapatkan, akan difilter berdasarkan tanggal dengan batasan hingga 17 Juni 2025, serta dilanjutkan pada tahap pelabelan data ulasan dengan memberikan label menjadi dua kategori sentimen pada ulasan secara manual. Dalam penelitian ini pada proses pelabelan akan mengelompokkan label menjadi kategori label positif dan negatif. Proses memberikan label dilakukan dengan memahami bagaimana ulasan para pengguna yang diungkapkan

berdasarkan emosi dan perasaan yang diberikan pada aplikasi tersebut.

Dalam proses pelabelan, ulasan dianggap memiliki nilai positif apabila ulasan tersebut mendukung adanya aplikasi tersebut, sedangkan ulasan dengan nilai negatif apabila ulasan tersebut mengungkapkan kritik atau ketidakpuasan pada aplikasi. Hasil pelabelan data ditunjukkan pada gambar dibawah berikut.



D5 — Apkikasinya sangat membantu banget loh bermanfaat apalagi buat anak sekolah yg biasa naik transportasi gini

	A	B	C	D	E	F
1	userName	score	at	content	Label	
2	Krisna	1	2025-6-17 23:58	tidak stabil	Negatif	
3	Mutiara Putriningsih	5	2025-6-12 6:06	Tampilan antarmukanya agak kaku dan kurang. Mungkin bi	Negatif	
4	Saiful Anwar	4	2025-6-7 9:52	Aplikasinya sangat informatif dan mudah digunakan, mem	Positif	
5	Mutiara	5	2025-6-5 9:59	Apkikasinya sangat membantu banget loh bermanfaat apal	Positif	
6	Master PB	5	2025-6-5 6:23	Aplikasi ini sangat memudahkan saya dalam mencari rute b	Positif	
7	Muhammad Raditya Al Aza	5	2025-6-3 1:30	Aplikasinya ini bikin naik bus Trans Semarang jadi lebih prak	Positif	
8	Gladys	5	2025-6-3 1:20	Update informasi rute dan jadwal busnya kurang cepat. Ka	Negatif	
9	Cia	5	2025-6-3 0:57	Aplikasi ini sangat lambat dan sering stuck di loading. Harus	Negatif	
10	Alaza2904	5	2025-6-3 0:56	Cocok untuk pengguna harian Trans Semarang. Sangat mer	Positif	

Gambar 4. 6 *Hasil Labelling Data*

Selanjutnya, dataset yang telah diberikan label secara manual dalam format file .xlsx akan dimuat ke dalam Google Colab untuk proses klasifikasi lebih lanjut. Hasil dataset setelah dilakukan pemfilteran tanggal menjadi sejumlah 500 data ulasan. Tampilan load dataset yang sudah diberi label ditunjukkan pada gambar 4.7 berikut.

```
my_df = pd.read_excel('LabelDuaKelasSentimen.xlsx') #Membaca file data yang sudah diberikan pelabelan secara manual
my_df #Menampilkan data yang sudah diberi label
```

	userName	score	at	content	Label
0	Krisna	1	2025-06-17 23:58:24	tidak stabil	Negatif
1	Mutiara Putriningsih	5	2025-06-12 06:06:54	Tampilan antarmukanya agak kaku dan kurang. Mu...	Negatif
2	Saiful Anwar	4	2025-06-07 09:52:51	Aplikasinya sangat informatif dan mudah diguna...	Positif
3	Mutiara	5	2025-06-05 09:59:06	Apkikasinya sangat membantu banget loh bermanf...	Positif
4	Master PB	5	2025-06-05 06:23:32	Aplikasi ini sangat memudahkan saya dalam menc...	Positif
...	...	...	...	...	...
495	Pengguna Google	5	2019-05-01 13:36:48	Sangat membantu mengetahui bus terdekat. Jadi ...	Positif
496	Pengguna Google	4	2019-05-01 05:16:47	Karena saya tidak tinggal di Semarang saya bel...	Positif
497	Pengguna Google	5	2019-04-26 03:50:29	amazing apk	Positif
498	Pengguna Google	5	2019-04-25 08:05:24	mantapppp	Positif
499	Pengguna Google	5	2019-04-25 07:48:10	mantab banget nih aplikasi	Positif

500 rows × 5 columns

Gambar 4. 7 *Load Dataset Yang Sudah Diberi Label*

Proses pemberian label yang dilakukan secara manual, idealnya dilakukan dengan melibatkan dua orang atau lebih untuk menghindari adanya perbedaan pendapat dalam menentukan sentimen ulasan. Dalam penelitian ini proses pelabelan data dilakukan oleh peneliti bersama satu orang lainnya yang merupakan seorang sarjana lulusan psikologi. Konsekuensi dari keterbatasan ini adalah kemungkinan terdapat beberapa komentar diberi label yang kurang tepat dengan sentimen sebenarnya.

C. Preprocessing Data

Pada tahap *preprocessing* ini terdiri atas beberapa langkah untuk membersihkan dataset mentah sebelum diolah dalam proses klasifikasi. Proses ini sangat

penting agar data lebih terstruktur sehingga hasil analisis menjadi lebih akurat. Beberapa tahap dalam *preprocessing* data meliputi:

a. Cleansing

Dalam proses *cleansing* dilakukan untuk membersihkan data ulasan dari karakter dan kata-kata yang tidak diperlukan. Untuk mendapatkan hasil ulasan yang maksimal, perlunya menghapus beberapa kategori seperti:

1. Penghapusan tanda baca
2. Penghapusan angka
3. Penghapusan emoticon

Dalam melakukan proses cleansing, *library re* dan *emoji* akan digunakan. *Library* re digunakan untuk mengenali pola karakter dalam teks dan memungkinkan pencarian teks berdasarkan pola tertentu. Sedangkan *library* emoji digunakan untuk mengidentifikasi dan menghapus emoji yang terdapat dalam teks. Selanjutnya melakukan instalasi pada *library* emoji yang ditunjukkan pada code gambar 4.7 berikut.

Tabel 4. 7 *Instalasi Library Emoji*

```
pip install emoji
```

Dilanjutkan dengan import *library* re dan *library* emoji ditunjukkan tabel 4.8 berikut.

Tabel 4. 8 *Import Library re dan Emoji*

```
import emoji
import re
```

Program dari proses tahap *cleansing* yang disatukan dengan tahap *case folding* ditunjukkan pada code tabel 4.9 berikut.

Tabel 4. 9 *Code Tahap Proses Cleansing dan CaseFolding*

```
import emoji
import re
def clean_text(df, text_field, new_text_field_name):
    my_df[new_text_field_name] =
my_df[text_field].str.lower()

    my_df[new_text_field_name] =
my_df[new_text_field_name].apply(lambda elem:
re.sub(r"(@[A-Za-z0-9]+)|([^0-9A-Za-z
\t])|(\w+:\/\/\S+)|^rt|http.+?", "", elem))
    my_df[new_text_field_name] =
my_df[new_text_field_name].apply(lambda elem:
re.sub(r"\d+", "", elem))
    return my_df

my_df['text_clean'] = my_df['content'].str.lower()
my_df['text_clean']

my_df = clean_text(my_df, 'content', 'text_clean')
my_df.head(10)
```

Hasil dari data ulasan sebelum dilakukan tahap *cleansing* bisa dilihat pada gambar 4.8 berikut.

```
my_df['content'] #Menampilkan data ulasan sebelum dibersihkan
```

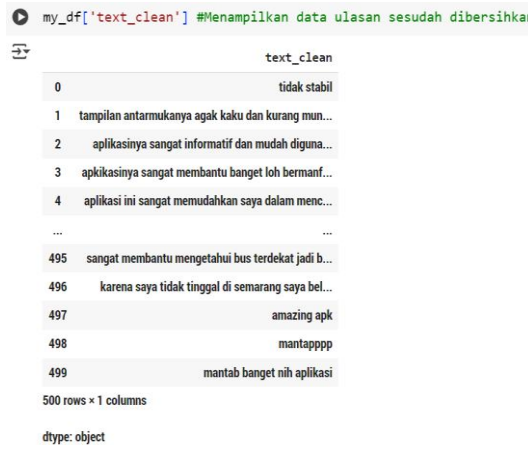
	content
0	tidak stabil
1	Tampilan antarmukanya agak kaku dan kurang. Mu...
2	Aplikasinya sangat informatif dan mudah diguna...
3	Apkikasinya sangat membantu banget loh bermanf...
4	Aplikasi ini sangat memudahkan saya dalam menc...
...	...
495	Sangat membantu mengetahui bus terdekat. Jadi ...
496	Karena saya tidak tinggal di Semarang saya bel...
497	amazing apk
498	mantapppp
499	mantab banget nih aplikasi

500 rows × 1 columns

dtype: object

Gambar 4. 8 *Ulasan Sebelum Proses Cleansing*

Sedangkan hasil dari ulasan yang sudah dilakukan proses *cleansing* akan dihapusnya beberapa simbol-simbol yang tidak diperlukan seperti ditunjukkan pada gambar 4.9 berikut.



Gambar 4. 9 *Ulasan Setelah Proses Cleansing*

Dibawah ini hasil tampilan dari proses *cleansing* dan *case folding* yang telah dilakukan ditunjukkan pada code tabel 4.10 berikut.

Tabel 4. 10 Code *Hasil Proses Cleansing dan Case Folding*

```
my_df['text_clean'] =
my_df['content'].str.lower()
my_df['text_clean']
my_df = clean_text(my_df, 'content',
'text_clean') my_df.head(10)
```

Hasil dari proses *cleansing* dan *case folding* ditunjukkan gambar 4.10 berikut

	userName	score	at	content	Label	text_clean
0	Krisna	1	2025-06-17 23:58:24	tidak stabil	Negatif	tidak stabil
1	Mutiara Putriningsih	5	2025-06-12 06:06:54	Tampilan antarmukanya agak kaku dan kurang. Mu...	Negatif	tampilan antarmukanya agak kaku dan kurang mun...
2	Saiful Anwar	4	2025-06-07 09:52:51	Aplikasinya sangat informatif dan mudah diguna...	Positif	aplikasinya sangat informatif dan mudah diguna...
3	Mutiara	5	2025-06-05 09:59:06	Apkikasinya sangat membantu banget loh bermanf...	Positif	apkikasinya sangat membantu banget loh bermanf...
4	Master PB	5	2025-06-05 06:23:32	Aplikasi ini sangat memudahkan saya dalam menc...	Positif	aplikasi ini sangat memudahkan saya dalam menc...
5	Muhammad Raditya Al Aza	5	2025-06-03 01:30:01	Aplikasinya ini bikin naik bus Trans Semarang ...	Positif	aplikasinya ini bikin naik bus trans semarang ...

Gambar 4. 10 *Hasil Proses Cleansing dan Case Folding*

b. Case Folding

Pada tahap *case folding*, semua huruf kapital akan dikonversi menjadi huruf kecil agar seluruh data ulasan memiliki format yang seragam. Pada code program tahap *case folding* ini digabungkan dengan tahap proses *cleansing* sebelumnya.

c. Remove Duplicate

Pada tahap *remove duplicate*, proses ini bertujuan untuk menghilangkan baris-baris yang memiliki duplikasi dalam dataframe 'my_df', yaitu ulasan yang terduplikasi atau memiliki kesamaan, sehingga hanya satu ulasan yang dipertahankan untuk digunakan dalam proses klasifikasi. Code program dibawah ini menunjukkan proses dari proses *remove duplicate* pada tabel 4.11.

Tabel 4. 11 *Code Proses Remove Duplicate*

```
my_df = my_df.drop_duplicates()
my_df = my_df.reset_index(drop=True)
my_df
```

Hasil dari proses *remove duplicate* pada data ulasan aplikasi trans semarang hasilnya tetap sama dan tidak mengurangi jumlah data yaitu sebanyak 500 ulasan, karena tidak ditemukan duplikasi atau kesamaan antar ulasan dalam dataframe tersebut. Hasilnya dapat ditunjukkan pada gambar 4.11 berikut.

	userName	score	at	content	Label	text_clean
0	Krisna	1	2025-06-17 23:58:24	tidak stabil	Negatif	tidak stabil
1	Mutiara Putriningsih	5	2025-06-12 06:06:54	Tampilan antarmukanya agak kaku dan kurang. Mu...	Negatif	tampilan antarmukanya agak kaku dan kurang mun...
2	Saiful Anwar	4	2025-06-07 09:52:51	Aplikasinya sangat informatif dan mudah diguna...	Positif	aplikasinya sangat informatif dan mudah diguna...
3	Mutiara	5	2025-06-05 09:59:06	Apkikasinya sangat membantu banget loh bermanf...	Positif	apkikasinya sangat membantu banget loh bermanf...
4	Master PB	5	2025-06-05 06:23:32	Aplikasi ini sangat memudahkan saya dalam menc...	Positif	aplikasi ini sangat memudahkan saya dalam menc...
...	...	...	...	...	...	...
495	Pengguna Google	5	2019-05-01 13:36:48	Sangat membantu mengetahui bus terdekat. Jadi ...	Positif	sangat membantu mengetahui bus terdekat jadi b...
496	Pengguna Google	4	2019-05-01 05:16:47	Karena saya tidak tinggal di Semarang saya bel...	Positif	karena saya tidak tinggal di semarang saya bel...
497	Pengguna Google	5	2019-04-26 03:50:29	amazing apk	Positif	amazing apk
498	Pengguna Google	5	2019-04-25 08:05:24	mantapppp	Positif	mantapppp
499	Pengguna Google	5	2019-04-25 07:48:10	mantab banget nih aplikasi	Positif	mantab banget nih aplikasi

500 rows × 6 columns

Gambar 4. 11 *Hasil Proses remove Duplicate*

d. Tokenizing

Tahap *tokenizing* adalah proses pemisahan atau pemecahan kata-kata dalam suatu kalimat, agar tahap selanjutnya tidak lagi memproses kalimat-kalimat melainkan memproses kata

demi kata. Pada tahap ini, pustaka 'nltlk' digunakan untuk melakukan tokenisasi pada teks yang telah dibersihkan dalam DataFrame, dimana setiap teks dalam kolom text_clean diubah menjadi daftar kata (tokens) menggunakan fungsi 'word_tokenize'. Code program pada tahap *tokenizing* ditunjukkan pada tabel 4.12 berikut.

Tabel 4. 12 *Code Proses Tokenizing*

```
import nltk
nltk.download('punkt')
nltk.download('punkt_tab')
from nltk.tokenize import sent_tokenize,
word_tokenize

my_df['text_tokens'] =
my_df['text_clean'].apply(word_tokenize)
my_df.head()
```

Selanjutnya, didapatkan hasil dari proses *tokenizing* tersebut ditunjukkan pada gambar 4.12 berikut.

	userName	score	at	content	Label	text_clean	text_tokens
0	Krisna	1	2025-06-17 23:58:24	tidak stabil	Negatif	tidak stabil	[tidak, stabil]
1	Mutiara Putriningsih	5	2025-06-12 06:06:54	Tampilan antarmukanya agak kaku dan kurang. Mu...	Negatif	tampilan antarmukanya agak kaku dan kurang mun...	[tampilan, antarmukanya, agak, kaku, dan, kura...
2	Saiful Anwar	4	2025-06-07 09:52:51	Aplikasinya sangat informatif dan mudah diguna...	Positif	aplikasinya sangat informatif dan mudah diguna...	[aplikasinya, sangat, informatif, dan, mudah, ...
3	Mutiara	5	2025-06-05 09:59:06	Apkikasinya sangat membantu banget loh bermanf...	Positif	apkikasinya sangat membantu banget loh bermanf...	[apkikasinya, sangat, membantu, banget, loh, b...
4	Master PB	5	2025-06-05 06:23:32	Aplikasi ini sangat memudahkan saya dalam menc...	Positif	aplikasi ini sangat memudahkan saya dalam menc...	[aplikasi, ini, sangat, memudahkan, saya, dala...

Gambar 4. 12 *Hasil Tahapan Proses Tokenizing*

e. Normalisasi Kata

Tahap normalisasi digunakan dengan tujuan mengubah kata-kata tidak baku serta kata yang mengandung singkatan menjadi kata yang sesuai dengan kamus standar KBBI. Pada proses ini diperlukan dengan mengimport beberapa *library* yang akan digunakan.

Tabel 4. 13 *Import Library Tahap Normalisasi*

```
import pandas as pd
import re
import csv
import string
```

Untuk proses normalisasi dilakukan dengan bantuan data pendukung yang memungkinkan pengubahan kata tidak baku menjadi kata baku yang tentunya sesuai dengan standar kamus KBBI. Dalam penelitian ini, penulis mengumpulkan berbagai sumber data pendukung berupa kumpulan data dalam format teks serta excel yang akan digunakan dalam proses normalisasi. Code proses tahapan dari normalisasi ditunjukkan pada tabel 4.14 berikut.

Tabel 4. 14 *Code Proses Normalisasi*

```
normalized_word =
pd.read_excel("kamuskatabakunormalisasi.xlsx")
normalized_string =
pd.read_csv('slangword.txt', sep='\t')
normalized_string = pd.read_csv('kbba.txt',
sep='\t')
normalized_string =
pd.read_csv('formalizationDict.txt', sep='\t')
normalized_word_dict = {}
for index, row in  normalized_word.iterrows():
    if        row.iloc[0]        not        in
    normalized_word_dict:
      normalized_word_dict[row.iloc[0]]      =
       row.iloc[1]

def normalized_term(document):
  return [normalized_word_dict[term] if term
in normalized_word_dict else term for term in
document]

my_df['text_normalized'] =
my_df['text_tokens'].apply(normalized_term)
my_df.head(50)
```

Pada tahap ini, dilakukan proses normalisasi menggunakan 4 data pendukung, yang didalamnya berisi kata tidak baku dengan padanan kata baku. Proses ini dilakukan dengan membaca data dari file-file tersebut dan menyimpannya dalam bentuk kamus (*dictionary*) yang memetakan kata tidak baku ke bentuk bakunya. Selanjutnya fungsi 'normalized_term()' digunakan pada setiap

dokumen dalam dataset untuk mengganti kata tidak baku dengan kata baku, sehingga menghasilkan teks yang telah dinormalisasi pada kolom 'text_normalized'. Hasil yang didapatkan dari proses normalisasi kata tersebut ditunjukkan pada gambar 4.13 berikut.

	userName	score	at	content	Label	text_clean	text_tokens	text_normalized
0	Krisna	1	2025-06-17 23:58:24	tidak stabil	Negatif	tidak stabil	[tidak, stabil]	[tidak, stabil]
1	Mutiara Putriningsih	5	2025-06-12 06:06:54	Tampilan antarmukanya agak kaku dan kurang. Mu...	Negatif	tampilan antarmukanya agak kaku dan kurang mun...	[tampilan, antarmukanya, agak, kaku, dan, kura...	[tampilan, antarmukanya, agak, kaku, dan, kura...
2	Saiful Anwar	4	2025-06-07 09:52:51	Aplikasinya sangat informatif dan mudah diguna...	Positif	aplikasinya sangat informatif dan mudah diguna...	[aplikasinya, sangat, informatif, dan, mudah, ...	[aplikasinya, sangat, informatif, dan, mudah, ...
3	Mutiara	5	2025-06-05 09:59:06	Apkikasinya sangat membantu banget loh bermanf...	Positif	apkikasinya sangat membantu banget loh bermanf...	[apkikasinya, sangat, membantu, banget, loh, b...	[apkikasinya, sangat, membantu, banget, loh, b...
4	Master PB	5	2025-06-05 06:23:32	Aplikasi ini sangat memudahkan saya dalam menc...	Positif	aplikasi ini sangat memudahkan saya dalam menc...	[aplikasi, ini, sangat, memudahkan, saya, dala...	[aplikasi, ini, sangat, memudahkan, saya, dala...
5	Muhammad Raditya Al Aza	5	2025-06-03 01:30:01	Aplikasinya ini bikin naik bus Trans Semarang ...	Positif	aplikasinya ini bikin naik bus trans semarang ...	[aplikasinya, ini, bikin, naik, bus, trans, se...	[aplikasinya, ini, bikin, naik, bus, trans, se...

Gambar 4. 13 *Hasil Proses Normalisasi*

f. Stopword Removal

Dalam tahap *stopword removal*, kata-kata yang tidak terlalu penting dan berpengaruh dalam sebuah ulasan akan dihapus. Penghapusan ini dilakukan karena kata-kata tersebut tidak mempengaruhi makna utama dalam sebuah kalimat ulasan, sehingga tidak mempengaruhi informasi yang disampaikan. Code program dalam proses *stopword* ditunjukkan pada tabel 4.15 berikut.

Tabel 4. 15 *Code Tahapan Proses Stopword Removal*

```
import nltk.corpus
nltk.download('stopwords')
from nltk.corpus import stopwords
stop = stopwords.words('indonesian')
def remove_stopwords(normalized):
   return [word for word in normalized if word not
in stop]
my_df['text_StopWord'] =
my_df['text_normalized'].apply(lambda x:
remove_stopwords(x))
my_df.head(50)
```

Dalam proses ini dilakukan dengan menggunakan pustaka 'nltk.corpus stopwords' dimana daftar stopword yang digunakan adalah Bahasa Indonesia dari Pustaka NLTK. Kemudian fungsi 'remove_stopwords()' untuk menyaring setiap kata dalam teks yang telah dinormalisasi dan menghapus kata-kata yang termasuk dalam daftar *stopwords*. Fungsi ini diterapkan pada kolom text_normalized sehingga menghasilkan kolom baru 'text_stopword' yang berisi teks yang hanya terdiri dari kata-kata yang memiliki makna penting. Hasil dari proses *stopword removal* ditunjukkan gambar 4.14 berikut.

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

	userName	score	at	content	Label	text_clean	text_tokens	text_normalized	text_Stopword
0	Krisna	1	2025-06-17 23:58:24	tidak stabil	Negatif	tidak stabil	[tidak, stabil]	[tidak, stabil]	[stabil]
1	Mutiara Putriningsih	5	2025-06-12 06:06:54	Tampilan antarmukanya agak kaku dan kurang. Mu...	Negatif	tampilan antarmukanya agak kaku dan kurang mun...	[tampilan, antarmukanya, agak, kaku, dan, kura...	[tampilan, antarmukanya, agak, kaku, dan, kura...	[tampilan, antarmukanya, kaku, userfriendly, m...
2	Saiful Anwar	4	2025-06-07 09:52:51	Aplikasinya sangat informatif dan mudah diguna...	Positif	aplikasinya sangat informatif dan mudah diguna...	[aplikasinya, sangat, informatif, dan, mudah, ...	[aplikasinya, sangat, informatif, dan, mudah, ...	[aplikasinya, informatif, mudah, memudahkan, p...
3	Mutiara	5	2025-06-05 09:59:06	Apkikasinya sangat membantu banget loh bermanf...	Positif	apkikasinya sangat membantu banget loh bermanf...	[apkikasinya, sangat, membantu, banget, loh, b...	[apkikasinya, sangat, membantu, banget, loh, b...	[apkikasinya, membantu, banget, loh, bermanfaa...
4	Master PB	5	2025-06-05 06:23:32	Aplikasi ini sangat memudahkan saya dalam menc...	Positif	aplikasi ini sangat memudahkan saya dalam menc...	[aplikasi, ini, sangat, memudahkan, saya, dala...	[aplikasi, ini, sangat, memudahkan, saya, dala...	[aplikasi, memudahkan, mencari, rute, bus, inf...
5	Muhammad Raditya Al Aza	5	2025-06-03 01:30:01	Aplikasinya ini bikin naik bus Trans Semarang ...	Positif	aplikasinya ini bikin naik bus trans semarang ...	[aplikasinya, ini, bikin, naik, bus, trans, se...	[aplikasinya, ini, bikin, naik, bus, trans, se...	[aplikasinya, bikin, bus, trans, semarang, pra...

Gambar 4. 14 *Hasil Proses Stopword Removal*

g. Stemming

Tahap *stemming* merupakan pengubahan kata berimbuhan menjadi bentuk kata dasar dan mengembalikan ke akar katanya. Dalam melakukan proses ini akan memanfaatkan *library Sastrawi*. Code penginstalan *library sastrawi* ditunjukkan tabel 4.16 berikut.

Tabel 4. 16 *Instalasi Library Sastrawi*

```
!pip install Sastrawi swifter
```

Setelah penginstallan *library sastrawi* dilakukan dilanjutkan dengan mengimport *library sastrawi* dengan pemanggilan kelas *stemmerfactory* ditunjukkan tabel 4.17 berikut.

Tabel 4. 17 *Code Pemanggilan Library Sastrawi*

```
from Sastrawi.Stemmer.StemmerFactory
import StemmerFactory
factory = StemmerFactory()
stemmer = factory.create_stemmer()
```

Dilanjutkan dengan menjalankan proses pada tahap *stemming,* code ditunjukkan tabel 4.18 berikut.

Tabel 4. 18 *Code Proses Stemming*

```
def stemmed_wrapper(term):
     return stemmer.stem(term)

term_dict = {}
hitung=0

for document in my_df['text_StopWord']:
    for term in document:
        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)
     hitung+=1
     print(hitung,":",term,":" ,term_dict[term])

print(term_dict)
print("------------------------")

def get_stemmed_term(document):
    return [term_dict[term] for term in document]

my_df['text_steamindo'] =
my_df['text_StopWord'].apply(lambda x:'
'.join(get_stemmed_term(x)))
my_df.head(20)
```

Dalam tahap ini digunakan fungsi ‘stemed_wrapper’ yang memanfaatkan ‘stemmer.stem’ untuk mengubah kata menjadi bentuk dasar. Setiap kata dalam kolom ‘text_Stopword’ diperiksa dan jika kata tersebut belum ada dikamus ‘term_dict’ maka kata tersebut akan distem dan disimpan dalam kamus. Fungsi ‘get_stemmed_term’ diterapkan pada kolom ‘text_Stopword menggunakan metode ‘apply’, sehingga menghasilkan kolom baru ‘text_steamindo’ yang berisi teks dengan kata-kata yang telah diubah ke bentuk dasarnya. Hasil dari proses *stemming* ditunjukkan gambar 4.15 berikut.

	userName	score	at	content	Label	text_clean	text_tokens	text_normalized	text_Stopword	text_steamindo
0	Krisna	1	2025-06-17 23:58:24	tidak stabil	Negatif	tidak stabil	[tidak, stabil]	[tidak, stabil]	[stabil]	stabil
1	Mutiara Putriningsih	5	2025-06-12 06:06:54	Tampilan antarmukanya agak kaku dan kurang. Mu...	Negatif	tampilan antarmukanya agak kaku dan kurang mun...	[tampilan, antarmukanya, agak, kaku, dan, kura...	[tampilan, antarmukanya, agak, kaku, dan, kura...	[tampilan, antarmukanya, kaku, userfriendly, m...	tampil antarmuka kaku userfriendly tarik
2	Saiful Anwar	4	2025-06-07 09:52:51	Aplikasinya sangat informatif dan mudah diguna...	Positif	aplikasinya sangat informatif dan mudah diguna...	[aplikasinya, sangat, informatif, dan, mudah, ...	[aplikasinya, sangat, informatif, dan, mudah, ...	[aplikasinya, informatif, mudah, memudahkan, p...	aplikasi informatif mudah mudah guna temu tran...
3	Mutiara	5	2025-06-05 09:59:06	Apkikasinya sangat membantu banget loh bermanf...	Positif	apkikasinya sangat membantu banget loh bermanf...	[apkikasinya, sangat, membantu, banget, loh, b...	[apkikasinya, sangat, membantu, banget, loh, b...	[apkikasinya, membantu, banget, loh, bermanfaa...	apkikasinya bantu banget loh manfaat anak seko...
4	Master PB	5	2025-06-05 06:23:32	Aplikasi ini sangat memudahkan saya dalam menc...	Positif	aplikasi ini sangat memudahkan saya dalam menc...	[aplikasi, ini, sangat, memudahkan, saya, dala...	[aplikasi, ini, sangat, memudahkan, saya, dala...	[aplikasi, memudahkan, mencari, rute, bus, inf...	aplikasi mudah cari rute bus informasi lengkap
5	Muhammad Raditya Al Aza	5	2025-06-03 01:30:01	Aplikasinya ini bikin naik bus Trans Semarang ...	Positif	aplikasinya ini bikin naik bus trans semarang ...	[aplikasinya, ini, bikin, naik, bus, trans, se...	[aplikasinya, ini, bikin, naik, bus, trans, se...	[aplikasinya, bikin, bus, trans, semarang, pra...	aplikasi bikin bus trans semarang praktis bing...

Gambar 4. 15 *Hasil Proses Stemming*

Setelah dilakukannya beberapa proses dalam tahap *preprocessing*, sehingga tampilan hasil yang telah

didapatkan berupa sebuah data bersih yang sudah siap untuk digunakan dalam klasifikasi lebih lanjut. Jumlah data setelah melewati tahap *preprocessing* hasilnya tetap yaitu sebanyak 500 ulasan, dikarenakan tidak adanya baris dalam dataframe yang memiliki duplikat yang sama dengan data lainnya. Hasil tahap *preprocessing* ditunjukkan pada gambar 4.16 berikut.

```
#Menampilkan data setelah melalui tahapan text preprocessing
my_df = pd.read_csv("HasilPreprocessing.csv")
my_df
```

	userName	score	at	content	Label	text_clean	text_tokens	text_normalized	text_StopWord	text_steamindo
0	Krisna	1	2025-06-17 23:58:24	tidak stabil	Negatif	tidak stabil	['tidak', 'stabil']	['tidak', 'stabil']	['stabil']	stabil
1	Mutiara Putriningsih	5	2025-06-12 06:06:54	Tampilan antarmukanya agak kaku dan kurang. Mu...	Negatif	tampilan antarmukanya agak kaku dan kurang mun...	['tampilan', 'antarmukanya', 'agak', 'kaku', '...	['tampilan', 'antarmukanya', 'agak', 'kaku', '...	['tampilan', 'antarmukanya', 'kaku', 'userfrie...	tampil antarmuka kaku userfriendly tarik
2	Saiful Anwar	4	2025-06-07 09:52:51	Aplikasinya sangat informatif dan mudah diguna...	Positif	aplikasinya sangat informatif dan mudah diguna...	['aplikasinya', 'sangat', 'informatif', 'dan',...	['aplikasinya', 'sangat', 'informatif', 'dan',...	['aplikasinya', 'informatif', 'mudah', 'memuda...	aplikasi informatif mudah mudah guna temu tran...
3	Mutiara	5	2025-06-05 09:59:06	Apkikasinya sangat membantu banget loh bermanf...	Positif	apkikasinya sangat membantu banget loh bermanf...	['apkikasinya', 'sangat', 'membantu', 'banget'...	['apkikasinya', 'sangat', 'membantu', 'banget'...	['apkikasinya', 'membantu', 'banget', 'loh', '...	apkikasinya bantu banget loh manfaat anak seko...
4	Master PB	5	2025-06-05 06:23:32	Aplikasi ini sangat memudahkan saya dalam menc...	Positif	aplikasi ini sangat memudahkan saya dalam menc...	['aplikasi', 'ini', 'sangat', 'memudahkan', 's...	['aplikasi', 'ini', 'sangat', 'memudahkan', 's...	['aplikasi', 'memudahkan', 'mencari', 'rute', ...	aplikasi mudah cari rute bus informasi lengkap
...	...	...	...	...	...	...	...	...	...	...
495	Pengguna Google	5	2019-05-01 13:36:48	Sangat membantu mengetahui bus terdekat. Jadi ...	Positif	sangat membantu mengetahui bus terdekat jadi b...	['sangat', 'membantu', 'mengetahui', 'bus', 't...	['sangat', 'membantu', 'mengetahui', 'bus', 't...	['membantu', 'bus', 'terdekat']	bantu bus dekat
496	Pengguna Google	4	2019-05-01 05:16:47	Karena saya tidak tinggal di Semarang saya bel...	Positif	karena saya tidak tinggal di semarang saya bel...	['karena', 'saya', 'tidak', 'tinggal', 'di', '...	['karena', 'saya', 'tidak', 'tinggal', 'di', '...	['tinggal', 'semarang', 'mencoba', 'langsung',...	tinggal semarang coba langsung lapang kilas ut...

Gambar 4. 16 *Hasil Setelah Tahap Preprocessing*

Selanjutnya, dapat diketahui jumlah ulasan dalam setiap kategori label sentimen setelah melalui beberapa tahap *preprocessing*. Sehingga dalam bahasa pemrograman, didapatkan jumlah tiap kategori label sentimen pada aplikasi Trans Semarang, diurutkan dengan jumlah paling besar yaitu terdapat 261 ulasan berlabel positif dan 239 berlabel negatif. Ditunjukkan pada gambar 4.17 berikut.

```
[35] my_df['Label'].unique() #Cek jenis sentimen
array(['Negatif', 'Positif'], dtype=object)

my_df.Label.value_counts() #Menghitung jumla
```

Label	count
Positif	261
Negatif	239

dtype: int64

Gambar 4. 17 *Jumlah Total Label Sentimen Tiap Kategori*

D. Seleksi Fitur dan Splitting Data

Setelah dilakukan tahap *preprocessing*, tentunya ulasan tersebut masih mengandung teks. Agar dapat diklasifikasikan, teks tersebut harus diubah menjadi representasi numerik melalui seleksi fitur kata. Untuk melakukan seleksi fitur, membutuhkan beberapa library yang nantinya akan digunakan, seperti *library sklearn* atau *library sckit-learn*. Adapun fungsi dari library ini untuk mendukung proses *preprocessing* data serta melakukan *training* model yang diperlukan dalam *machine learning* dan *data science* (Pedregosa et al., 2011).

Dalam tahap seleksi fitur ini, data terlebih dahulu dipisahkan menjadi dua bagian, fitur (x) yang merupakan data dari kolom 'text_steamindo' dari proses stemming, sebagai masukan dalam proses

pelatihan model klasifikasi. Sementara label (y), diambil dari kolom 'label' yang menunjukkan kategori sentimen sebagai target output pada proses model klasifikasi. Proses ini ditunjukkan pada tabel 4.19 berikut.

Tabel 4. 19 *Proses Pemisahan Data*

```
X = my_df['text_steamindo']
y = my_df['Label']
```

Beberapa modul dari *sklearn* akan digunakan seperti pada kelas *train_test_split,* serta *TfidfVectorizer*. Setelahnya data akan dibagi pada proses *splitting* data yang akan membagi data uji dengan data latih untuk membangun model klasifikasi.

Pembagian data dilakukan dengan rasio 20% dari total data sebagai data uji guna mengevaluasi performa model, dengan menetapkan parameter 'test_size' sebesar 0.20 atau 20% data sebagai data uji. Code proses tahapan membagi data ditunjukkan pada tabel 4.20 berikut.

Tabel 4. 20 *Proses Splitting Data*

```
from sklearn.model_selection import
train_test_split
X_train,    X_test,    y_train,    y_test    =
train_test_split(X_tfidf,   y,   test_size=0.2,
random_state=42)
```

Setelah dilakukan proses *splitting* data, lalu akan menampilkan jumlah data yang telah terbagi menjadi data latih (*training* data) dan data uji (*testing* data). Code proses jumlah data latih dan data uji ditunjukkan pada tabel 4.21 berikut.

Tabel 4. 21 *Code Jumlah Data Latih dan Data Uji*

```
print("X_train = ", X_train.shape[0])
print("y_train = ", len(y_train))

print("X_test = ", X_test.shape[0])
print("y_test = ", len(y_test))
```

Dari hasil pembagian tersebut, digunakan sebanyak 400 data latih dan data uji sebanyak 100 data. Hasilnya ditunjukkan pada gambar 4.18 berikut.

```
X_train =  400
y_train =  400
X_test =  100
y_test =  100
```

Gambar 4. 18 *Jumlah Data Latih dan Data Uji*

Pada tahapan proses seleksi fitur, metode *Term Frequency-Inverse Document Frequency* (TFIDF) digunakan untuk menentukan bobot masing-masing kata dalam data. Pembuatan *word vector* dan pembobotan kata dilakukan dengan bantuan pustaka *sklearn* melalui kelas *TfidfVectorizer*, yang bertujuan

merepresentasikan data dalam bentuk vektor, sehingga dapat digunakan dalam proses klasifikasi secara lebih efektif.

Selain itu, juga digunakan parameter 'ngram_range=(1,2)' agar proses seleksi fitur tidak hanya mempertimbangkan kata tunggal tetapi juga pasangan kata berurutan. Code proses ini ditunjukkan pada tabel 4.22 berikut.

Tabel 4. 22 *Code Seleksi Fitur TFIDF*

```
from sklearn.feature_extraction.text import
TfidfVectorizer
tfidf = TfidfVectorizer(ngram_range=(1, 2))
X_tfidf = tfidf.fit_transform(X)
```

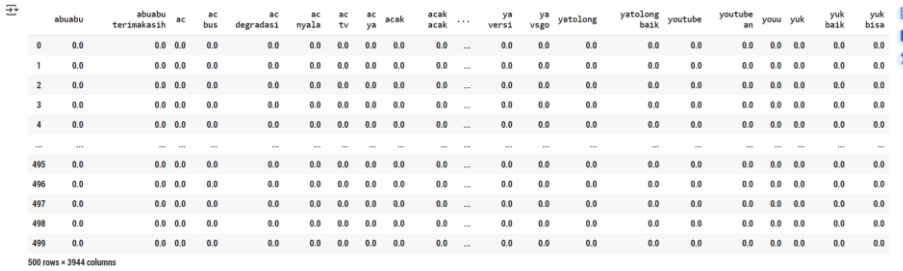
Proses selanjutnya mengonversi hasil seleksi fitur TF-IDF ke dalam bentuk dataframe. Sebelumnya, data latih (X_train) dan data uji (X_test) telah ditransformasikan ke dalam bentuk matriks TFIDF. Transformasi ini menghasilkan representasi numerik berupa bobot dari setiap kata yang menjadi fitur, dimana bobot tersebut menunjukkan tingkat kepentingan kata dalam sebuah dokumen berdasarkan frekuensi dan distribusinya dalam seluruh korpus data. Hasil matriks transformasi yang masih berbentuk array dikonversi dalam bentuk dataframe agar lebih mudah

dianalisis. Code pada proses ini ditunjukkan tabel 4.23 berikut.

Tabel 4. 23 *Code Konversi Seleksi Fitur TFIDF*

```
features_df = pd.DataFrame(X_tfidf.toarray(),
columns=tfidf.get_feature_names_out())
features_df
```

Hasil dari proses konversi seleksi fitur TFIDF tersebut akan berupa tabel dimana setiap baris mewakili suatu ulasan, dan setiap kolom menunjukkan bobot pentingnya sebuah kata dalam ulasan tersebut berdasarkan metode TFIDF. Proses ini memungkinkan teks ulasan untuk diubah menjadi format numerik yang lebih mudah diproses dalam klasifikasi lebih lanjut. Tampilan hasil dari pembobotan TFIDF pada tahap seleksi fitur ditunjukkan gambar 4.19 berikut.



	abuabu	abuabu terimakasih	ac	ac bus	ac degradasi	ac nyala	ac tv	ac ya	acak	acak acak	...	ya versi	ya vsgo	yatolong	yatolong baik	youtube	youtube an	youu	yuk	yuk baik	yuk bisa
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
495	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
496	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
497	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
498	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
499	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

500 rows × 3944 columns

Gambar 4. 19 *Tampilan Hasil Seleksi Fitur TFIDF*

Selanjutnya akan diterapkan contoh dari penelitian yang menerapkan perhitungan secara manual dalam

pembobotan TFIDF menggunakan tiga komentar ulasan bisa dilihat sebagai berikut:

(Doc1)= 'Membantu sekali saat keluarga kami liburan di Semarang'.

(Doc2)= 'Kadang suka mancet sendiri apknya'.

(Doc3)= 'gk kedetect kalo ada feeder yg dateng, sy kecewa bgt jd ketinggalan feeder'.

Setelah melalui tahap *preprocessing*, hasil dari dokumen tersebut adalah:

(Doc1)=['bantu','keluarga','libur','semarang']

(Doc2)=['kadang','suka','mancet,'aplikasi']

(Doc3)=['kedetect','feeder','kecewa','banget','tinggal,'feeder']

Kemudian dilanjutkan dengan melakukan perhitungan menggunakan metode TFIDF untuk membentuk vektor kata yang nantinya akan diberikan bobot nilai. Dalam metode proses TFIDF terdapat dua kata TF dan IDF. TF menunjukkan jumlah kemunculan suatu kata dalam dokumen, dan IDF berfungsi untuk mengurangi bobot kata yang sering muncul di seluruh dokumen agar tidak terlalu berpengaruh. Perhitungan TFIDF dimulai dengan menghitung nilai TF secara manual. Contoh perhitungan TF secara manual ditunjukkan pada tabel 4.24 berikut.

Tabel 4. 24 *Contoh Perhitungan TF*

Tokens	**TF**		
	Doc 1	**Doc 2**	**Doc 3**
bantu	1	0	0
keluarga	1	0	0
libur	1	0	0
semarang	1	0	0
kadang	0	1	0
suka	0	1	0
mancet	0	1	0
aplikasi	0	1	0
kedetect	0	0	1
feeder	0	0	2
kecewa	0	0	1
banget	0	0	1
tinggal	0	0	1

Setelah memperoleh nilai TF, langkah selanjutnya adalah menghitung nilai DF sebelum melanjutkan ke perhitungan IDF. DF ialah jumlah dokumen yang mengandung kata tersebut. Contoh perhitungan DF ditunjukkan pada tabel 4.25 berikut.

Tabel 4. 25 *Contoh Perhitungan DF*

Terms	**TF**			**DF**
	Doc 1	**Doc 2**	**Doc 3**	
bantu	1	0	0	1
keluarga	1	0	0	1
libur	1	0	0	1
semarang	1	0	0	1
kadang	0	1	0	1
suka	0	1	0	1

mancet	0	1	0	1
aplikasi	0	1	0	1
kedetect	0	0	1	1
feeder	0	0	2	1
kecewa	0	0	1	1
banget	0	0	1	1
tinggal	0	0	1	1

Setelah memperoleh nilai TF dan nilai DF dari tiga komentar ulasan dalam tabel, diketahui bahwa jumlah dokumen atau (D) adalah 3. Lalu, perhitungan IDF dan TF-IDF akan dilakukan berdasarkan rumus yang telah dijabarkan di bab sebelumnya. Perhitungan IDF dan TFIDF secara manual dapat ditunjukkan pada tabel 4.26 berikut

Tabel 4. 26 *Hasil Perhitungan IDF dan TFIDF*

Terms	**DF**	**D/DF (D=3)**	**IDF (log(D/Df))**	**IDF +1**	**TF*IDF**		
					D1	**D2**	**D3**
bantu	1	3	Log 3 = 0,477	1,477	1,477	0	0
keluarga	1	3	Log 3 = 0,477	1,477	1,477	0	0
libur	1	3	Log 3 = 0,477	1,477	1,477	0	0
semarang	1	3	Log 3 = 0,477	1,477	1,477	0	0
kadang	1	3	Log 3 = 0,477	1,477	0	1,477	0

suka	1	3	Log 3 = 0,477	1,477	0	1,477	0
mancet	1	3	Log 3 = 0,477	1,477	0	1,477	0
aplikasi	1	3	Log 3 = 0,477	1,477	0	1,477	0
kedetect	1	3	Log 3 = 0,477	1,477	0	0	1,477
feeder	1	3	Log 3 = 0,477	1,477	0	0	1,477
kecewa	1	3	Log 3 = 0,477	1,477	0	0	1,477
banget	1	3	Log 3 = 0,477	1,477	0	0	1,477
tinggal	1	3	Log 3 = 0,477	1,477	0	0	1,477

Setelah berhasil memperoleh nilai dari perhitungan TFIDF secara manual, jika hasil tersebut direpresentasikan dalam bentuk array bukan dalam bentuk tabel, adapun hasil dari bentuk array pada perhitungan manual adalah seperti berikut:

Array([
[1,477, 1,477, 1,477, 1,477, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 1,477, 1,477, 1,477, 1,477, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 1,477, 1,477, 1,477, 1,477, 1,477]
])

Hasil tersebut menunjukkan bahwa setiap baris pada array tersebut merepresentasikan masing-masing dokumen, sedangkan tiap kolom menggambarkan setiap kata yang terdapat dalam keseluruhan teks.

E. Klasifikasi Naive Bayes Classifier

Setelah proses pembagian data latih dan data uji selesai dilakukan, pada tahap pemodelan ini proses klasifikasi akan dilakukan dengan menggunakan metode Naive Bayes Classifier dengan dua kategori sentimen, yaitu positif dan negatif. Tujuan dari tahap klasifikasi ini untuk memprediksi label pada setiap data uji berdasarkan model yang telah dibangun.

Data latih yang diberikan sebesar 80% dan dilanjutkan dengan proses pengujian menggunakan data uji untuk mengukur ketetapan sistem dalam melakukan klasifikasi. Dalam proses klasifikasi, digunakan *library scikit-learn* (*sklearn*), sebagaimana yang telah diterapkan pada tahap seleksi fitur sebelumnya. Beberapa modul dari *sklearn* yang digunakan dalam tahap ini yaitu *MultinomialNB*, serta metrik evaluasi seperti *accuracy_score, precision_score, recall_score, f1-score, classification_report* dan *confusion matrix*.

Proses dimulai dengan menyiapkan *library sklearn*, yang mana pada google colab sudah menyediakan *library* ini secara bawaan. Selanjutnya mengimport *library* dengan pemanggilan beberapa kelas yang diperlukan untuk proses klasifikasi. Code Program import *library sklearn* dalam tahapan klasifikasi

ditunjukkan pada tabel 4.27 berikut.

Tabel 4. 27 *Code Import Library Tahapan Klasifikasi*

```
from sklearn.feature_extraction.text import
TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score,
precision_score, recall_score, f1_score
from sklearn.metrics import
classification_report
from sklearn.metrics import confusion_matrix
```

Setelah melakukan pengimportan library akan dilakukan proses pengklasifikasian dengan menggunakan metode naive bayes classifier. Code program pada proses klasifikasi naive bayes classifier ditunjukkan pada tabel 4.28 berikut.

Tabel 4. 28 *Code Proses Klasifikasi Naive Bayes Classifier*

```
clf = MultinomialNB()
clf.fit(X_train, y_train)
predicted = clf.predict(X_test)

print("MultinomialNB Accuracy:",
accuracy_score(y_test,predicted))
print("MultinomialNB Precision:",
precision_score(y_test,predicted,
average='weighted'))
print("MultinomialNB Recall:",
recall_score(y_test,predicted, average='weighted'))
print("MultinomialNB f1_score:",
f1_score(y_test,predicted, average='weighted'))

print(f'confusion_matrix:\n
{confusion_matrix(y_test, predicted)}')
```

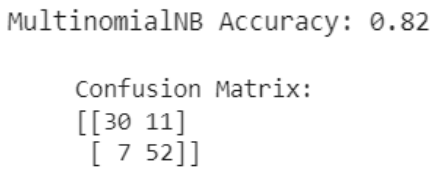
```
print('=========================================
=======\n')
print(classification_report(y_test,       predicted,
zero_division=0))
```

Proses klasifikasi menggunakan multinomial naive bayes dimulai dengan inisialisasi model menggunakan 'MultinomialNB()' yang kemudian dilatih dengan data latih 'X_train' dan label 'y_train' dengan metode .fit(). Setelah model selesai dilatih, dilakukan prediksi terhadap data uji 'X_test' menggunakan .predict() untuk mendapatkan hasil klasifikasi. Selanjutnya evaluasi model dilakukan dengan beberapa metriks yaitu akurasi, *recall*, presisi, *f1-score* dimana semua metriks dihitung dengan rata-rata berbobot (average='weighted').

F. Pengujian Model

Setelah proses klasifikasi dan pelatihan model dilakukan, juga dilanjutkan dengan proses uji model yang akan menghasilkan *confusion matrix* untuk mengevaluasi kinerjanya. Pada Pengujian ini dilakukan dengan membuat *confusion matrix* 2x2 dengan dua kategori sentimen yang digunakan, yaitu positif dan negatif agar dapat diketahui nilai akurasinya. Maka dapat diketahui hasil proses uji model dengan *confusion*

matrix 2x2 ditunjukkan gambar 4.20 berikut.



Gambar 4. 20 *Hasil Uji Model Confusion Matrix*

Dari hasil pengujian model menggunakan *confusion matrix* tersebut, diperoleh bahwa nilai akurasi sebesar 0.82 atau 82%. Selanjutnya, nanti akan dilanjutkan dengan proses perhitungan evaluasi dari performa model yang digunakan.

G. Evaluasi Performa Model

Tahap evaluasi performa dilakukan untuk mengetahui nilai dari performa model yang digunakan. Untuk mengetahui performa tersebut, akan dilakukan dengan menghitung metrik seperti akurasi, *precision*, *recall*, dan *f1-score*. Perhitungan ini didasarkan pada perhitungan *confusion matrix*. Pada tabel nilai perhitungan *confusion matrix* yang dihasilkan dari proses klasifikasi model, jika dijumlahkan semua nilainya sebanyak 100 data. Jumlah nilai di *confusion matrix* tidak 500 data atau sebanyak jumlah data yang digunakan, hal ini dikarenakan nilai yang ada pada

confusion matrix hanya akan menampilkan jumlah hasil dari data uji/*testing* yang digunakan pada model. Pada penelitian ini, jumlah data uji pada tahap *splitting* yaitu sebanyak 100 data uji, maka jumlah semua nilai yang ada pada perhitungan *confusion matrix* juga 100 data. Berikut ini merupakan tabel yang menampilkan hasil *confusion matrix* 2x2 yang diperoleh dari tahap pengujian model sebelumnya.

Tabel 4. 29 *Hasil Confusion Matrix*

		Predicted Class	
		Negatif	*Positif*
True Class	*Negatif*	30	11
	Positif	7	52

Selanjutnya menghitung nilai performa model secara manual. Hasil perhitungan akurasi berdasarkan tabel *confusion matrix* akan dilakukan secara manual dan dijelaskan dalam perhitungan berikut:

$$\text{Akurasi} = \frac{TPos+TNeg}{TPos+TNeg+FPos+FNeg} x\ 100\% \qquad (4.1)$$

$$\text{Akurasi} = \frac{52+30}{52+30+11+7} x\ 100\% \qquad (4.2)$$

$$\text{Akurasi} = \frac{82}{100} \text{x } 100\% \qquad (4.3)$$

Akurasi = 0,82 x 100%

Akurasi = 82%

Setelah memperoleh hasil akurasi, dilanjutkan dengan menghitung metrik performa model yang lainnya, yaitu *precission*, *recall*, dan *f1-score*. Untuk menghitungnya, harus diperlukan penentuan nilai *true positif*, *true negatif*, *false positif*, dan *false negatif* dari *confusion matrix* 2x2. Seperti yang ditunjukkan pada tabel berikut:

a. Kelas Negatif

Tabel 4. 30 *Perhitungan Nilai Kelas Negatif*

True\Predic	**Negatif**	**Tidak Negatif**
Negatif	TPos = 30	FNeg = 11
Tidak Negatif	FPos = 7	TNeg = 52

b. Kelas Positif

Tabel 4. 31 *Perhitungan Nilai Kelas Positif*

True\Predic	**Positif**	**Tidak Positif**
Positif	TPos = 52	FNeg = 7
Tidak Positif	FPos = 11	TNeg = 30

Sehingga, dari tabel diatas dapat diperoleh hasil dari nilai *true positif*, *true negatif*, *false positif*, serta *false negatif*. Hal ini akan mempermudah perhitungan nilai *precission*, *recall,* dan *f1-score* menggunakan rumus

yang telah dijelaskan pada bab sebelumnya. Hasil perhitungan tersebut ditunjukkan pada tabel 4.32 berikut.

Tabel 4. 32 *Hasil Perhitungan Performa Model*

Kelas	**(TP) True Positive**	**(FP) False Positive**	**(FN) False Negative**	***Precission***	***Recall***	***F1-*** **Sore**
Negatif	30	7	11	0,81%	0,73%	0,77%
Positif	52	11	7	0,83%	0,88%	0,85%

Berdasarkan tabel diatas, didapatkan nilai *precission*, *recall*, serta *f1*-score untuk masing-masing kelas sentimen. Pada kelas sentimen negatif, untuk nilai *precission* mencapai sebesar 81%, nilai *recall* sebesar 73%, dan *f1*-score sebesar 77%. Dan, kelas sentimen positif nilai *precission* sebesar 83%, *recall* sebesar 88%, dan *f1*-score sebesar 85%. Perhitungan *presisi*, *recall*, dan *f1-score* pada tiap kelas juga dilakukan secara manual sebagai berikut:

Perhitungan *precision* secara manual:

$$Precission = \frac{TP}{TP+FP} \quad (4.4)$$

$$Negatif = \frac{30}{30+7} = \frac{30}{37} = 0.810 = 0.81\% \quad (4.5)$$

$$Positif = \frac{52}{52+11} = \frac{52}{63} = 0.825 = 0.83\% \quad (4.6)$$

Perhitungan *recall* secara manual:

$$Recall = \frac{TP}{TP+FN} \quad (4.7)$$

$$Negatif = \frac{30}{30+11} = \frac{30}{41} = 0.731 = 0.73\% \quad (4.8)$$

$$Positif = \frac{52}{52+7} = \frac{52}{59} = 0.881 = 0.88\% \quad (4.9)$$

Perhitungan *f1-score* secara manual:

$$f1 - score = 2 * \frac{presisi*recall}{presisi+recall} \quad (4.10)$$

$$Negatif = 2 * \frac{0.810*0.731}{0.810+0.732} = 2 * \frac{0.5913}{1.542} = 0.766 = 0.77\% \quad (4.11)$$

$$Positif = 2 * \frac{0.825*0.881}{0.825+0.881} = 2 * \frac{0.7268}{1.706} = 0.852 = 0.85\% \quad (4.12)$$

Perhitungan pada nilai akurasi, *precission*, *recall*, dan *f1*-score juga dilakukan melalui sistem dengan code program yang ditunjukkan pada tabel 4.33 berikut.

Tabel 4. 33 *Code Hitung Naive Bayes Classifier Sistem*

```
import pandas as pd
from      sklearn.feature_extraction.text      import
TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from    sklearn.metrics    import    accuracy_score,
precision_score, recall_score, f1_score
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix

clf = MultinomialNB()
clf.fit(X_train, y_train)
predicted = clf.predict(X_test)

print("MultinomialNB                        Accuracy:",
accuracy_score(y_test,predicted))
print("MultinomialNB                       Precision:",
precision_score(y_test,predicted,
average='weighted'))
print("MultinomialNB                          Recall:",
recall_score(y_test,predicted, average='weighted'))
print("MultinomialNB                        f1_score:",
f1_score(y_test,predicted, average='weighted'))

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

Selanjutnya, dilakukan evaluasi terhadap performa model dengan melihat nilai akurasi, presisi, *recall*, dan *f1*-score untuk masing-masing kelas sentimen. Nilai-nilai tersebut diperoleh berdasarkan proses perhitungan yang dilakukan melalui code sistem yang telah dibuat tersebut. Hasil lengkap dari evaluasi performa dari model beserta *confusion matrix* nya

ditunjukkan pada gambar 4.21 berikut.

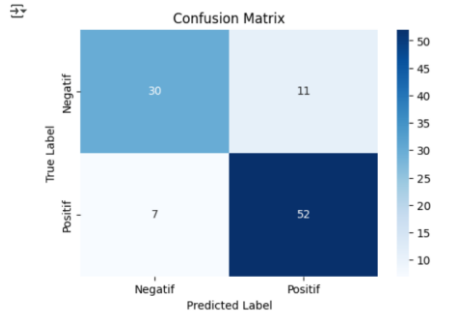
```
MultinomialNB Accuracy: 0.82
MultinomialNB Precision: 0.8194165594165593
MultinomialNB Recall: 0.82
MultinomialNB f1_score: 0.8183354350567466

Confusion Matrix:
[[30 11]
 [ 7 52]]
=====================================================

              precision    recall  f1-score   support

     Negatif       0.81      0.73      0.77        41
     Positif       0.83      0.88      0.85        59

    accuracy                           0.82       100
   macro avg       0.82      0.81      0.81       100
weighted avg       0.82      0.82      0.82       100
```



Gambar 4. 21 *Hasil Perhitungan Evaluasi Performa Model*

Dari hasil evaluasi performa model naive bayes classifier yang digunakan, diperoleh bahwa kemampuan model dalam memprediksi setiap kelas sentimen menunjukkan hasil yang baik. Sehingga dapat dikatakan, pada aspek ketepatan atau *precision*, sistem mencapai nilai sebesar 81% untuk kelas sentimen

negatif dan 83% untuk sentimen positif. Sementara itu, pada aspek *recall*, diperoleh nilai sebesar 73% untuk sentimen negatif dan 88% untuk sentimen positif.

Secara keseluruhan, sistem menghasilkan rata-rata nilai sebesar 82% untuk tingkat akurasi, 81% untuk *precision*, 82% untuk *recall*, dan 81% untuk *f1*-score. Nilai-nilai ini menggambarkan bahwa performa model sudah cukup baik dalam mengklasifikasikan ulasan pengguna.

H. Implementasi Testing Prediksi Sentimen

Setelah dilakukan proses klasifikasi dan evaluasi performa menggunakan model naive bayes classifier, tahap selanjutnya dilakukan testing prediksi label sentimen. Pengujian ini bertujuan untuk mengetahui kemampuan model dalam memprediksi label sentimen dari sebuah data ulasan yang dimasukkan oleh pengguna ke dalam sistem. Pada tahap ini, pengguna dapat memberikan input berupa kalimat ulasan, kemudian model akan memproses input tersebut dan mengklasifikasikannya ke dalam kategori label sentimen positif atau negatif. Code untuk testing prediksi model ditunjukkan pada tabel 4.34 berikut.

Tabel 4. 34 *Code Testing Prediksi Sentimen*

```
new_text = input("Masukkan Teks Yang
Ingin Di Prediksi: ")

new_text_vec =
tfidf.transform([new_text])

predicted_sentimen =
clf.predict(new_text_vec)

if predicted_sentimen[0] == "positif":
   sentiment_label = "positif"
elif predicted_sentimen[0] == "negatif":
   sentiment_label = "negatif"
print("Hasil prediksi sentimen:",
predicted_sentimen[0])
```

Sebagai contoh, dilakukan pengujian dengan satu komentar ulasan yang dimasukkan kedalam sistem dengan kalimat, 'aplikasinya sangat bagus dan membantu warga semarang'. Kalimat tersebut mengandung makna positif dan model berhasil memprediksinya sebagai ulasan dalam kategori sentimen positif. Hasil dari prediksi klasifikasi sentimen ini ditunjukkan pada gambar 4.22 berikut.

Gambar 4. 22 *Hasil Prediksi Klasifikasi Sentimen*

I. Implementasi Hasil Visualisasi

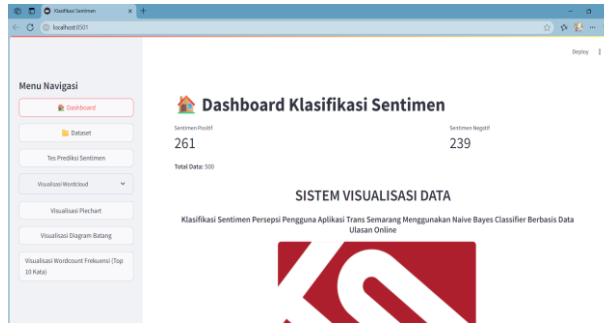
Setelah didapatkan performa hasil dari klasifikasi model, langkah terakhir akan dilakukan visualisasi dari hasil sentimen tersebut. Visualisasi ini mencakup *wordcloud* pada masing-masing kategori sentimen serta visualisasi dalam bentuk diagram.

Hasil visualisasi ini akan diimplementasikan ke dalam tampilan berbasis *web* dengan menggunakan *framework streamlit* dari *pyhton.*

Tampilan sistem berbasis *web* yang dibangun ini selain menampilkan hasil visualisasi data sentimen juga menyediakan fitur untuk menguji prediksi sentimen berdasarkan ulasan yang dimasukkan oleh pengguna. Berikut merupakan implementasi tampilan dari sistem *web streamlit* yang dibangun:

A. Tampilan Halaman Dashboard

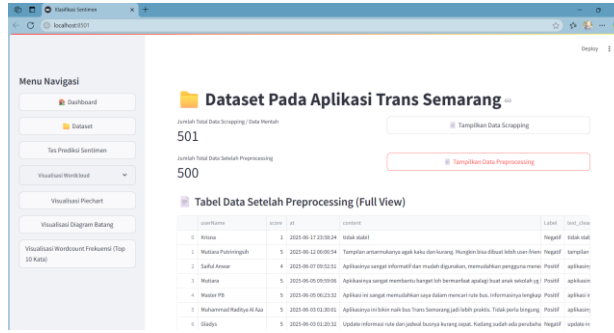
Pada halaman dashboard sistem visualisasi data ini untuk menampilkan jumlah data dari masing-masing label sentimen positif dan negatif serta total seluruh dataset. Tampilan halaman dashboard ditunjukkan pada gambar 4.23 berikut.



Gambar 4. 23 *Tampilan Halaman Dashboard*

B. Tampilan Halaman Dataset

Pada halaman dataset untuk menampilkan jumlah data mentah (hasil *scrapping*) dan jumlah data yang telah melalui tahap preprocessing (data bersih). Pada menu ini, juga bisa menampilkan seluruh isi data yang berhasil didapatkan. Tampilan halaman dataset ditunjukkan pada gambar 4.24 berikut.



Gambar 4. 24 *Tampilan Halaman Dataset*

C. Tampilan Halaman Testing Prediksi Sentimen Pada Sistem

Halaman menu pada fitur testing prediksi sentimen digunakan untuk menguji kemampuan model dalam memprediksi label sentimen dari suatu kalimat ulasan yang dimasukkan oleh pengguna ke dalam sistem. Model akan memproses input tersebut dan mengklasifikasikannya ke dalam kategori label sentimen tertentu. Proses prediksi ini dilakukan dengan memuat model klasifikasi 'naive_bayes_model.joblib' dan vektorisasi 'tfidf_vectorizer.joblib' yang sebelumnya telah dibuat melalui proses pelatihan di Google Colab. Tampilan halaman testing prediksi sentimen ditunjukkan pada gambar 4.25 berikut.

Gambar 4. 25 *Tampilan Prediksi Sentimen*

Hasil tersebut menunjukkan bahwa kalimat ulasan yang dimasukkan berupa kalimat bernada positif, dan model berhasil memprediksinya ke dalam kategori label sentimen positif.

D. Tampilan Halaman Hasil Visualisasi Wordcloud

Halaman menu visualisasi *wordcloud* akan menampilkan representasi visual dari sentimen dalam bentuk *wordcloud* pada setiap kategori sentimen. Pada visualisasi *wordcloud*, ukuran kata dalam *wordcloud* menunjukkan frekuensi kemunculannya, kata yang lebih besar dari kata yang lainnya memiliki frekuensi lebih besar, juga semakin sering kata tersebut muncul dalam ulasan pengguna aplikasi (Agusia et al., 2024).

Informasi ini dapat memberikan wawasan yang berguna bagi pihak pengembang aplikasi Trans Semarang dalam memahami persepsi pengguna terhadap layanan aplikasi. Dengan melihat kata-kata yang paling sering muncul dalam ulasan, pengembang dapat mengidentifikasi kelebihan dan kelemahan aplikasi, guna perbaikan kualitas aplikasi yang sesuai dengan kepuasan pengguna (Adela et al., 2024).

1) Wordcloud Ulasan Aplikasi

Dalam visualisasi *wordcloud* untuk menampilkan gambaran umum dari keseluruhan ulasan aplikasi, dilakukan proses dengan menggabungkan seluruh teks ulasan yang telah melalui tahap *preprocessing* dari proses ''text_steamindo''. Tampilan visualisasi *wordcloud* keseluruhan, menampilkan kata yang paling sering diungkapkan oleh para pengguna ulasan aplikasi Trans Semarang secara umum. Dari hasil yang didapatkan, bahwa contoh pada kata "aplikasi", "bantu", "rute", "bis", dan lainnya tampak menonjol yang menujukkan kata tersebut sering digunakan oleh para pengguna pada ulasan aplikasi.

Hal ini menunjukkan bahwa tanggapan pengguna cenderung berkaitan dengan kualitas aplikasi secara umum, serta informasi mengenai rute dan layanan bus yang disediakan, menjadi topik yang cukup sering diulas oleh pengguna. Tampilan hasil *wordcloud* keseluruhan ulasan aplikasi ditunjukkan pada gambar 4.26 berikut.

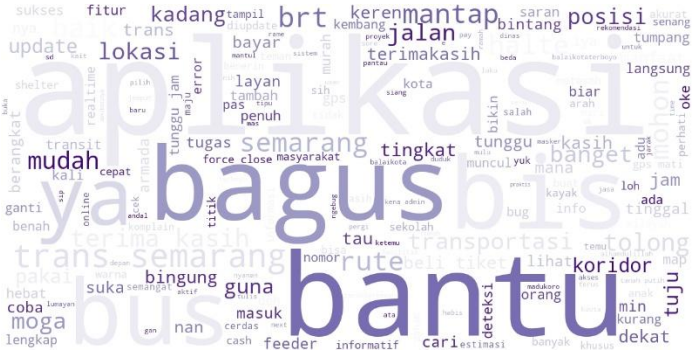
Gambar 4. 26 *Visualisasi Wordcloud Seluruh Ulasan*

2) Wordcloud Sentimen Positif

Dalam visualisasi *wordcloud* sentimen positif, untuk menampilkan gambaran dari *wordcloud* positif dilakukan dengan cara memfilter data ulasan yang memiliki label positif yang diambil dari kolom "*text steamindo*".

Pada visualisasi *wordcloud* untuk sentimen positif, dimana kata-kata yang muncul dengan ukuran lebih besar, seperti kata "aplikasi", "bagus", "bantu", "ya", dan lainnya menunjukkan bahwa kata tersebut paling sering digunakan oleh pengguna dalam ulasan positif terhadap aplikasi Trans Semarang. Contoh kemunculan kata "aplikasi", “bantu” dan "bagus" menunjukkan

adanya kepuasan pengguna terhadap kualitas layanan aplikasi, baik dari kemudahan penggunaan maupun fungsionalitasnya. Tampilan hasil *wordcloud* sentimen positif ditunjukkan pada gambar 4. 27 berikut.



Gambar 4. 27 *Visualisasi Wordcloud Sentimen Positif*

3) Wordcloud Sentimen Negatif

Dalam visualisasi *wordcloud* sentimen negatif, untuk menampilkan gambaran dari *wordcloud* negatif dilakukan dengan menyaring data ulasan pada label negatif yang diambil dari kolom "*text steamindo*". Tampilan visualisasi *wordcloud* pada sentimen negatif, kata yang ditampilkan dengan ukuran lebih besar seperti kata "aplikasi", "bus", "halte", "bug", "tunggu", dan lainnya, merupakan bahwa kata tersebut

paling sering digunakan oleh pengguna saat memberikan ulasan bernada negatif terhadap aplikasi.

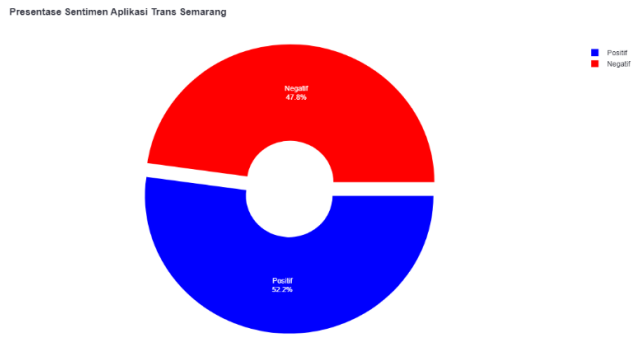
Kemunculan kata "aplikasi" menunjukkan bahwa masih terdapat keluhan atau ketidakpuasan pengguna terhadap performa maupun fitur dari aplikasi ini. Sedangkan kata seperti "bis", "halte" mengindikasikan adanya masalah atau kendala dalam layanan transportasi, seperti keterlambatan, ketidaksesuaian jadwal, atau lokasi halte yang kurang akurat. Kata “bug” berarti bahwa aplikasi masih sering mengalami kendala bug dan eror.

Dengan demikian, visualisasi wordcloud untuk sentimen negatif ini dapat memberikan wawasan mengenai aspek yang dianggap kurang memuaskan oleh pengguna. Informasi ini sangat penting bagi pengembang untuk mengevaluasi serta meningkatkan layanan aplikasi agar penilaian pengguna dapat meningkat lebih baik. Hasilnya ditunjukkan pada gambar 4.28 berikut.

Gambar 4. 28 *Visualisasi Wordcloud Sentimen Negatif*

E. Tampilan Hasil Visualisasi Diagram Lingkaran

Hasil visualisasi juga disajikan dalam bentuk diagram lingkaran yang menggambarkan presentase data dari masing-masing sentimen positif dan negatif. Tampilan hasil visualisasi diagram batang ditunjukkan pada gambar 4.29 berikut.

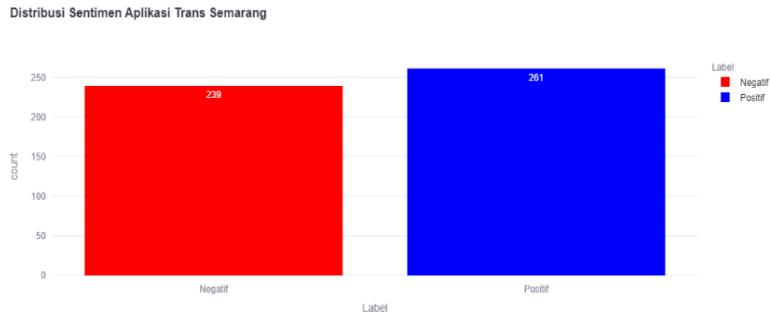


Gambar 4. 29 *Tampilan Visualisasi Diagram Lingkaran*

Berdasarkan diagram tersebut, bahwa presentase untuk sentimen positif memiliki presentase tertinggi sebesar 52,2%, dan sentimen negatif sebesar 47,8%. Dimana dapat disimpulkan bahwa sentimen positif lebih mendominasi persepsi pengguna terhadap aplikasi Trans Semarang. Sebanyak 52,2% pengguna memberikan ulasan dengan sentimen positif, sementara untuk 47,8% sisanya menandakan adanya persepsi kurang baik dari sebagian pengguna terhadap aplikasi.

F. Tampilan Halaman Hasil Visualisasi Diagram Batang

Hasil visualisasi yang ditampilkan dalam bentuk diagram batang menggambarkan distribusi jumlah sentimen positif dan negatif berdasarkan jumlah label sentimen dari setiap kategori sentimen setelah melalui beberapa tahap *preprocessing*. Hasilnya menunjukkan bahwa sentimen positif lebih mendominasi dengan 261 data dan label sentimen negatif hanya 239 data. Hal ini menunjukkan bahwa tanggapan pengguna terhadap ulasan aplikasi Trans Semarang cenderung bersifat positif. Tampilan hasil visualisasi diagram batang ditunjukkan pada gambar 4.30 berikut.



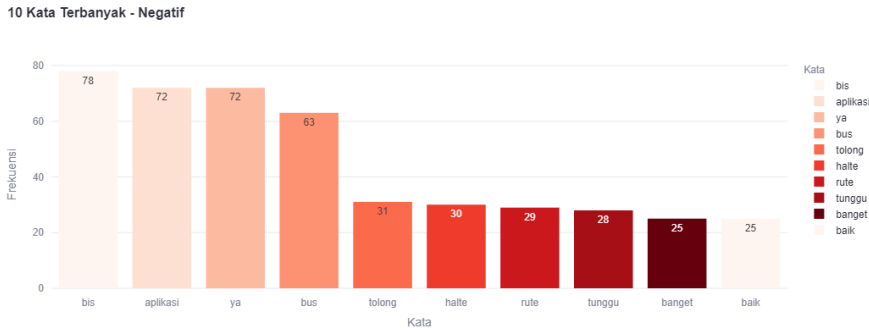
Gambar 4. 30 *Tampilan Visualisasi Diagram Batang*

G. Tampilan Visualisasi Wordcount Frekuensi 10 Kata

Tampilan halaman pada visualisasi *wordcount* menampilkan 10 jumlah kata teratas berdasarkan frekuensi nilai kemunculannya pada setiap sentimen. *Wordcount* ini merepresentasikan seberapa sering nilai pada suatu kata muncul, yang sebelumnya telah divisualisasikan dalam bentuk *wordcloud*.

a. Wordcount 10 Kata Terbanyak – Negatif

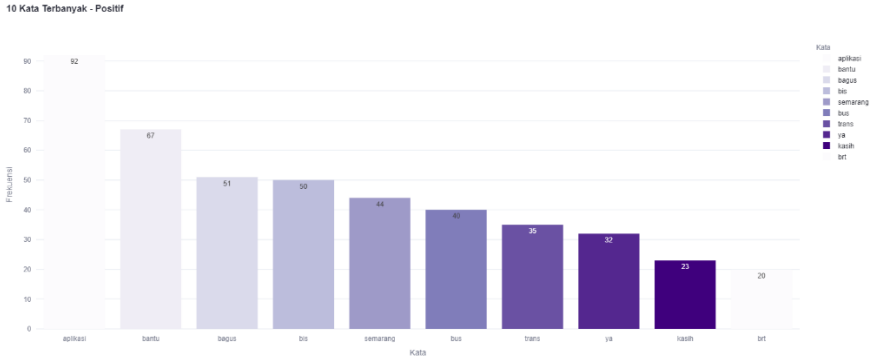
Hasil visualisasi *wordcount* untuk sentimen negatif menampilkan 10 kata dengan frekuensi kemunculan tertinggi dimana pada kata "aplikasi" mempunyai nilai frekuensi kata sebanyak 72, kata "bis" sebanyak 78, kata "bus" sebanyak 63, kata "halte" sebanyak 30, dan frekuensi kata lainnya. Tampilan hasil visualisasi *wordcount* negatif ditunjukkan gambar 4.31 berikut.



Gambar 4. 31 *Tampilan Visualisasi Wordcount Negatif*

b. Wordcount 10 Kata Terbanyak – Positif

Hasil visualisasi *wordcount* untuk sentimen positif juga menunjukkan 10 jumlah frekuensi kata tertinggi pada sentimen positif dimana pada kata "aplikasi" mempunyai nilai frekuensi kata sebanyak 92, kata "bantu" sebanyak 67, kata "bagus" sebanyak 51 kata "bis" sebanyak 50 dan kata lainnya dengan frekuensi yang lebih rendah. Tampilan hasil visualisasi *wordcount* positif ditunjukkan gambar 4.32 berikut.



Gambar 4. 32 *Tampilan Visualisasi Wordcount Positif*

BAB V
KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil pembahasan dan penelitian yang telah dijelaskan pada bab sebelumnya dapat simpulkan hal-hal sebagai berikut:

1. Penerapan metode *Naïve Bayes Classifier* dalam mengklasifikasikan sentimen persepsi pengguna terhadap aplikasi Trans Semarang menunjukkan hasil klasifikasi yang baik. Penelitian ini diawali dengan proses *scrapping* lalu labeling data, dilanjutkan dengan tahap *preprocessing* data, serta pemberian bobot kata menggunakan metode TFIDF. Selanjutnya, dilakukan proses klasifikasi menggunakan *Naïve Bayes Classifier*, yang kemudian diklasifikasi dan dievaluasi untuk menghasilkan dua kategori sentimen, positif dan negatif. Sebanyak 500 data ulasan teks yang sudah diberikan label berhasil dikumpulkan, dan setelah melalui tahap *preprocessing* jumlah data tetap dengan 261 ulasan sentimen positif dan 239 sentimen negatif. Masing-masing sentimen memiliki nilai presentase sebesar 52,2% untuk sentimen positif dan 47,8% untuk sentimen

negatif.

2. Algoritma Naïve Bayes Classifier dalam mengklasifikasikan sentimen persepsi pengguna aplikasi Trans Semarang menghasilkan klasifikasi dengan hasil akurasi yang baik. Hasil performa pada model Naïve Bayes Classifier menunjukkan nilai yang baik dengan nilai akurasi sebesar 82%, *precission* sebesar 81%, *recall* sebesar 82%, dan *f1-score* sebesar 81%.

B. Saran

Berdasarkan hasil penelitian yang telah dilaksankan, penulis menyadari masih terdapat sejumlah keterbatasan. Oleh karena itu, beberapa saran diajukan untuk penelitian selanjutnya agar dapat dikembangkan lebih optimal:

1. Menggunakan penerapan model algoritma untuk klasifikasi yang berbeda seperti *KNN*, *Support Vector Machine* (SVM), *Decission Tree* atau algoritma yang lainnya guna memperoleh performa yang lebih baik serta sebagai bahan perbandingan untuk menentukan model klasifikasi yang paling baik.
2. Dalam penelitian menggunakan teknik pelabelan manual, yang kemungkinan masih mengandung

ketidaktepatan dalam penentuan label sentimen. Oleh karena itu, pada penelitian selanjutnya dapat menggunakan teknik pelabelan otomatis seperti *vader lexicon* atau *textblob* untuk memperoleh hasil yang lebih akurat dan meningkatkan performa model.

3. Dalam penelitian, tampilan sistem *web* dikembangkan menggunakan *framework streamlit* dengan tampilan yang masih sederhana. Untuk penelitian selanjutnya, disarankan agar tampilan sistem dapat dikembangkan lebih lanjut dengan menggunakan *framework Python* lain atau dengan mengimplementasikan sistem *web* berbasis PHP untuk tampilan yang lebih interaktif, serta dapat mengembangkan sistem *web* ini dengan menambahkan beberapa proses klasifikasi sentimen lainnya yang dilakukan ke dalam sistem web.

DAFTAR PUSTAKA

Adela, C. N., Karnila, S., Sutedi, S., & Agarina, M. (2024). Analisis Ulasan Pengguna Aplikasi Seabank Dengan Support Vector Machine Dan Naïve Bayes. *J. Tekno Kompak*, *18*(2), 441.

Afifi, W. (2022). Analisis sentimen pengguna twitter terhadap layanan internet pt indosat tbk dengan metode k-nearest neighbor (k-nn) dan naive bayes classifier (nbc). *Repository.Uinjkt.Ac.Id*.

Agusia, P., Uli, M., Manurung, A., Calista, V., & Mawardi, V. C. (2024). *Pemanfaatan Word Cloud Pada Analisis Sentimen Dalam Menggali Persepsi Publik*. 25–30.

Agustiranti, T., Izzati Kurdiana, A. K., Al Ghiffari, B., Dwi Juniar, E., & Gita Purnama, D. (2024). Penerapan Naive Bayes Terhadap Sentimen Analisis Media Sosial Twitter Pengguna Kereta Cepat Jakarta-Bandung (Whoosh). *Jurnal Ilmu Komputer Dan Sistem Informasi (JIKOMSI)*, *7*(1), 297–305. https://doi.org/10.55338/jikomsi.v7i1.2946

Akbar, M. A., & Solichin, A. (2024). *Perbandingan Sentimen Ulasan Pengguna Aplikasi Ride-Hailing Gojek dan Grab Menggunakan Algoritma Multinomial Naïve Bayes Sentiment Comparison of User Reviews of Ride-Hailing Apps Gojek and Grab Using Multinomial Naïve Bayes Algorithm*. *4*, 1–11.

Akbar, M. N., Hasanahlmar'iyah Rusydi, N., Hasrul, M., & Ramadhanti, S. (2022). Sentiment Analysis Terhadap Review Aplikasi Maxim di Google Play Store Menggunakan Support Vector Machine (SVM). *JOURNAL of AGENTS*, *2*(2), 1.

Akbar, M. N., & Nirwana Samrin. (2023). Analisis Sentimen Komentar Pengguna Aplikasi Threads Pada Google Playstore Menggunakan Algoritma Multinominal Naive Bayes Classfier. *AGENTS: Journal of Artificial Intelligence and Data Science*, *3*(2), 21–29. https://doi.org/10.24252/jagti.v3i2.67

Ardiana, P. P., & Erawan, L. (2019). Aplikasi BRT Trans Semarang Berbasis Android. *JOINS (Journal of Information System)*, *4*(2), 190–202. https://doi.org/10.33633/joins.v4i2.2625

Astuti, K. C., Firmansyah, A., & Riyadi, A. (2024). Implementasi Text Mining Untuk Analisis Sentimen Masyarakat Terhadap Ulasan Aplikasi Digital Korlantas Polri pada Google Play Store. *REMIK: Riset Dan E-Jurnal Manajemen Informatika Komputer*, *8*(1), 383–394.

Atimi, R. L., & Enda Esyudha Pratama. (2022). Implementasi Model Klasifikasi Sentimen Pada Review Produk Lazada Indonesia. *Jurnal Sains Dan Informatika*, *8*(1), 88–96. https://doi.org/10.34128/jsi.v8i1.419

Barupal, D. K., & Fiehn, O. (2019). Generating the blood exposome database using a comprehensive text mining and database fusion approach. *Environmental Health Perspectives*, *127*(9), 2825–2830. https://doi.org/10.1289/EHP4713

Chairunnisa, C., Ernawati, I., & Santoni, M. M. (2022). Klasifikasi Sentimen Ulasan Pengguna Aplikasi PeduliLindungi di Google Play Menggunakan Algoritma Support Vector Machine dengan Seleksi Fitur Chi-Square. *Informatik : Jurnal Ilmu Komputer*, *18*(1), 69. https://doi.org/10.52958/iftk.v17i4.4594

Chintia, P. D. (2018). *INOVASI PELAYANAN TRANSPORTASI PUBLIK BRT (BUS RAPID TRANSIT) TRANS SEMARANG OLEH DINAS PERHUBUNGAN KOTA SEMARANG*. Faculty of Social and Political Sciences.

Ernianti Hasibuan, & Elmo Allistair Heriyanto. (2022). Analisis Sentimen Pada Ulasan Aplikasi Amazon Shopping Di Google Play Store Menggunakan Naive Bayes Classifier. *Jurnal Teknik Dan Science*, *1*(3), 13–24. https://doi.org/10.56127/jts.v1i3.434

Fachreza, R., & Handoko, W. T. (2024). Analisis Sentimen Masyarakat Terhadap Kinerja Trans Semarang Menggunakan Metode Random Forest. *Progresif: Jurnal Ilmiah Komputer*, *20*(2), 724–734.

Fatmawati, F., Irawan, B., & Bahtiar, A. (2024). Analisis Sentimen Pengguna Aplikasi Shejek Berdasarkan Ulasan Di Google Play Store Menggunakan Metode Naïve Bayes. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *8*(3), 2976–2981. https://doi.org/10.36040/jati.v8i3.9607

Febriyanti, A. (2020). Analisis Sentimen Persepsi Pengguna JNE Menggunakan Algoritma Naive Bayes Classifier. *Skripsi*, *16522259*, 129.

Giovani, A. P., Ardiansyah, A., Haryanti, T., Kurniawati, L., & Gata, W. (2020). Analisis Sentimen Aplikasi Ruang Guru Di Twitter Menggunakan Algoritma Klasifikasi. *Jurnal Teknoinfo*, *14*(2), 115. https://doi.org/10.33365/jti.v14i2.679

Hakim, S. N. (2021). *Analisis Sentimen Persepsi Pengguna Myindihome Menggunakan Metode Support Vector Machine (Svm) Dan Naïve Bayes Classifier (Nbc)*.

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

Harahap, Eva Darwisah Kurniawan, R. (2024). Analisis Sentimen Komentar Terhadap Kebijakan Pemerintah Mengenai Tabungan Perumahan Rakyat (TAPERA) Pada Aplikasi X Menggunakan Metode Naïve Bayes. *Jurnal Teknik Informatika Unika ST. Thomas (JTIUST)*, *9*(1), 2657–1501. https://ejournal.ust.ac.id/index.php/JTIUST/article/view/3911

Hasanah, A. N., & Sari, B. N. (2024). Analisis Sentimen Ulasan Pengguna Aplikasi Jasa Ojek Online Maxim Pada Google Play Dengan Metode Naïve Bayes Classifier. *Jurnal Informatika Dan Teknik Elektro Terapan*, *12*(1), 90–96. https://doi.org/10.23960/jitet.v12i1.3628

Indransyah, R., Chrisnanto, Y. H., Sabrina, P. N., & Kom, S. (2022). Klasifikasi Sentimen Pergelaran MotoGP di Indonesia Menggunakan Algoritma Correlated Naive

Bayes Clasifier. *INFOTECH Journal*, *8*(2), 60–66. https://doi.org/10.31949/infotech.v8I2.3103

Irsyad, A., & Geralda, R. D. (2023). Analisis Sentimen SEA Games 2023 di Twitter Metode dengan Machine Learning. *Adopsi Teknologi Dan Sistem Informasi (ATASI)*, *2*(2), 126–131. https://doi.org/10.30872/atasi.v2i2.1138

Jiana, N. A., & Hartono, B. (2024). Sentimen Analisa Ulasan Aplikasi Access by KAI pada Google Play Store menggunakan Algoritma K-NN. *Jurnal Media Informatika Budidarma*, *8*(3), 1388. https://doi.org/10.30865/mib.v8i3.7730

Junianto, H., Arsi, P., Kusuma, B. A., & Saputra, D. I. S. (2024). Evaluasi Aplikasi Raileo Melalui Analisis Sentimen Ulasan Playstore Dengan Metode Naive Bayes. *SINTECH (Science and Information Technology) Journal*, *7*(1), 27–40. https://doi.org/10.31598/sintechjournal.v7i1.1505

Kevin, K., Enjeli, M., & Wijaya, A. (2024). Analisis Sentimen Pengunaaan Aplikasi Kinemaster Menggunakan Metode Naive Bayes. *Jurnal Ilmiah Computer Science*, *2*(2), 89–98. https://doi.org/10.58602/jics.v2i2.24

LING, J., N. KENCANA, I. P. E., & OKA, T. B. (2014). Analisis Sentimen Menggunakan Metode Naïve Bayes Classifier Dengan Seleksi Fitur Chi Square. *E-Jurnal Matematika*, *3*(3), 92. https://doi.org/10.24843/mtk.2014.v03.i03.p070

Mahfudh, A. A., & Mustofa, H. (2019). Klasifikasi Pemahaman Santri Dalam Pembelajaran Kitab Kuning Menggunakan Algoritma Naive Bayes Berbasis Forward Selection. *Walisongo Journal of Information Technology*, *1*(2), 101. https://doi.org/10.21580/wjit.2019.1.2.4529

Marisa, F., Maukar, A. L., & Akhriza, T. M. (2021). *Data mining konsep dan penerapannya*. Deepublish.

Muhammadiyah, U., Bungo, M., Sakit, R., Lamongan, M., & Dan, M. T. (2024). *Jurnal Informatika Medis (J-INFORMED)*. *2*(1), 27–32.

Mustofa, H., & Mahfudh, A. A. (2019). Klasifikasi Berita Hoax Dengan Menggunakan Metode Naive Bayes. *Walisongo*

Journal of Information Technology, *1*(1), 1. https://doi.org/10.21580/wjit.2019.1.1.3915

Muttaqin, M. N., & Kharisudin, I. (2021). Analisis Sentimen Pada Ulasan Aplikasi Gojek Menggunakan Metode Support Vector Machine dan K Nearest Neighbor. *UNNES Journal of Mathematics*, *10*(2), 22–27. http://journal.unnes.ac.id/sju/index.php/ujm

Nugraha, D., & Gustian, D. (2023). Analisis Sentimen Penggunaan Aplikasi Transportasi Online Pada Ulasan Google Play Store Menggunakan Algoritma Svm (Support Vector Machine). *SISMATIK (Seminar Nasional Sistem Informasi Dan Manajemen Informatika)*, *1*(1), 326–335.

Nursalim Nursalim, & Agus Windu Sancono. (2023). Kualitas Pelayanan Bus Rapid Transit Trans Semarang. *MIMBAR ADMINISTRASI FISIP UNTAG Semarang*, *20*(1), 253–260. https://doi.org/10.56444/mia.v20i1.679

Pratmanto, D., Imaniawan, F. F. D., & Maarif, V. (2023). Analisis Sentimen Pada Ulasan Pengguna Aplikasi Identitas Kependudukan Digital Dengan Metode Naive Bayes Dan K-Nearest. *Computatio : Journal of Computer Science and Information Systems*, *7*(2), 155–166. https://doi.org/10.24912/computatio.v7i2.26322

Rahmadhani, A. Y. (2021). Analisis Sentimen Masyarakat Terhadap Program Prakerja Pada Sosial Media Twitter Menggunakan Metode Support Vector Machine. *Gastronomía Ecuatoriana y Turismo Local.*, *1*(69), 5–24.

Ramadhani, B., & Suryono, R. R. (2024). Komparasi Algoritma Naïve Bayes dan Logistic Regression Untuk Analisis Sentimen Metaverse. *Jurnal Media Informatika Budidarma*, *8*(2), 714. https://doi.org/10.30865/mib.v8i2.7458

Rieuwpassa, J. A., Sugito, S., & Widiharih, T. (2024). Implementasi Metode Naive Bayes Classifier Untuk Klasifikasi Sentimen Ulasan Pengguna Aplikasi Netflix Pada Google Play. *Jurnal Gaussian*, *12*(3), 362–371. https://doi.org/10.14710/j.gauss.12.3.362-371

Rish, I. (2001). An empirical study of the naive Bayes classifier.

IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence, *3*(22), 41–46.

Santoso, D. P., & Wibowo, W. (2022). Analisis Sentimen Ulasan Aplikasi Buzzbreak Menggunakan Metode Naïve Bayes Classifier pada Situs Google Play Store. *Jurnal Sains Dan Seni ITS*, *11*(2). https://doi.org/10.12962/j23373520.v11i2.72534

Soraya, F. A., & Indriyanti, A. D. (2024). *Perbandingan Algoritma Naïve Bayes dan Support Vector Machine dalam Analisis Sentimen Aplikasi Teman Bus*. *05*(02), 118–125.

Syam, A., Natsir, M. S., Muhtamar, S., & Azzahra, A. N. F. (2023). *Analisis Sentimen Menggunakan Algoritma Naïve Bayes Classifier Studi Kasus: Aplikasi Teman Bus Di Play Store*. *XII*(2), 74–83.

Tiara, L., Syaputra, H., Cholil, W., & Mirza, A. H. (2021). Web Scraper Dan Graphql API Untuk Data Perguruan Tinggi Di Indonesia Berdasarkan Website Kementerian Ristekdikti (Studi Kasus: Website Kementerian Ristekdikti). *Jurnal Nasional Ilmu Komputer*, *2*(3), 193–212. https://doi.org/10.47747/jurnalnik.v2i3.533

Utami, N. W., Purnama, I. N., & Prayoga, I. M. A. (2024). *Sentiment Analysis System Of Bali Tourism Using Naive Bayes Algorithm And Web Framework*. *14*(03), 275–282. https://doi.org/10.54209/infosains.v14i03

Verawati, I., & Jaelani, S. N. (2024). Analisis Sentimen Pengguna Twitter Terhadap Bus Listrik Menggunakan Naïve Bayes. *Jurnal Media Informatika Budidarma*, *8*(2), 832. https://doi.org/10.30865/mib.v8i2.7030

Yolanda, K., Yusra, Y., & Fikry, M. (2023). Klasifikasi Sentimen Ulasan Aplikasi WhatsApp di Play Store Menggunakan Naive Bayes Classifier. *J-Intech*, *11*(1), 1–9. https://doi.org/10.32664/j-intech.v11i1.867

Yuniarti, W. D., Faiz, A. N., & Setiawan, B. (2020). Identifikasi Potensi Keberhasilan Studi Menggunakan Naïve Bayes Classifier. *Walisongo Journal of Information Technology*, *2*(1), 1. https://doi.org/10.21580/wjit.2020.2.1.5204

Zanini, N., & Dhawan, V. (2015). Text Mining: An introduction

to theory and someapplications. *Research Matters: A Cambridge Assessment Publication*, *19*, 38–44.
Zhao, B. (2020). Encyclopedia of Big Data. *Encyclopedia of Big Data*, *May 2017*. https://doi.org/10.1007/978-3-319-32001-4

DAFTAR LAMPIRAN

LAMPIRAN 1: Lembar Pengesahan Proposal

KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI WALISONGO
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Prof. Dr. Hamka Ngaliyan Semarang
Telp.024-7601295Fax.7615387

LEMBAR PENGESAHAN

Naskah proposal skripsi berikut ini:
Judul : **Klasifikasi Sentimen Persepsi Pengguna Aplikasi Trans Semarang Menggunakan Naïve Bayes Classifier Berbasis Data Ulasan Online**
Penulis : **Ilma Salsabila**
NIM : 2108096029
Jurusan : Teknologi Informasi

Telah diujikan dalam seminar proposal skripsi oleh Dewan Penguji Fakultas Sains dan Teknologi UIN Walisongo Semarang pada Rabu, 08 Januari 2025.

Semarang, 21 Januari 2025

DEWANPENGUJI

Penguji I,
Dr. Khotibul Umam, M.Kom
NIP. 197908272011011007

Penguji II,
Dr. Wenty Dwi Yuniarti, S.Pd.,M.Kom
NIP. 197706222006042005

Penguji III,
Siti Nuraini, M.Kom
NIP. 198401312018012001

Penguji IV,
Nur Cahyo Hendro Wibowo, S.T.,M.Kom
NIP. 197312222006041001

LAMPIRAN 2: Contoh Data Hasil Scrapping Google Play Store

No	Username	Content
1	Mutiara Putriningsih	Tampilan antarmukanya agak kaku dan kurang. Mungkin bisa dibuat lebih user-friendly agar lebih menarik digunakan.
2	Saiful Anwar	Aplikasinya sangat informatif dan mudah digunakan, memudahkan pengguna menemukan transportasi umum dengan cepat dan tepat.
3	Mutiara	Apkikasinya sangat membantu banget loh bermanfaat apalagi buat anak sekolah yg biasa naik transportasi gini
4	Master PB	Aplikasi ini sangat memudahkan saya dalam mencari rute bus. Informasinya lengkap dan jelas
5	Mhammad Raditya Al	Aplikasinya ini bikin naik bus Trans Semarang jadi lebih praktis. Tidak perlu bingung lagi harus nunggu dimana
6	Gladys	Update informasi rute dan jadwal busnya kurang cepat. Kadang sudah ada perubahan di lapangan tapi belum tercermin di aplikasi
7	Cia	Aplikasi ini sangat lambat dan sering stuck di loading. Harusnya bisa jadi solusi, malah jadi bikin repot
8	Alaza2904	Cocok untuk pengguna harian Trans Semarang. Sangat membantu sekali, Lanjutkan inovasinya!
	…..	
497	Pengguna Google	Amazing apk
498	Pengguna Google	mantapppp
499	Pengguna Google	mantab banget nih aplikasi

LAMPIRAN 3: Contoh Data Hasil Pelabelan Manual

No	Username	Content	Label
1	Mutiara Putriningsih	Tampilan antarmukanya agak kaku dan kurang. Mungkin bisa dibuat lebih user-friendly agar lebih menarik digunakan.	Negatif
2	Saiful Anwar	Aplikasinya sangat informatif dan mudah digunakan, memudahkan pengguna menemukan transportasi umum dengan cepat dan tepat.	Positif
3	Mutiara	Apkikasinya sangat membantu banget loh bermanfaat apalagi buat anak sekolah yg biasa naik transportasi gini	Positif
4	Master PB	Aplikasi ini sangat memudahkan saya dalam mencari rute bus. Informasinya lengkap dan jelas	Positif
5	Mhammad Raditya Al	Aplikasinya ini bikin naik bus Trans Semarang jadi lebih praktis. Tidak perlu bingung lagi harus nunggu dimana	Positif
6	Gladys	Update informasi rute dan jadwal busnya kurang cepat. Kadang sudah ada perubahan di lapangan tapi belum tercermin di aplikasi	Negatif
7	Cia	Aplikasi ini sangat lambat dan sering stuck di loading. Harusnya bisa jadi solusi, malah jadi bikin repot	Negatif
8	Alaza2904	Cocok untuk pengguna harian Trans Semarang. Sangat membantu sekali, Lanjutkan inovasinya!	Positif
	…..		
497	Pengguna Google	Amazing apk	Positif
498	Pengguna Google	mantapppp	Positif
499	Pengguna Google	mantab banget nih aplikasi	Positif

LAMPIRAN 4: Contoh Data Pendukung Tahap Normalisasi (Kamuskatabakunormalisasi.xlsx)

tidak_baku	**kata_baku**
jgn	jangan
skarang	sekarang
karna	karena
gimna	bagaimana
kpan	kapan
gk	tidak
yaa	ya
kereeen	keren
dlm	dalam
mngkin	mungkin
mksh	terima kasih
makasih	terima kasih
....	
makin	semakin
telat	terlambat
jos	mantap
aplikasiny	aplikasinya
mancet	macet
oprasional	operasional
tlng	tolong
bangetttttt	banget
trnyata	ternyata
ohh	oh
yg	yang
pdahal	padahal
yyaaa	ya
trs	terus
klo	kalau
pdahal	padahal

LAMPIRAN 5: Contoh Data Pendukung Tahap Normalisasi (FormalizationDict.txt)

donlot	unduh
dpt	dapat
dr	dari
dri	dari
drmana	darimana
drmn	darimana
drpd	daripada
drtd	dari tadi
…..	
ya	iya
yaudah	ya sudah
yg	yang
yng	yang
yo	iya
yoha	iya
yowes	ya sudah
yuk	ayo
yup	iya

LAMPIRAN 6: Contoh Data Pendukung Tahap Normalisasi (KBBA.txt)

```
dah     deh
dapet   dapat
deket   dekat
dengerin        dengarkan
deyh    deh
dg      dengan
dgn     dengan
dh      deh
skalian sekalian
skl     sekali
sklh    sekolah
skrg    sekarang
....

happy   bahagia
irit    hemat
tak     tidak
mantepmantap
trending        trend
karna   karena
keliatan        kelihatan
mskpn   meskipun
nunggumenunggu
rubahnn         rubah
mbaur   baur
trlalu  terlalu
setting atur
emank   memang
seting  akting
maybe   mungkin
next    lanjut
sebelom         sebelum
```

LAMPIRAN 7: Contoh Data Pendukung Tahap Normalisasi (Slangword.txt)

aq:aku
bklan:bakalan
bkn:bukan
blg:bilang
bnr:benar
bnyk:banyak
br:baru
dgn:dengan
dkt:dekat
dlu:dulu
dmn:dimana
dpt:dapat
dr:dari
drpd:daripada
g:tidak
ga:tidak
gaada:tidak ada
....

tp:tapi
tkt:takut
tpii:tapi
trs:terus
trus:terus
ttg:tentang
ttp:tetap
yg:yang
slh:salah
sll:selalu
sm:sama
smga:semoga

LAMPIRAN 8: Contoh Dokumen Hasil Testing Prediksi Klasifikasi Sentimen

No	Data Ulasan	Label Aktual	Label Prediksi
1	tampilan antarmukanya agak kaku dan kurang mungkin bisa dibuat lebih userfriendly agar lebih menarik digunakan	Negatif	Negatif
2	aplikasinya sangat informatif dan mudah digunakan memudahkan pengguna menemukan transportasi umum dengan cepat dan tepat	Positif	Positif
3	apkikasinya sangat membantu banget loh bermanfaat apalagi buat anak sekolah yg biasa naik transportasi gini	Positif	Positif
4	aplikasi ini sangat memudahkan saya dalam mencari rute bus informasinya lengkap dan jelas	Positif	Positif
5	aplikasinya ini bikin naik bus trans semarang jadi lebih praktis tidak perlu bingung lagi harus nunggu dimana	Positif	Positif
6	update informasi rute dan jadwal busnya kurang cepat kadang sudah ada perubahan di lapangan tapi belum tercermin di aplikasi	Negatif	Negatif
7	aplikasi ini sangat lambat dan sering stuck di loading harusnya bisa jadi solusi malah jadi bikin repot	Negatif	Negatif
8	cocok untuk pengguna harian trans semarang sangat membantu sekali lanjutkan inovasinya	Positif	Positif

LAMPIRAN 9: Daftar Riwayat Hidup

RIWAYAT HIDUP

A. Identitas Diri

1. Nama Lengkap : Ilma Salsabila
2. Tempat & Tgl. Lahir : Semarang, 07 September 2003
3. Alamat Rumah : Mangkang Wetan, Gang Kutuk RT 04 RW 06, Kec.Tugu, Kota Semarang
4. HP : 085712262259
5. E-mail : ilmasalsabila324@gmail.com

B. Riwayat Pendidikan

a. SDN Mangkang Wetan 01 Semarang
b. Sekolah Menengah Pertama (SMP) Negeri 28 Semarang
c. Sekolah Menengah Kejuruan (SMK) Texmaco Semarang

Semarang, 20 Juni 2025
Ilma Salsabila

NIM : 2108096029