ANALISIS SENTIMEN PADA ULASAN APLIKASI SHELL ASIA DI *GOOGLE PLAY STORE* MENGGUNAKAN METODE *NAIVE BAYES*

TUGAS AKHIR

Diajukan untuk Memenuhi Sebagaian Syarat Guna
Memperoleh Gelar Sarjana Program Sastra Satu (S-1)
Dalam Ilmu Teknologi Informasi

Diajukan Oleh:

DESMI SALSABILA

NIM: 2108096028

PROGRAM STUDI TEKNOLOGI INFORMASI

FAKULTAS DAINS DAN TEKNOLOGI

UNIVERSITAS ISLAM NEGERI WALISONGO SEMARANG

TAHUN 2025

PERNYATAAN KEASLIAN

Yang bertandatangan dibawah ini:

Nama	:	Desmi Salsabila
NIM	:	2108096028
Jurusan	:	Teknologi Informasi

Menyatakan bahwa skripsi yang berjudul:

Analisis Sentimen Pada Ulasan Aplikasi Shell Asia Di *Google Play Store* Menggunakan Metode *Naive Bayes*

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

Semarang, 23 Juni 2025

Pembuat Pernyataan

Desmi Salsabila

2108096028

LEMBAR PENGESAHAN

KEMENTRIAN AGAMA
UNIVERSITAS ISLAM NEGERI WALISONGO
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Prof Dr. Hamka kampus III Ngaliyan
Semarang Telp. 7601295 Fax.7615387

LEMBAR PENGESAHAN

Naskah skripsi berikut ini:

Judul	: ANALISIS SENTIMEN PADA ULASAN APLIKASI SHELL ASIA DI GOOGLE PLAY STORE MENGGUNAKAN METODE NAIVE BAYES
Penulis	: **Desmi Salsabila**
NIM	: 2108096028
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, 27 Juni 2025

DEWAN PENGUJI

Penguji I,

Dr. Khothibul Umam S.T., M. Kom
NIP. 198108122011011007

Penguji II,

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

Penguji III,

Nur Cahyo Hendro W. S.T., M.Kom.
NIP. 197312222006041003

Penguji IV,

Dr. Masy Ari Ulinuha, M.T
NIP. 198108122011011007

Pembimbing I,

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

Pembimbing II,

Mokhamad Iklil M., M.Kom.
NIP. 198808072019031010

NOTA PEMBIMBING

Semarang, 23/05/2025

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	:	ANALISIS SENTIMEN PADA ULASAN APLIKASI SHELL ASIA DI *GOOGLE PLAY STORE.* MENGGUNAKAN METODE NAIVE BAYES
Penulis	:	Desmi Salsabila
NIM	:	2108096028
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 PEMBIMBING

Semarang, 23/05/2025

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	:	ANALISIS SENTIMEN PADA ULASAN APLIKASI SHELL ASIA DI GOOGLE PLAY STORE MENGGUNAKAN METODE NAIVE BAYES
Penulis	:	Desmi Salsabila
NIM	:	2108096028
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 II,

Mokhamad Iklil Mustofa M.Kom
NIP. 198808072019031010

ABSTRAK

Perkembangan teknologi digital mendorong perusahaan untuk menyediakan layanan berbasis aplikasi, termasuk Shell Asia dalam mendukung konsumennya. Ulasan pengguna di *Google Play Store* menjadi salah satu sumber informasi penting untuk mengevaluasi kepuasan pelanggan. Penelitian ini bertujuan untuk melakukan analisis sentimen terhadap ulasan pengguna aplikasi Shell Asia dengan menggunakan metode *Naive Bayes*. Metode ini dipilih karena kemampuannya dalam mengklasifikasikan teks secara efisien berdasarkan probabilitas.

Data yang digunakan berupa ulasan dalam bahasa Indonesia yang diambil melalui proses *web scraping*. Setelah melalui tahap praproses teks seperti *cleaning, case folding, normalisasi, tokenisasi,* dan *stemming.* Setelah praposes dilanjutkan dengan split data, pembobotan TF-IDF, lalu uji model dengan menggunakan *Naive Bayes*. Data diklasifikasikan ke dalam dua kategori sentimen, yaitu positif dan negatif. Berdasarkan hasil performa, metode *Naive Bayes* terbukti sebagai algoritma yang akurat dengan menghasilkan nilai *accuracy* sebesar 91%, *precision* pada kelas positif sebesar 92% nilai kelas, negatif sebesar 90%. *Recall* pada kelas positif sebesar 95%, nilai kelas negatif sebesar 86%. *F1-score* pada kelas positif sebesar 93%, nilai pada kelas negatif sebesar 88%. Hasil penelitian ini diharapkan dapat membantu pihak Shell Asia dalam memahami persepsi pengguna dan meningkatkan kualitas layanan aplikasinya.

Kata kunci : Analisis Sentimen, *Naive Bayes*, Shell Asia, *Google Play Store*, Ulasan Pengguna.

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT atas segala rahmat, taufik, dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul "Analisis Sentimen pada Ulasan Shell Asia di *Google Play Store* dengan Menggunakan Metode *Naive Bayes*" sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer pada Program Studi Teknologi Informasi, Fakultas Sains dan Teknologi, Universitas Islam Negeri Walisongo Semarang.

Penyusunan skripsi ini tidak lepas dari bantuan, bimbingan, dan dukungan dari berbagai pihak. Oleh karena itu, penulis menyampaikan ucapan terima kasih dan penghargaan yang sebesar-besarnya kepada:

1. Dr. Khotibul Umam, S.T.,M.Kom., selaku Ketua Program Studi Teknologi Informasi.
2. Ibu Dr. Wenty Dwi Yuniarti S.Pd.,M.Kom., selaku Dosen Pembimbing I yang telah memberikan arahan, kritik, dan masukan dengan penuh kesabaran dan ketelitian dalam proses penyusunan skripsi ini.
3. Bapak Mokhamad Iklil Mustofa M.Kom., selaku Dosen Pembimbing II yang telah memberikan arahan, kritik, dan masukan dengan penuh kesabaran dan ketelitian dalam proses penyusunan skripsi ini.

4. Bapak/Ibu dosen dan staf pengajar Program Studi Teknologi Informasi UIN Walisongo Semarang yang telah memberikan ilmu dan bimbingan selama masa studi.
5. Kedua orang tua tercinta, Bapak Misno dan Ibu Sukinah terimakasih atas segala pengorbanan, doa, dan kasih sayang yang tak pernah putus selama menempuh pendidikan.
6. Kelima kaka saya yaitu Haryanti, Purwati, Purwanto, Sapto, dan Nurul serta adik Risky yang selalu memberikan semangat dan motivasi dalam penyusunan skripsi ini.
7. Dias Bayu terima kasih atas kesabaran, dukungan emosional pengertiannya selama masa penyusunan skripsi doa dan menjadi penyemangat di setiap proses.
8. Putri Nabila, Thami, Qyla, Zikrina, Bellin, dan Nanda yang telah memberikan semangat, perhatian, mendukung dan menemani selama perjalanan kuliah ini.
9. Teman-teman Daffa, Shauqi, Tria, Ass, dan Nisrina terima kasih atas diskusi produktif, berbagi referensi, dan saling mendukung dalam menghadapi tantangan akademik bersama.

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I
PENDAHULUAN

A. Latar Belakang

Era digital telah membawa transformasi besar dalam berbagai aspek kehidupan manusia, termasuk sektor energi dan bahan bakar. Perkembangan teknologi informasi dan komunikasi (TIK) telah mendorong perusahaan-perusahaan energi untuk mengadopsi inovasi digital guna meningkatkan layanan dan pengalaman pelanggan (Islamia et al., 2022). Di Indonesia, PT. Pertamina (Persero) dan Shell Indonesia adalah dua contoh perusahaan bahan bakar yang telah mengembangkan aplikasi digital untuk memudahkan transaksi dan meningkatkan loyalitas pelanggan.

Shell Indonesia menghadapi berbagai tantangan dalam mempertahankan eksistensinya di pasar bahan bakar yang kompetitif. Salah satu tantangan utama adalah penutupan beberapa SPBU, seperti di Medan pada 1 Juni 2024 (CNN Indonesia 2024, diakses 16 Januari 2025). Penutupan beberapa Stasiun Pengisian Bahan Bakar Umum (SPBU) di Medan telah menimbulkan reaksi beragam dari masyarakat pengguna layanan bahan bakar di wilayah tersebut. Penutupan ini tidak hanya berdampak pada aksesibilitas fisik penggunaan bahan bakar, tetapi

juga mempengaruhi persepsi dan kepuasan pengguna terhadap aplikasi pendukung seperti Shell Asia yang menyediakan informasi dan kemudahan layanan terkait SPBU. Reaksi pengguna seringkali disampaikan melalui ulasan di platform seperti *Google Play Store*, yang menjadi media utama bagi konsumen untuk berbagi pengalaman dan opini mereka terhadap aplikasi tersebut.

Namun, jumlah ulasan yang mencapai ribuan membuat analisis manual ulasan menjadi sangat sulit dan tidak efektif. Informasi penting terkait sentimen pengguna yang terkandung dalam ulasan tersebut terabaikan jika tidak diolah secara tepat. Sentimen positif dan negatif yang muncul pada ulasan dapat memberikan gambaran yang jelas mengenai tingkat kepuasan pengguna, kendala yang dialami, dan aspek aplikasi yang perlu ditingkatkan. Oleh karena itu, analisis terhadap ulasan pengguna menjadi langkah penting untuk memahami persepsi masyarakat terhadap aplikasi Shell Asia.

Analisis sentimen dengan metode *Naive Bayes* menjadi pilihan tepat dalam mengolah ulasan pengguna aplikasi Shell Asia. Pada metode ini dipilih karena kemampuannya dalam menangani data teks secara efisien, terutama dalam analisis sentimen yang melibatkan ulasan pengguna. *Naive Bayes* mengasumsikan bahwa pengaruh

nilai atribut terhadap kelas tertentu tidak tergantung pada kelas itu sendiri maupun dipengaruhi oleh atribut lainnya. Metode ini membuat asumsi independensi antar atribut dalam mempengaruhi kelas (Ndapamuri et al., 2023).

Analisis sentimen dengan Naive Bayes membantu mengolah data ulasan dengan lebih cepat dan objektif. Metode ini juga sesuai dengan nilai Islam yang mengajarkan kejujuran dan tanggung jawab dalam penelitian, karena bisa mengurangi penilaian yang subjektif dari peneliti.

Dengan pendekatan ini**,** hasil penelitian diharapkan mampu memberikan gambaran komprehensif tentang bagaimana aplikasi Shell Asia diterima oleh masyarakat, memberikan rekomendasi perbaikan yang sesuai, dan mendorong pengembangan fitur inovatif yang selaras dengan kebutuhan pengguna di masa mendatang. Seperti pada firman Allah Subhanahu Wa Ta'ala dalam QS. Al-Hujurat 49: Ayat 11 yang berbunyi:

يٰٓاَيُّهَا الَّذِيْنَ اٰمَنُوْا لَا يَسْخَرْ قَوْمٌ مِّنْ قَوْمٍ عَسٰٓى اَنْ يَّكُوْنُوْا خَيْرًا مِّنْهُمْ وَلَا نِسَاۤءٌ مِّنْ نِّسَاۤءٍ عَسٰٓى اَنْ يَّكُنَّ خَيْرًا مِّنْهُنَّۚ وَلَا تَلْمِزُوْٓا اَنْفُسَكُمْ وَلَا تَنَابَزُوْا بِالْاَلْقَابِۗ بِئْسَ الِاسْمُ الْفُسُوْقُ بَعْدَ الْاِيْمَانِۚ وَمَنْ لَّمْ يَتُبْ فَاُولٰۤىِٕكَ هُمُ الظّٰلِمُوْنَ ﴿١١﴾

“Wahai orang-orang yang beriman, janganlah suatu kaum

mengolok-olok kaum yang lain (karena) boleh jadi mereka (yang diolok-olokkan itu) lebih baik daripada mereka (yang mengolok-olok) dan jangan pula perempuan-perempuan (mengolok-olok) perempuan lain (karena) boleh jadi perempuan (yang diolok-olok itu) lebih baik daripada perempuan (yang mengolok-olok). Janganlah kamu saling mencela dan saling memanggil dengan julukan yang buruk. Seburuk-buruk panggilan adalah (panggilan) fasik setelah beriman. Siapa yang tidak bertobat, mereka itulah orang-orang zalim."

Ayat ini menekankan pentingnya saling menghormati dan menghargai pendapat orang lain. Prinsip ini memiliki implikasi penting dalam konteks analisis sentimen, khususnya dalam penelitian terhadap aplikasi Shell Asia di *Google Play Store*. Dengan menerapkan prinsip-prinsip etika dan moral, peneliti dapat berkontribusi pada pengembangan aplikasi yang lebih baik dan bermanfaat bagi masyarakat.

B. Identifikasi Masalah

Berdasarkan latar belakang yang telah diuraikan, penelitian ini mengidentifikasi permasalahan utama yaitu perlunya identifikasi ulasan pengguna aplikasi Shell Asia yang tidak terstruktur dengan cara memisahkannya ke dalam kategori sentimen positif dan negatif. Untuk mencapai tujuan tersebut, metode *Naive Bayes* dipilih

sebagai pendekatan analisis karena kesederhanaan, keefektifan, kemampuannya menghasilkan performa yang baik, serta kesesuaiannya dalam menangani data dalam skala besar.

C. Rumusan Masalah

Berdasarkan latar belakang yang sudah dijelaskan diatas, maka peneliti mengambil rumusan masalah sebagai berikut:

1. Bagaimana menerapkan metode *Naive Bayes* dalam membantu analisis sentimen ulasan pengguna aplikasi shell asia di *Google Play Store?*
2. Bagaimana performa metode *Naive Bayes* dalam analisis sentimen terhadap aplikasi Shell Asia di *Google Play Store*?

D. Tujuan Penelitian

Berdasarkan rumusan masalah yang telah ditetapkan, penelitian ini bertujuan untuk:

1. Mengimplementasi metode *Naive Bayes* dalam analisis sentimen ulasan pengguna aplikasi Shell Asia di *Google Play Store* kedalam setimen positif atau negatif.
2. Mengetahui performa yang dihasilkan oleh metode

Naive Bayes dalam mengkalisifikasi ulasan pengguna Shell Asia kedalam sentiment positif dan negatif.

E. Batasan Masalah

Untuk mencapai penelitian yang objektif dan terarah, peneliti membatasi ruang lingkupnya dengan menetapkan batasan masalah yaitu:

1. Data yang diambil adalah data dari ulasan pengguna atau komentar pada aplikasi Shell Asia yang ada di *Google Play Store*.
2. Klasifikasi pada analisis sentiment ini menggunakan metode *Naive bayes*
3. Ulasan pengguna pada aplikasi Shell Asia akan diklasisfikasi menjadi dua sentiment yaitu positif dan negatif.
4. Penelitian ini menggunakan bahasa pemograman *Python* dan juga menggunakan *software google colab*.
5. Data ulasan yang diambil dari aplikasi Shell Asia memeiliki rentan waktu 10 Oktober 2021-29 November 2024.
6. Visualisasi disajikan pada website sederhana menggunakan *framwork streamlit*.

F. Manfaat Penelitian

Manfaat penelitian ini terbagi menjadi dua kategori, yaitu manfaat teoritis dan manfaat praktis.

1. Manfaat Teoritis

Penelitian ini berkontribusi pada literatur yang ada tentang aplikasi *Naive Bayes* dalam analisis sentimen. Melalui penelitian ini, bisa didemonstrasikan bagaimana algoritma *Naive Bayes* dapat diadaptasi dan dioptimalkan untuk menganalisis data ulasan pengguna di platform digital. Ini juga bisa memperlihatkan kekuatan dan kelemahan model ini dalam konteks yang spesifik.

2. Manfaat Praktis

a. Analisis sentimen dapat memahami respon pengguna aplikasi. Yang dimana memanfatkan data digital pada komentar di *Google Play Store.* Baik yang bersifat komentar positif, dan negatif.

b. Membantu mengevaluasi kinerja pada aplikasi Shell Asia. Dari evaluasi tersebut agar bisa meningkatkan layanan aplikasi, untuk meningkatkan pengalaman pengguna.

c. Jika respon pengguna dari aplikasi positif bisa

menjadi pertimbangan untuk membuka dan memperluas jaringan SPBU Shell di tempat atau daerah lainya.

G. Sistematika Penelitian

Pada laporan penelitian ini terbagi menjadi beberapa bab, dan untuk membantu pembaca memahami alur pembahasan, sistematika penulisannya disusun sebagai beriku:

1. BAB I: PENDAHULUAN

   Bab ini mencakup beberapa elemen penting, yaitu latar belakang, rumusan masalah, tujuam penelitian, batasan masalah, manfaat penelitian, dan sistematika penulisan

2. BAB II: LANDASAN TEORI

   Bab landasan pustaka ini mengulas berbagai kajian pustaka yang mendukung penelitian ini, serta memaparkan teori-teori dasar terkait, khususnya mengenai analisis sentimen dengan menggunakan metode *Naive Bayes.*

3. BAB III: METODELOGI PENELITIAN

   Bab ini membahas metode yang digunakan oleh peneliti dalam memperoleh data. Selain itu, bab ini

juga menguraikan perangkat-perangkat yang digunakan dalam penelitian, serta menjelaskan alur pengerjaan penelitian dan memberikan gambaran umum mengenai metodologi yang diuraikan.

4. BAB IV: HASIL DAN PEMBAHASAN

   Bab ini memaparkan hasil penelitian dan pembahasannya, yang diharapkan dapat menjawab permasalahan yang telah dianalisis pada metodologi penelitian di Bab III.

5. BAB V: KESIMPULAN DAN SARAN

   Bab ini merangkum kesimpulan dari hasil penelitian yang disajikan di Bab IV secara singkat, jelas, dan sistematis. Selain itu, bab ini juga berisi saran-saran dari penulis untuk penelitian selanjutnya.

BAB II

LANDASAN PUSTAKA

A. Analisis sentimen

Menurut (Liu, 2016). analisis sentimen, atau yang dikenal sebagai penggalian opini adalah bidang ilmu yang memungkinkan kita untuk memahami opini, sentimen, penilaian, sikap, dan emosi seseorang terhadap suatu entitas. Entitas ini dapat berupa produk, layanan, organisasi, individu, acara, masalah, atau topik yang diungkapkan dalam teks tertulis.

Analisis sentimen menjadi alat yang sangat penting untuk melihat dan memahami opini publik. Kita dapat menggunakannya untuk menjelajahi berbagai platform online seperti *Blog, Twitter, Facebook*, dan lainnya untuk mengumpulkan dan menganalisis opini masyarakat terhadap suatu objek, entah itu produk, layanan, tokoh, atau isu tertentu. Tujuan utama analisis sentimen adalah untuk mengukur sentimen publik, apakah mayoritas berpendapat positif atau negatif terhadap objek tersebut (Mas Pintoko & Muslim L, 2018).

B. *Google Play Store.*

Google Play Store merupakan platform yang menyediakan berbagai kebutuhan digital pengguna

Android dan Web. Pengguna dapat menemukan aplikasi, game, musik, buku, film, hingga acara TV. Salah satu fitur penting *Google Play Store* adalah kolom ulasan. Pengguna dapat memberikan komentar dan penilaian terhadap aplikasi, game, atau konten digital lainnya. Ulasan ini menjadi sumber informasi yang berharga bagi pengguna lain dalam mengambil keputusan sebelum mengunduh atau membeli sesuatu (Luqman Adi, 2023).

Namun, peringkat ulasan di *Google Play Store.* yang hanya berkisar antara 1 hingga 5, sering kali tidak cukup menggambarkan kualitas aplikasi. Ulasan dalam bentuk kalimat lebih efektif dalam menunjukkan bagaimana tanggapan pengguna terhadap aplikasi, termasuk ulasan positif dan negatif. Ulasan ini pun dapat mempengaruhi keputusan pengunjung yang ingin mengunduh aplikasi (Sujjada et al., 2023).

C. Shell Asia

Shell Indonesia, anak perusahaan dari Royal Dutch Shell asal Belanda yang terdaftar di Inggris, merupakan perusahaan ritel bahan bakar yang berdiri sejak 2005. Perusahaan ini bergerak di bidang penjualan bahan bakar minyak (BBM) retail, pelumas untuk industri, otomotif, transportasi, bahan bakar kelautan, bahan bakar komersial, dan bitumen. Sejarah Shell Indonesia dimulai

dengan peresmian SPBU pertamanya di Karawaci, Tangerang (Maulidy & Suharto, 2024). Saat ini, di bawah kepemimpinan Ingrid Siburian sejak 1 Juli 2022, Shell Indonesia mengoperasikan 215 SPBU yang tersebar di lima provinsi: DKI Jakarta, Banten, Jawa Barat, Jawa Timur, dan Sumatera Utara (Shell Indonesia 2024, diakses 16 Januari 2025).

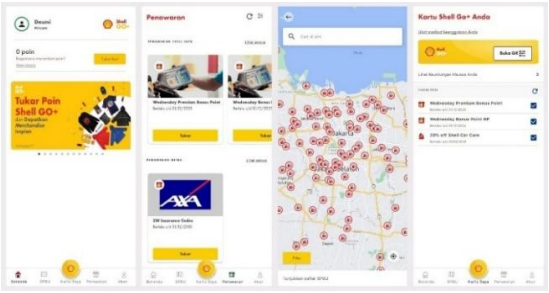
Shell Indonesia mengembangkan aplikasi bernama *Shell Go+* atau Shell Asia. Aplikasi Shell Asia adalah program loyalitas digital, yang dimana memiliki nama *Shell Go+*. Aplikasi memudahkan pelanggan dalam mengakses berbagai layanan, seperti lokasi SPBU, promo, dan riwayat pembelian dan juga asuransi. Program asuransi gratis berupa personal accident insurance dengan perlindungan lebih dari 100 juta rupiah, bekerja sama dengan *AXA Insurance* selama 1 tahun, diberikan kepada anggota yang rutin membeli produk oli bertipe 4W. Promo-promo dalam program *Shell Go+ Club* dapat diperoleh melalui penukaran poin yang secara otomatis dihitung berdasarkan total transaksi anggota (Maulidy & Suharto, 2024). Berikut gambar aplikasi dari Shell Asia disajikan pada gambar 2.1.

Shell Indonesia secara resmi meluncurkan program loyalitas dengan nama *Shell Go+* di aplikasi Shell Asia pada

hari Jumat, 4 Februari 2022 (Shell Indonesia 2022, diakses 16 Januari 2025).

Gambar 2. 1 Aplikasi Shell Asia

Shell Indonesia secara resmi meluncurkan program loyalitas dengan nama *Shell Go+* di aplikasi Shell Asia pada hari Jumat, 4 Februari 2022 (Shell Indonesia 2022, diakses 16 Januari 2025). Aplikasi Shell Asia gratis di *Google Play Store,* memungkinkan penggunanya untuk mengunduh dan merasakan berbagai fiturnya. Popularitasnya terlihat dari jumlah unduhan yang mencapai 5 juta pengguna, dengan rating rata-rata 4.9 yang terbilang tinggi. Lebih dari 41.000 ulasan pun telah diberikan, menjadikannya sumber data berharga untuk memahami persepsi pengguna terhadap aplikasi ini.



Gambar 2. 2 Tampilan Aplikasi

D. *Text mining*

Text mining adalah proses mengungkap informasi atau tren baru yang sebelumnya tersembunyi dengan memproses dan menganalisis data dalam jumlah besar. *Text mining* merupakan salah satu cabang dari ilmu data mining yang memfokuskan pada analisis data dalam bentuk dokumen teks (Rachman Chajannah et al., 2020). *Text mining*, juga dikenal sebagai text data mining atau pencarian pengetahuan dari basis data teks, merupakan suatu proses untuk mencari pengetahuan yang terutama berfokus pada data yang berupa dokumen atau teks, dengan tujuan untuk mengekstrak informasi bermanfaat dan mengidentifikasikannya. (Rivki & Mukharil Bachtiar, 2017).

Data yang diproses oleh *Text mining* adalah data dalam bentuk teks. Sebelum melakukan *Text mining*, ada tahap text preprocessing. Tahapan dari text preprocessing yaitu *case folding*, cleaning, *Tokenizing*, dan *stemming* (Yoga Pratama et al., 2021). *Text mining* adalah proses analisis data berbentuk teks yang berasal dari dokumen, seperti kalimat, kata, dan konsep. Biasanya, *Text mining* digunakan untuk mengklasifikasikan dokumen-dokumen tekstual, di mana dokumen-dokumen tersebut akan dikelompokkan berdasarkan topik yang sesuai (Darwis et

al., 2021).

E. *Python*

Python adalah bahasa pemrograman tingkat tinggi yang dikembangkan oleh Guido Van Rossum dan pertama kali dirilis pada tahun 1991. Bahasa ini semakin populer dalam beberapa tahun terakhir karena sifatnya yang serbaguna. *Python* dapat digunakan untuk berbagai keperluan, termasuk *Machine learning dan Deep Learning*. Pemilihan *Python* dalam penelitian sering kali didasarkan pada sintaksisnya yang sederhana dan mudah dipahami. Selain itu, *Python* memiliki koleksi pustaka yang sangat lengkap dan didukung oleh komunitas yang besar dan aktif, mengingat *Python* bersifat open source (Riziq sirfatullah Alfarizi et al., 2023).

Python digunakan untuk memproses dan membersihkan data ulasan dari *Google Play Store.*, sehingga memungkinkan klasifikasi opini menggunakan algoritma data mining. Selain itu, *Python* juga digunakan untuk menganalisis dan memvisualisasikan data ulasan secara efektif (Fauziyyah & Gautama, 2020).

Python dapat digunakan untuk melakukan klasifikasi menggunakan berbagai algoritma, termasuk *Naive Bayes* yang menjadi metode utama dalam penelitian ini. Penelitian ini memanfaatkan sejumlah pustaka

bawaan *Python*, seperti *Pandas, Numpy, Matplotlib, dan Seaborn*, serta beberapa pustaka tambahan yang perlu diinstal, seperti *Google ID Client Library for Python, Scikit-learn (sklearn), Jcopml (Journey to the Center of Python Machine learning), dan Sastrawi* (Ilayah, 2023) .

F. Scraping Data

*Web scraping*atau *web crawling* adalah proses mengambil informasi dari situs web menggunakan program komputer. Program ini bisa bekerja seperti manusia saat menjelajahi internet, misalnya dengan menggunakan peramban web seperti *Mozilla Firefox atau Internet Explorer.* Selain itu, program ini juga bisa langsung mengakses situs web melalui protokol HTTP, yang merupakan cara dasar komputer berbicara satu sama lain di internet. *Web scraping* adalah teknik untuk mengubah data dari web yang tidak teratur menjadi data yang teratur, sehingga bisa disimpan dan dianalisis dalam spreadsheet atau database. Dengan teknik ini, bot bisa mengumpulkan data dalam jumlah besar dengan cepat, yang sangat bermanfaat di zaman sekarang, terutama karena kita memiliki banyak data besar *(big data)* yang terus berubah dan diperbarui (Khder, 2021).

Teknik *web scraping* merupakan proses mengumpulkan dokumen semi-terstruktur dari internet,

biasanya dalam bentuk laman web yang dibangun dengan bahasa markup seperti HTML atau XHTML. Proses ini bertujuan mengekstrak informasi tertentu dari halaman web, baik secara keseluruhan maupun sebagian, untuk digunakan sesuai kebutuhan (Tripathy et al., 2015)

G. Pre-processing

Tahap awal dalam *Text mining*, yang mencakup seluruh rutinitas dan proses untuk menyiapkan data yang akan digunakan oleh sistem *Text mining*, dikenal sebagai tahap *pre-processing* teks (Feldman & Sanger, 2007). *Pre-processing* teks adalah tahap penting dalam *Text mining* untuk mengubah data teks yang tidak terstruktur menjadi data terstruktur. ra-pemrosesan teks merupakan tahapan krusial yang bertujuan untuk meningkatkan akurasi dalam proses klasifikasi data teks. Proses ini melibatkan normalisasi morfologis kata-kata berimbuhan melalui analisis afiksasi yang komprehensif. Dalam bahasa Indonesia, terdapat empat jenis utama pembentukan kata berimbuhan seperti, awalan, akhiran, gabungan awalan dan akhiran, serta sisipan. Tujuan akhir dari proses ini adalah untuk mereduksi semua varian kata berimbuhan kembali ke bentuk kata dasarnya sesuai dengan entri lema yang terdaftar dalam Kamus Besar Bahasa Indonesia (KBBI). (Putri Santoso & Wibowo, 2022). Tahapan *pre-*

processing adalah sebagai berikut:

1. *Cleaning data*

   Proses pembersihan, yang dikenal sebagai cleaning, bertujuan untuk menghapus atribut-atribut yang tidak memiliki dampak signifikan terhadap proses klasifikasi, seperti tanda baca, spasi kosong, dan emoji. Langkah ini penting untuk mempersiapkan data dengan lebih baik sebelum dilakukan analisis lebih lanjut.

2. *Case folding*

   Case folding adalah proses standarisasi karakter dalam data dengan mengonversi semua huruf menjadi huruf kecil *(lowercase).* Proses ini dilakukan untuk memudahkan pencarian kata dalam data.

3. *Normalization*

   Normalization adalah proses untuk memperbaiki kesalahan yang ada pada kata, seperti ejaan yang salah, agar kata yang memiliki makna sama menjadi setara. Tujuannya adalah untuk meningkatkan konsistensi dan keakuratan data teks sebelum dilakukan analisis lebih lanjut.

4. *Tokenizing*

Tokenizing adalah proses untuk membagi kalimat menjadi bagian-bagian kata atau token. Ini membantu dalam mempersiapkan teks untuk analisis lebih lanjut dengan mengidentifikasi unit-unit dasar dari teks yang akan dianalisis.

5. *Stemming*

Stemming adalah proses terakhir dari preprocessing di mana kata-kata dalam teks dikonversi menjadi bentuk kata dasar. Tujuannya adalah untuk mengurangi variasi kata yang memiliki akar kata yang sama, sehingga mempermudah analisis teks secara keseluruhan.

H. Split Validation Data

Split validation adalah operator yang menerapkan validasi sederhana dengan membagi (split) data set secara acak menjadi data set training dan data set testing untuk mengevaluasi model yang dihasilkan. *Split validation* digunakan untuk memvalidasi dan mengukur kinerja algoritma atau metode yang diterapkan Kinerja model biasanya diukur dengan menilai tingkat akurasinya dalam menjalankan fungsinya (Suryono et al., 2018). Proses split data dilakukan menggunakan *RapidMiner Studio*, yang melibatkan pembagian hasil dari tahap labeling menjadi dua bagian, yaitu data training dan data testing (Zain &

Kamayani, 2023). Pada penelitian ini data training menjadi 80% dan untuk data testing menjadi 20%.

I. Perhitungan TF-IDF

TF-IDF *(Term Frequency-Inverse Document Frequency)* adalah metode yang digunakan untuk mengubah data tekstual menjadi data numerik, dengan tujuan memberikan bobot pada setiap kata atau fitur. TF-IDF adalah ukuran statistik yang digunakan untuk menilai pentingnya suatu kata dalam sebuah dokumen (Andre Septian et al., 2019).

(TF) *Term Frequency* mengukur seberapa sering sebuah kata muncul dalam sebuah dokumen. Semakin sering sebuah kata muncul, semakin tinggi bobotnya, yang menunjukkan tingkat relevansi yang lebih besar. Sebaliknya, dalam konsep IDF, kata yang sering muncul di berbagai dokumen justru memiliki bobot yang rendah (Yutika et al., 2021). TF-IDF adalah salah satu cara untuk mengukur seberapa penting suatu kata dalam sebuah dokumen dengan mempertimbangkan dua aspek utama. Pertama, metode ini melihat seberapa sering kata tersebut muncul dalam dokumen tersebut TF (*Term Frequency*), di mana semakin tinggi frekuensinya, semakin besar kemungkinan kata itu penting. Kedua, metode ini juga memperhitungkan seberapa umum kata tersebut muncul

di seluruh kumpulan dokumen IDF (*Inverse Document Frequency*). TF-IDF bekerja dengan prinsip bahwa kata yang muncul di banyak dokumen merupakan kata umum yang memiliki kontribusi kecil dalam representasi dokumen. Sebaliknya, kata yang jarang muncul di dokumen lain dianggap lebih penting dalam membedakan dokumen tersebut.Dengan menggabungkan kedua faktor ini, TF-IDF memberikan bobot yang tinggi pada kata-kata yang sering muncul dalam dokumen tertentu tetapi jarang ditemukan di dokumen lain, sehingga membantu mengidentifikasi kata kunci yang benar-benar relevan dan unik untuk dokumen tersebut (Sujadi et al., 2022). Berikut Rumus TF-IDF:

$$TFIDF(d,t) = TF(d,t).IDF(t)$$

Dimana :

$$TF(d,t) = \frac{(Jumlah\ kemunculan\ kata\ \ (t)\ pada\ dokumen\ (d)\ )}{Jumlah\ kata\ dalam\ dokumen\ (d)}$$

$$IDF(t) = log\frac{Jumblah\ dokumen}{Jumlah\ dokumen\ yang\ mengandung\ kata\ t} + 1$$

Keterangam :

d = Dokumen

t = Kata

J. Klasifikasi

Klasifikasi adalah metode untuk mengelompokkan benda berdasarkan karakteristik yang dimiliki oleh objek tersebut. Proses klasifikasi bisa dilakukan dengan berbagai cara, baik secara manual maupun dengan bantuan teknologi. Klasifikasi manual dilakukan oleh manusia tanpa menggunakan algoritma cerdas dari komputer. Sebaliknya, klasifikasi yang dibantu oleh teknologi menggunakan beberapa algoritma, termasuk *Naive Bayes, Support Vector Machine, Decision Tree, Fuzzy, dan Jaringan Saraf Tiruan* (Wibawa et al., 2018).

Klasifikasi adalah salah satu tugas penting dalam data mining. Sebuah pengklasifikasi dibuat dari sekumpulan data latih yang sudah memiliki kelas yang ditentukan sebelumnya. Klasifikasi adalah proses mengelompokkan fitur-fitur ke dalam kelas yang sesuai. Vektor fitur pelatihan yang tersedia telah diketahui kelasnya, dan vektor-vektor ini digunakan untuk merancang pemilah. Proses pengenalan pola ini disebut terbimbing atau supervised(Wibawa et al., 2018).

K. Impelementasi Algoritma *Naive Bayes*

Naive Bayes adalah salah satu algoritma data mining yang sederhana dan mudah digunakan, dengan waktu pemrosesan yang cepat. Algoritma ini memiliki struktur yang sederhana sehingga mudah diimplementasikan,

serta dikenal memiliki efektivitas yang tinggi(Fitri et al., 2020).

Naive Bayes merupakan algoritma probabilistik yang sering digunakan untuk tugas klasifikasi. *Naive Bayes Classifier* bukanlah satu algoritma tunggal, melainkan sekumpulan algoritma klasifikasi yang semuanya didasarkan pada prinsip yang sama, yaitu Teorema Bayes(Ghojogh et al., 2019).

Algoritma *Naive Bayes Classifier* adalah algoritma yang digunakan untuk menentukan nilai probabilitas tertinggi dalam mengklasifikasikan data uji ke dalam kategori yang sesuai (Feldman & Sanger, 2007). Secara umum, rumus dasar dari persamaan teorema Bayes dapat dirumuskan sebagai berikut:

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

Keterangan:

X = Data dengan kelas yang belum diketahui

H = Hipotesis *X* merupakan suatu kelas spesifik

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

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

P(X|H) = Probabilitas *X* berdasarkan hipotesis *H*

P(X) = Probabilitas *X* (data sampel yang diamati)

L. Evaluasi Data

Tahap evaluasi adalah tahap yang dilakukan untuk melihat seberapa baik performansi algoritma klasifikasi yang digunakan dalam penelitian. Metode yang digunakan untuk evaluasi dalam penelitian ini adalah *confusion matrix. Confusion matrix* adalah tabel yang memberikan informasi perbandingan antara hasil klasifikasi yang diprediksi oleh sistem dengan hasil klasifikasi yang sebenarnya (Fikri et al., 2020). Contoh table *Confusion matrix* 2x2:

Tabel 2. 1 Confusion matrix Dua Kelas

		Predicted Class	
		Negative	*Positive*
True Class	*Negative*	*(TN) True Negative*	*(FP) False Positive*
	Positive	*(FN) False Negative*	*(TP) True Positive*

Keterangan :

True Positive (TP) adalah jumlah prediksi positif yang benar.

False Positive (FP) adalah jumlah prediksi positif yang salah.

True Negative (TN) adalah jumlah prediksi negatif yang benar.

False Negative (FN) adalah jumlah prediksi negatif yang salah.

Confusion matrix yang digunakan adalah matriks 2 x 2 untuk merangkum seluruh hasil klasifikasi benar dan salah. Empat variabel pengukuran dibuat dari kombinasi data *true positive (TP), false positive (FP), false negative (FN),* dan *true negative (TN),* berdasarkan hasil dari *confusion matrix*, dapat menghasilkan nilai *accuracy*, nilai *precision*, nilai *recall,* dan *f1-score* (Ndapamuri et al., 2023). Berikut perhitungan *accuracy,* nilai *precision,* nilai *recall,* dan *f1-score*:

1. *Accuracy*

   Tingkat akurasi klasifikasi model diukur menggunakan akurasi. Untuk mengevaluasi kinerja model, data uji harus disiapkan dengan cermat, atau dapat dipahami sebagai sejauh mana nilai yang diprediksi sesuai dengan nilai sebenarnya. Rumus untuk menghitung akurasi adalah sebagai berikut:

$$Akurasi = \frac{TP + TN}{TP + FP + FN + TN} * 100\%$$

2. *Precision*

Precision mengukur seberapa akurat hasil prediksi model, atau sejauh mana sistem merespon permintaan informasi pengguna dan menghasilkan hasil yang relevan. Rumus untuk menghitung *precision* adalah sebagai berikut:

$$Precision = \frac{TP}{TP + FP}$$

3. *Recall*

Recall menggambarkan seberapa sukses model dalam menemukan informasi yang relevan. Ini juga dapat disebut sebagai jumlah dokumen teks yang relevan yang berhasil diidentifikasi. Rumus untuk menghitung *recall* adalah sebagai berikut:

$$Recall = \frac{TP}{TP + FN}$$

4. *F1-score*

F1-score merupakan rata-rata harmonis dari *precision* dan *recall*. *F1-score* memberikan gambaran kinerja algoritma pada berbagai ukuran dataset dan menghasilkan analisis yang lebih akurat. Rumus untuk menghitung *F1-score* adalah sebagai berikut:

$$f1 - Scrore = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

M. Visualisasi

Pada tahap akhir, hasil analisis sentimen akan divisualisasikan menggunakan diagram batang dan *Wordcloud. Wordcloud* akan menampilkan hasil klasifikasi sentimen untuk mengidentifikasi topik yang paling sering dibahas oleh pengguna. Visualisasi ini bertujuan untuk mengekstrak informasi penting dari berbagai tanggapan yang ada, terutama terkait respons pengguna terhadap aplikasi Shell Asia di *Google Play Store*. Dengan demikian, informasi yang diperoleh dapat membantu memahami respons pengguna secara lebih efektif (Zidan, 2022).

Hasil dari visualisasi ini nantinya akan ditampilkan pada sebuah web sederhana yang dirancang untuk memudahkan pengguna dalam melihat dan memahami data secara lebih interaktif. Web ini akan menyajikan diagram, *Wordcloud,* dan prediksi sentimen sehingga pengguna dapat mengeksplorasi berbagai aspek sentimen dengan lebih jelas.

N. Kajian Terkait

Berikut detail dari rujukan penelitian ditunjukan pada tabel 2.2

Tabel 2. 2 Kajian Terkait

No	Topik	Penulis	Metode	Objek	Klasisfikasi
1	Sentimen pada	(Hasibuan	*Naive*	*Google*	Positif,

	ulasan aplikasi Amazon	& Heriyanto, 2022)	*Bayes*	*Play Store*	negaitf, dan netral
2	Sentimen terhadap mobil listrik	(Aryanti & Santoso, 2023)	*Naive Bayes*	Twitter	Positif, Negatif
3	Presepsi masyarakat terhadap pengguna aplikasi Mypertamina	(Maria et al., 2023)	*Naïve Bayes Classifier*	Twitter	Positif, negatif, dan netral
4	Perbandingan metode terhadap analisis sentimen pada ulasan e-wallet aplikasi dana	(Elfansyah, Muhammad Rayhan et al., 2024)	*K-nearest neighbor (KKN) dan Naive Bayes*	*Google Play Store*	Posistif, negatif, dan netral

Pada bagian ini, penulis menelaah berbagai penelitian terdahulu dengan tujuan mendukung dan memperkuat pelaksanaan penelitian ini. Pada penelitian yang dilakukan oleh Ernianti Hasibuana dan Elmo Allistair Heriyanto yang berjudul "Analisis Sentimen pada Ulasan Aplikasi Amazon Shopping di *Google Play Store* Menggunakan *Naive Bayes* Classifier" yang dimana penelitian ini menggunakan metode naïve bayes data pengguna diambil dari *Google Play Store*. Adapun objek yang diteliti adalah aplikasi amazon shopping dengan klasifikasi positif, negatif dan netral. Dengan hasil akhirnya yaitu algoritma naïve bayes Bayes menghasilkan

akurasi rata-rata sebesar 82.15%, *precision* sebesar 72.25%, *recall* sebesar 83.49%, dan *f1-score* sebesar 77.41% (Hasibuan & Heriyanto, 2022).

Penelitian berikutnya yang dilakukan oleh Putri Gea Aryanti dan Imam Santoso yang berjudul "Analisis Sentimen Pada Twitter Terhadap Mobil Listrik Menggunakan Algoritma *Naive Bayes*" yang dimana penelitian ini menggunakan metode naïve bayes, data diambil dari twitter. Adapun objek adalah sentiment terhadap mobil listrik dengan kalisifikasi positif dan negatif. Dengan hasil akhir yaitu akurasi sebesar 87,43% dan AUC 0,518% (Aryanti & Santoso, 2023).

Pada penelitian lain yang dilakukan oleh Rini Maria dan kawan-kawan yang berjudul "Analisis Sentimen Persepsi Masyarakat Terhadap Penggunaan Aplikasi My Pertamina Pada Media Sosial Twitter Menggunakan Metode Naïve Bayes Classifier" yang dimana menggunakan metode naïve bayes data pengguna diambil dari twitter. Adapun objek yang diteliti aplikasi My Pertamina dengan kliasifikasi positif, negatif, dan netral. Dengan hasil askhir akurasinya yaitu akurasi sebesar 84,08%, nilai *precision* sebesar 84,51%, nilai *recall* sebesar 83,81% dan nilai AUC yang didapat sebesar 0,903 (Maria et al., 2023).

Penelitian selanjutnya yang dilakukan oleh

Muhammad Rayhan Elfansyaha dan kawan-kawan yang berjudul "Perbandingan Metode K – Nearest Neighbor (KNN) dan *Naive Bayes* terhadap Analisis Sentimen pada Pengguna E-Wallet Aplikasi DANA Menggunakan Fitur Ekstraksi TF-IDF" yang dimana menggunakan dua metode yaitu KNN dan naïve bayes data pengguna diambil dari *Google Play Store*. Dengan hasil akhir akurasi KNN mencapai 78% dan *Naive Bayes* 74%. Yang dimana metode KNN lebih unggul dari pada metode naïve bayes (Elfansyah, Muhammad Rayhan et al., 2024).

Berdasarkan penelitian terkait yang telah dilakukan oleh beberapa penulis sebelumnya, hasil-hasil tersebut dapat dijadikan sebagai acuan dalam penelitian ini. Peneliti tertarik untuk menganalisis sentimen pada ulasan aplikasi Shell Asia di *Google Play Store* menggunakan metode *Naive Bayes*. Penelitian ini bertujuan untuk mengevaluasi performa metode *Naive Bayes* dalam mengklasifikasikan sentimen ulasan tersebut, yang terdiri dari dua kategori sentimen, yaitu positif dan negatif. Hasil akhir penelitian akan divisualisasikan dalam bentuk word cloud dan diagram batang, dengan data yang diperoleh dari ulasan aplikasi Shell Asia di *Google Play Store*. Penelitian ini dilakukan karena bahasan serupa belum pernah diteliti sebelumnya.

Selain itu, visualisasi hasil analisis sentimen ini nantinya akan dipresentasikan dalam bentuk yang interaktif pada sebuah web sederhana. Web tersebut akan memungkinkan pengguna untuk mengeksplorasi hasil visualisasi, mempermudah pemahaman tentang distribusi sentimen positif dan negatif pada ulasan aplikasi Shell Asia. Di dalam web tersebut, baik word cloud maupun diagram batang akan ditampilkan secara jelas, sehingga memudahkan pengamatan pola sentimen yang muncul dari tanggapan pengguna aplikasi.

BAB III
METODE PENELITIAN

A. Metode Pengumpulan Data

Peneliti melakukan penelitian mendalam dengan membaca berbagai macam sumber seperti buku, jurnal ilmiah, dan skripsi untuk memahami konsep, metode analisis, dan permasalahan yang berhubungan dengan topik penelitiannya. Selain itu, peneliti juga mencari informasi terbaru di internet menggunakan mesin pencari untuk menemukan data dan penelitian sebelumnya yang berkaitan dengan analisis sentimen, pengolahan teks (*Text mining*), dan metode klasifikasi *Naive Bayes*.

Tujuan dari kegiatan ini adalah untuk mengumpulkan informasi yang lengkap dan terkini, sehingga peneliti dapat membandingkan hasil penelitiannya dengan penelitian-penelitian yang sudah ada. Dengan cara ini, peneliti berharap dapat membangun dasar pengetahuan yang kuat dan relevan untuk mendukung penelitiannya.

B. Kebutuhan Perangkan Penelitian

Penelitian ini membutuhkan sejumlah perangkat keras dan perangkat lunak dengan spesifikasi tertentu untuk mendukung kelancaran proses penelitian.

1. Perangkat Keras

Tabel 3. 1 Kebutuhan Perangkat Keras

No	Perangkat Keras	Spesifikasi
1	*Device*	*Asus Vivabook M1403QA*
2	*Processor*	*AMD Ryzen 5 5600H with Radeon Graphics 3.30 GHz*
3	Memori (RAM)	8.00 GB
4	Monitor	14 Inch
5	Keyboard dan Mouse	Normal

2. Kebutuhan Perangkat Lunak

Tabel 3. 2 Kebutuhan Perangkat Lunak

No	Perangkat Keras	Spesifikasi
1	Sistem Operasi	*Windows 11 Home Single Language*
2	Bahasa Pemograman	*Python*
3	*Ms. Office*	*Ms. Word & Ms. Excel*
4	*Browser*	*Chrome*
5	*Goggle Drive*	*Goggle Colab*

C. Langkah Analisis

Tahapan dalam analisis data dalam penelitian ini sebagai berikut:

1. Scrapping data ID aplikasi Shell Asia dari *Google Play Store*
    a. Mengambil ID aplikasi Shell Asia
    b. Scrapping data menggunakan ID aplikasi Shell Asia menggunakan *Python*

c. Menyimpan data bentuk *csv*

2. Menyiapkan data ulasan aplikasi dari goole playstore

    a. Melakukan pelebelan data dengan nilai positif dan negatif

    b. Membaca data *csv* dalam *Python*

3. Text Processing

    a. Melakukan *cleaning data* seperti menghapus URL, tanda baca, emoji dan lain-lain

    b. Melakukan *case folding*

    c. Melakukan *Normalization*

    d. Melakukan *Tokenizing*

    e. Melakukan *stemming*

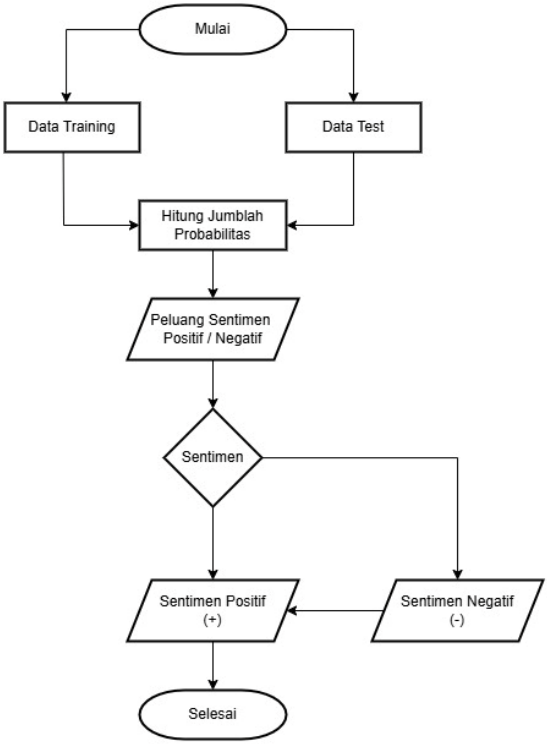
4. Melakukan Split data dengan membagi data training dan data testing

5. Melakukan TF-IDF pada data

6. Klasifikasi *Naive Bayes*

    Berikut alur klasifikasi *Naive Bayes* dituangkan pada flowchart. Flowchart adalah diagram representatif yang menunjukkan langkah-langkah atau urutan operasi yang dilakukan untuk menyelesaikan suatu masalah (Wenty Dwi Yuniarti,

2019). Berikut flowchart yang disajikan pada gambar 3.1.



Gambar 3. 1 Alur Naive Bayes

Proses klasifikasi menggunakan *Naive Bayes* dalam analisis sentimen meliputi beberapa langkah:

a. Identifikasi fitur, yaitu memilih kata atau frasa yang menjadi penanda suatu sentimen.

b. Penghitungan peluang, dengan menghitung frekuensi kemunculan fitur tertentu dalam

sebuah ulasan.

c. Penerapan asumsi independensi, di mana setiap kata dianggap tidak saling berhubungan satu sama lain.

d. Penentuan probabilitas kategori, yaitu menghitung kemungkinan setiap kategori (positif atau negatif) berdasarkan ulasan yang ada.

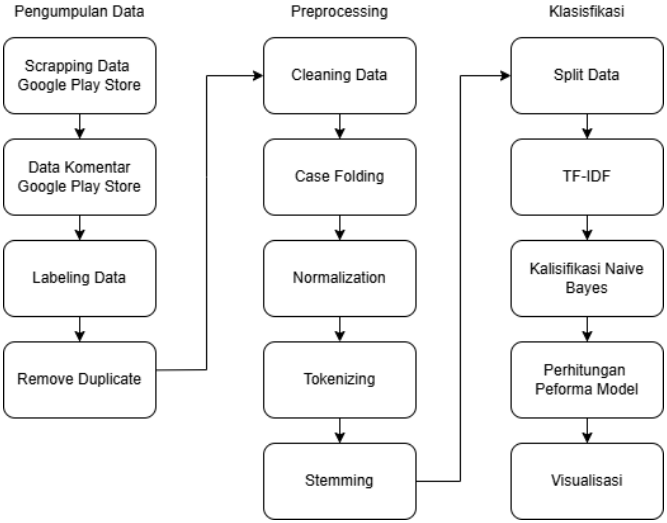
7. Menghitung akurasi, presisi, *recall*, dan *f1-scrore* menggunakan *library Python* untuk setiap jenis klasifikasi, kemudian membandingkan hasilnya.

8. Menentukan hasil klasifikasi terbaik menggunakan metode *Naive Bayes*, dengan disertai visualisasi untuk memudahkan pemahaman.

D. Alur Penelitian

Perancangan alur penelitian ini membahas gambaran umum tahapan pengerjaan yang dilakukan oleh peneliti, dimulai dari awal hingga akhir. Alur penelitian dirancang secara sistematis untuk memastikan seluruh proses dapat dilaksanakan dengan terstruktur dan mencapai tujuan penelitian. Rangkaian proses peneliatian tersebut disajikan pada gambar 3.2 . Alur ini dirancang untuk

memandu peneliti dalam menjalankan penelitian secara sistematis dan terukur.

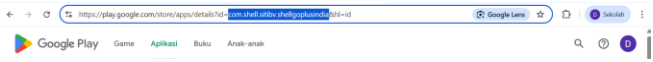


Gambar 3. 2 Alur Penelitian

1. Pengambilan Data

   Data yang digunakan dalam penelitian ini adalah data komentar atau ulasan pengguna aplikasi Shell Asian yang diambil dari *Google Play Store*, menggunakan teknik scraping data. Pada tahap ini, pengumpulan data dilakukan menggunakan bahasa pemrograman *Python* dengan bantuan Google Colaboratory dan package *google-play-scraper*. Proses ini menjamin data yang diperoleh tepat dan sesuai kebutuhan penelitian.

Teknik scrapping data dilakukan dengan membuat program terlebih dahulu, kemudian memasukkan keyword "Aplikasi Shell Asia" pada *Google Play Store.* untuk menampilkan detail aplikasi. Setelah aplikasi Shell Asia muncul, ID aplikasi diambil melalui halaman *Google Play Store.*, di mana ID aplikasi tersebut adalah 'com.shell.sitibv.shellgoplusindia'.



Gambar 3. 3 Pengambilan ID Aplikasi

Data yang diambil sebanyak 3000 data, setelah difiltering dengan rentan data bulan 21 Oktober 2022 sampai dengan 29 November 2024 data yang didapatkan yaitu data. Data yang diambil seperti tanggal, bintang, dan komentar. Data review pengguna aplikasi Shell Asia yang diambil dari situs Google Play berbahasa Indonesia yang kemudian disimpan dalam file berformat *.csv*.

	score	at	content	sentimen
2	5	11/29/2024 17:57	Pilih Shell, lebih bagus!.ðŸ‘ðŸ‘	positif
3	5	11/29/2024 15:07	Kami sebagai pelanggan ,pelayanannya baik	positif
4	5	11/29/2024 13:40	Pelayanan baik, sopan, dan profesional. Terdapat poin-poin yang dapat digunakan untuk pembayaran.	positif
5	5	11/29/2024 11:46	Selain kualitas bahan bakarnya yang bagus, ditambah tidak perlu antre. ðŸ‘ðŸ‘ðŸ‘	positif
6	3	11/29/2024 11:05	Agak sulit untuk menjadi member. Kalau pembayaran lewat apa, Min? Apakah bisa melalui ATM? Terk	negatif
7	5	11/28/2024 23:39	Spbu Shell pelayanan nya memuaskan	positif
8	5	11/28/2024 16:01	Kualitas sangat bagus, takaran juga pas. Sayangnya, belum tersedia di daerah saya, hanya ada di kota-	positif
9	5	11/28/2024 3:02	Shell jangan tutup tolong, aku sudah nyaman sama kualitas mu ðŸ¥°ðŸ¥°, apalagi kualitas oli mu sanga	positif
10	5	11/27/2024 5:25	Mudah fiturnya	positif
11	5	11/26/2024 10:43	Bahan bakar bersih, andalan pokoknya. Harganya cukup bersaing.	positif
12	5	11/26/2024 4:56	Pelayanannya terbaik, cepat, ramah, dan sangat memuaskan.	positif
13	5	11/26/2024 4:55	Tolong jangan tutup dulu Shell di Indonesia. Saya sering menggunakan Shell Super untuk ojol. Walaupu	positif
14	5	11/26/2024 4:44	Shell adalah tempat yang nyaman untuk membeli bahan bakar. Tidak perlu antre lama, pelayanannya	positif
15	1	11/26/2024 3:52	Kenapa tidak bisa daftar? Mau generate OTP tapi ada error.	negatif
16	5	11/25/2024 11:47	Kualitas bahan bakar sangat bagus, membuat performa kendaraan lebih enak dan responsif.	positif
17	5	11/25/2024 9:57	Poin Shell sangat berguna!	positif

Gambar 3. 4 Hasil Scrapping Data

2. Labeling Data

Data yang telah dikumpulkan kemudian dikategorikan berdasarkan sentimen yang terkandung di dalamnya. Dalam penelitian ini, menggunakan dua kategori sentimen, yaitu positif dan negatif. Pelabelan data dilakukan oleh ahli bahasa *(expert)* yaitu Ibu Sri Pujiyati S.Pd guru bahasa Indonesia di SMA Negeri 1 Pundong Bantul. Proses pelabelan ini merupakan langkah awal yang penting dalam proses klasifikasi data.

Tabel 3. 3 Penerapan Labeling

Komentar	Sentimen
Pelayanan baik, sopan, dan profesional. Terdapat poin-poin yang dapat digunakan untuk pembayaran. Lingkungannya juga bersih	positif
agx sulit untuk menjadi member, klo pembayaran lewat apa min atm kah? terkadang saya klo beli shell adanya uang cash, terkadang ditanya punya member gx tapi saya gx tau cara daftarnya +pembayarannya	negatif

3. *Remove Duplicate*

Ulasan pengguna di *Google Play Store.* sering kali memiliki duplikasi, yaitu ulasan yang sama atau sangat mirip. Untuk efisiensi dalam proses *preprocessing* dan analisis data, duplikasi ulasan ini perlu dihapus. Dengan demikian, setiap ulasan unik

hanya akan diproses satu kali. *Remove duplicate* ini juga membantu mencegah bias dalam hasil analisis, sehingga gambaran data yang diperoleh lebih akurat dan representatif.

4. *Preprocessing*

Sebelum melakukan pemodelan, data yang telah dikumpulkan perlu dipersiapkan terlebih dahulu melalui proses *preprocessing*. Tahap ini bertujuan untuk membersihkan dan menyusun data agar siap digunakan dalam analisis lebih lanjut. Tahap preprocessing text ini terdiri dari 5 tahapan yaitu: *Cleaning data*, *Case folding*, *Normalization*, *Tokenizing*, dan *Stemming*. Setiap tahapan dilakukan secara berurutan untuk memastikan kualitas data yang optimal.

1. *Cleaning*

Cleaning merupakan tahapan penting dalam *text preprocessing* yang bertujuan untuk membersihkan teks dari karakter-karakter yang tidak diperlukan seperti URL, tanda baca, emoji, dan pengulangan karakter. Proses ini dilakukan untuk menyederhanakan teks dan memudahkan proses pengolahan data lebih lanjut. Misalnya, dalam ulasan produk, sering ditemukan

penggunaan emoji, simbol, dan tanda baca yang berlebihan. Dengan melakukan *cleaning*, karakter-karakter tersebut akan dihapus sehingga teks menjadi lebih bersih dan siap untuk dianalisis. Contoh *cleaning* pada table 3.4.

Tabel 3. 4 Penerapan Cleaning data

Input Process	Outpu Process
Pelayanan baik, sopan, dan profesional. Terdapat poin-poin yang dapat digunakan untuk pembayaran. Lingkungannya juga bersih	pelayanan baik sopan dan profesional terdapat poinpoin yang dapat digunakan untuk pembayaran lingkungannya juga bersih

2. *Case folding*

Case folding adalah salah satu langkah penting dalam text preprocessing yang berfungsi untuk 38 mengubah semua huruf dalam teks menjadi huruf kecil *(lowercase).* Tujuan utama dari *case folding* adalah untuk menyamaratakan bentuk penulisan kata, sehingga mempermudah proses pengolahan data teks seperti pencarian kata, analisis sentimen, dan pemodelan bahasa. Contoh *case folding* pada table 3.5.

Tabel 3. 5 Penerapan Case folding

Input Process	Outpu Process
pelayanan baik sopan dan profesional terdapat poinpoin yang	pelayanan baik sopan dan profesional terdapat poinpoin

dapat digunakan untuk pembayaran lingkungannya juga bersih	yang dapat digunakan untuk pembayaran lingkungannya juga bersih

3. *Normalization*

Normalisasi adalah proses dalam text *preprocessing* yang bertujuan untuk menyatukan bentuk kata yang berbeda namun memiliki makna yang sama, mencakup kata-kata tidak baku, seperti singkatan, bahasa gaul yang tidak sesuai dengan kamus baku, menjadi bentuk baku yang sesuai dengan kaidah bahasa Indonesia. Contoh *Normalization* 3.6.

Tabel 3. 6 Penerapan Normalization

Input Process	**Outpu Process**
pelayanan baik sopan dan profesional terdapat poinpoin yang dapat digunakan untuk pembayaran lingkungannya juga bersih	pelayanan baik sopan dan profesional terdapat poin poin yang dapat digunakan untuk pembayaran lingkungannya juga bersih

4. *Tokenizing*

Tokenisasi adalah proses membagi teks menjadi unit-unit kata atau token. Tanda baca dan karakter khusus biasanya dihilangkan dalam proses ini karena tidak dianggap sebagai bagian penting dari kata yang akan dianalisis.

Contoh disajikan pada tabel 3.8.

Tabel 3. 7 Penerapan Tokenization

Input Process	Outpu Process
pelayanan baik sopan dan profesional terdapat poin poin yang dapat digunakan untuk pembayaran lingkungannya juga bersih	['pelayanan', 'baik', 'sopan', 'dan', 'profesional', 'terdapat', 'poinpoin', 'yang', 'dapat', 'digunakan', 'untuk', 'pembayaran', 'lingkungannya', 'juga', 'bersih']

5. *Stemming*

Setelah Tokenization selanjutnya tahap *Stemming*. *Stemming* adalah proses mengubah kata-kata menjadi bentuk dasarnya dengan menghapus imbuhan. Contoh *stemming* 3.9.

Tabel 3. 8 Penerapan Stemming

Input Process	Outpu Process
['pelayanan', 'baik', 'sopan', 'dan', 'profesional', 'terdapat', 'poinpoin', 'yang', 'dapat', 'digunakan', 'untuk', 'pembayaran', 'lingkungannya', 'juga', 'bersih']	layan baik sopan dan profesional dapat poinpoin yang dapat guna untuk bayar lingkung juga bersih

5. *Split Validation Data*

Split validation adalah operator yang menerapkan validasi sederhana dengan membagi (split) data set secara acak menjadi data training

20% dan data testing 80% untuk mengevaluasi model yang dihasilkan (Suryono et al., 2018). Pengujian model dilakukan setelah proses pelatihan data selesai. Tahap pengujian ini bertujuan untuk menilai kinerja model. Hasil dari pengujian tersebut akan menunjukkan seberapa baik performa metode yang digunakan (Zidan, 2022).

6. Ektrasi Fitur dengan TF-IDF

Setelah semua data teks melalui tahap preprocessing, langkah berikutnya adalah membuat fitur menggunakan TF-IDF. TF-IDF adalah salah satu teknik pembobotan kata yang digunakan untuk mengukur pentingnya sebuah kata di dalam dokumen tertentu relatif terhadap seluruh kumpulan dokumen atau corpus. Pembobotan kata ini dilakukan dengan menghitung *Term Frequency* yang menilai frekuensi kemunculan kata dalam dokumen serta *Inverse Document Frequency* yang menilai seberapa jarang kata tersebut muncul di semua dokumen. Hasil perhitungan TF dikalikan dengan IDF untuk mendapatkan skor TF-IDF. Setelah data berhasil dilatih kemudian akan dilakukan pengujian data uji untuk menguji ketepatan klasifikasi yang dilakukan.

7. Klasifikasi *Naive Bayes*

Setelah melalui tahap preprocessing dan ekstraksi fitur menggunakan TF-IDF, data kemudian diuji menggunakan data training dan data testing untuk mengevaluasi akurasi klasifikasi dengan metode *Naive Bayes*. Data yang telah dilakukan preprocessing dan ekstraksi fitur, tahap selanjutnya yakni proses pengklasifikasian menggunakan *Naive Bayes*. Peneliti akan menerapkan metode *Naive Bayes*untuk mengukur ketepatan klasifikasi dengan dibagi menjadi dua output yakni positif dan negatif. Kelas dengan skor probabilitas tertinggi akan dianggap sebagai hasil prediksi kelas tersebut.

8. Evaluasi Model

Tujuan dari evaluasi ini adalah untuk menilai performa model yang telah dikembangkan. Kami menggunakan matriks kesalahan *(confusion matrix)* sebagai alat evaluasi. Matriks ini memberikan gambaran rinci tentang kinerja model dalam mengklasifikasikan data. Dari matriks ini, kita dapat mengetahui jumlah data yang diklasifikasikan dengan benar *(true positive, true negative)* dan jumlah data yang diklasifikasikan dengan salah *(false positive, false negative)* (Fikri et al., 2020).

Data uji diterapkan pada model yang dilatih menggunakan data latih, menghasilkan daftar prediksi kelas untuk data uji. Prediksi kelas tersebut kemudian dibandingkan dengan label asli dari data uji, yang sebelumnya disembunyikan. Proses ini digunakan untuk mengevaluasi kinerja model *Naive Bayes*, dengan metrik seperti tingkat akurasi, presisi, *recall*, dan skor F1 (Zidan, 2022).

9. Visualisasi

Proses visualisasi data dilakukan dengan menggunakan *word cloud*, yang menampilkan kata-kata yang paling sering muncul dalam dataset. Semakin sering sebuah kata muncul, semakin besar ukuran kata tersebut pada *word cloud*. Visualisasi ini diterapkan pada seluruh data sentimen, baik positif maupun negatif, yang berasal dari ulasan pengguna tentang aplikasi Shell Asia. Dengan demikian, *word cloud* memberikan gambaran umum mengenai kata-kata yang dominan dalam ulasan, sehingga mempermudah analisis pola sentimen secara keseluruhan.

Hasil visualisasi ini nantinya akan ditampilkan dalam sebuah web sederhana dengan menggunakan *framwork streamlit*, sehingga pengguna dapat

dengan mudah mengakses dan memahami distribusi kata-kata yang sering muncul dalam ulasan.

BAB IV
HASIL DAN PEMBAHASAN

A. Pengambilan Data Ulasan *Google Play Store*

Pada pengambilan data ulasan pengguna aplikasi Shell Asia di *Google Play Store* menggunakan teknik web scrapping. Dimana proses ini menggunakan bahasa pemograman *Python* yang memiliki tahapan-tahapan sebagai berikut:

1. Install Google Play *Scrapper Package*

   Tahap awal dalam proses scraping adalah menginstal *package google-play-scraper*, yang memungkinkan pengambilan data ulasan dari *Google Play Store*.

Gambar 4. 1 Package Google Play Store

2. Pemanggilan *Library*

   Proses scraping memerlukan beberapa *library Python* untuk mengakses berbagai fitur yang dibutuhkan. Yang dimana memiliki fungsi untuk pengambilan data ulasan seperti nama pengguna, gambar profil pengguna, data ulasan, bintang dan lain-

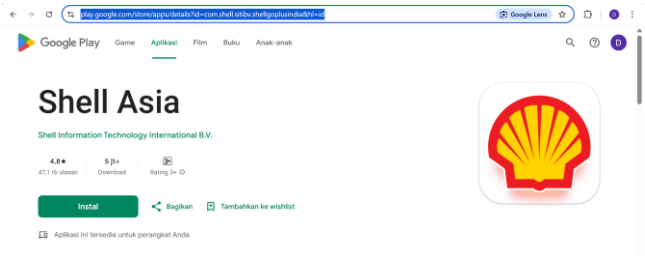
lain. Yang dimana data itu disusun menjadi dataframe.

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

Gambar 4. 2 Pemanggilan Library Proses Scraper

3. Mengambil ID Aplikasi Shell Asia pada Halaman *Google Play Store*

Langkah selanjutnya adalah menyalin ID dari *Google Play Store*. ID ini dimasukkan ke dalam kode program untuk mengambil ulasan yang terkait dengan aplikasi tersebut.



Gambar 4. 3 ID Aplikasi Shell Asia di Google Play Store

4. Scrapping Ulasan Pengguna Aplikasi Shell Asia

Setelah mendapatkan ID aplikasi Shell Asia, proses scraping data ulasan dapat dilakukan. Tahap ini melibatkan penetapan parameter seperti jumlah ulasan yang dibutuhkan, bahasa, dan negara asal pengulas. ID aplikasi dimasukkan ke dalam kode program untuk memulai pengambilan data ulasan

secara otomatis dari *Google Play Store*. Data yang terkumpul kemudian disimpan untuk diproses dalam analisis sentimen.

```
from google_play_scraper import Sort, reviews
result, continuation_token = reviews(
    'com.shell.sitibv.shellgoplusindia',
                lang='id',
    country='id',
    sort=Sort.MOST_RELEVANT,
    count=3000,
    filter_score_with=None
)
```

Gambar 4. 4 Proses Scraping Data Ulasan

Penelitian ini menggunakan *library* "*google_play_scraper*" dalam *Python* untuk melakukan scraping data ulasan aplikasi Shell Asia (ID "com.shell.sitibv.shellgoplusindia") dari Google Play Store. Parameter yang diterapkan meliputi "lang='id'" dan "country='id'" untuk memastikan ulasan yang diambil berbahasa Indonesia dan berasal dari pengguna dalam negeri, serta "sort=Sort.MOST_RELEVANT" untuk mengurutkan ulasan berdasarkan tingkat relevansinya. Penelitian membatasi pengambilan data sebanyak 3000 ulasan melalui parameter "count=3000". Data yang diambil dari semua bintang 1-5 "filter_score_with=None". Proses scraping dilakukan dengan memanggil

fungsi "reviews()" yang akan menyimpan hasilnya dalam variabel "result". Untuk jumlah ulasan melebihi batas maksimum, menggunakan token atau disimpan pada "continuation_token" dan melanjutkan pengambilan data secara berkelanjutan.

5. Menyimpan Data Scraping ke Datafame.

Data hasil scraping diubah menjadi *dataframe* menggunakan *library pandas* dan *numpy* lalu dipisahkan atau dibuat dalam bentuk list sebelum diubah 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()
```

Gambar 4. 5 Proses Mengubah menjadi Dataframe

Data ulasan diubah menjadi dataframe dengan satu kolom "'review'" dalam bentuk dictionary. Kemudian data diextrac menggunakan variabel "pop" dan "join" diubah menjadi list dengan variabel "tolist" agar menjadi kolom yang terpisah. Hasilnya yaitu menjadi kolom-kolom terpisah seperti atribut (username, content, score, dan lain-lain). Lalu "pd.DataFrame" menjadi tabel baru atau dataftrame baru. Dengan "df_busu.head()" berfungsi untuk memanggil atau menampilkan dataframe 5 baris

pertama dengan tabel yang rapih dan tersusun.

6. Mengambil data yang dibutuhkan

Mengambil dataframe yang lebih spesifik dan yang dibutuhkan seperti kolom username, score, at, dan content.

```
df_busu[['userName', 'score','at', 'content']].head()
```

Gambar 4. 6 Memilih Kolom Dataframe

Memilih kolom dari dataframe sorted_df untuk dibuat dataframe baru bernama my_df dan "`[['userName', 'score','at', 'content']]`" memilih 4 kolom yang dibutuhkan.

7. Menyimpan Data Scraping dalam Format csv.

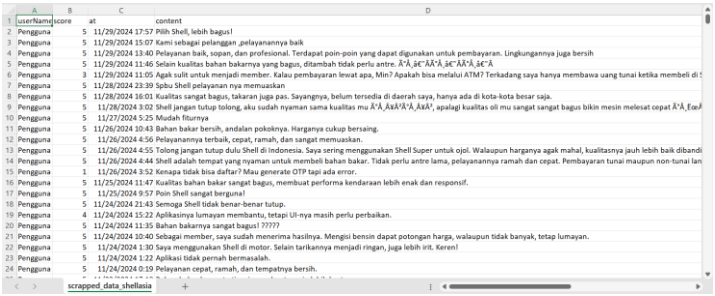
Dataftame disimpan dalam format CSV Dengan nama "scrapped_data_shellasia.csv".

```
scrapped_data_shellasia.csv", index = False)
```

Gambar 4. 7 Menyimpan Dataframe dalam Bentuk CSV

Menyimpan dataframe my_df ke dalam file CSV dengan nama "`scrapped_data_shellasia.csv`". "`index = False`" Menghindari penyimpanan *index* (nomor baris otomatis dari dataframe) ke dalam file CSV. File yang tersimpan akan digunakan untuk proses analisis da klasifikasi dengan metode *naive bayes* Berikut hasil

data scraping yang telah disimpan dalam format csv.



	A	B	C	D
1	userName	score	at	content
2	Pengguna	5	11/29/2024 17:57	Pilih Shell, lebih bagus!
3	Pengguna	5	11/29/2024 15:07	Kami sebagai pelanggan ,pelayanannya baik
4	Pengguna	5	11/29/2024 13:40	Pelayanan baik, sopan, dan profesional. Terdapat poin-poin yang dapat digunakan untuk pembayaran. Lingkungannya juga bersih
5	Pengguna	5	11/29/2024 11:46	Selain kualitas bahan bakarnya yang bagus, ditambah tidak perlu antre. Â˚Â¸â€¯ÃÂ˚Â¸â€¯Â˚Â
6	Pengguna	3	11/29/2024 11:05	Agak sulit untuk menjadi member. Kalau pembayaran lewat apa, Min? Apakah bisa melalui ATM? Terkadang saya hanya membawa uang tunai ketika membeli di S
7	Pengguna	5	11/28/2024 23:39	Spbu Shell pelayanan nya memuaskan
8	Pengguna	5	11/28/2024 16:01	Kualitas sangat bagus, takaran juga pas. Sayangnya, belum tersedia di daerah saya, hanya ada di kota-kota besar saja.
9	Pengguna	5	11/28/2024 3:02	Shell jangan tutup tolong, aku sudah nyaman sama kualitas mu Â˚Â¸ÂœÂ¶Â˚Â¸ÂœÂ², apalagi kualitas oli mu sangat sangat bagus bikin mesin melesat cepat Â˚Â¸Ëœ#
10	Pengguna	5	11/27/2024 5:25	Mudah fitumya
11	Pengguna	5	11/26/2024 10:43	Bahan bakar bersih, andalan pokoknya. Harganya cukup bersaing.
12	Pengguna	5	11/26/2024 4:56	Pelayanannya terbaik, cepat, ramah, dan sangat memuaskan.
13	Pengguna	5	11/26/2024 4:55	Tolong jangan tutup dulu Shell di Indonesia. Saya sering menggunakan Shell Super untuk ojol. Walaupun harganya agak mahal, kualitasnya jauh lebih baik dibandi
14	Pengguna	5	11/26/2024 4:44	Shell adalah tempat yang nyaman untuk membeli bahan bakar. Tidak perlu antre lama, pelayanannya ramah dan cepat. Pembayaran tunai maupun non-tunai lan
15	Pengguna	1	11/26/2024 3:52	Kenapa tidak bisa daftar? Mau generate OTP tapi ada error.
16	Pengguna	5	11/25/2024 11:47	Kualitas bahan bakar sangat bagus, membuat performa kendaraan lebih enak dan responsif.
17	Pengguna	5	11/25/2024 9:57	Poin Shell sangat berguna!
18	Pengguna	5	11/24/2024 21:43	Semoga Shell tidak benar-benar tutup.
19	Pengguna	4	11/24/2024 15:22	Aplikasinya lumayan membantu, tetapi UI-nya masih perlu perbaikan.
20	Pengguna	5	11/24/2024 11:35	Bahan bakarnya sangat bagus! ?????
21	Pengguna	5	11/24/2024 10:40	Sebagai member, saya sudah menerima hasilnya. Mengisi bensin dapat potongan harga, walaupun tidak banyak, tetap lumayan.
22	Pengguna	5	11/24/2024 1:30	Saya menggunakan Shell di motor. Selain tarikannya menjadi ringan, juga lebih irit. Keren!
23	Pengguna	5	11/24/2024 1:22	Aplikasi tidak pernah bermasalah.
24	Pengguna	5	11/24/2024 0:19	Pelayanan cepat, ramah, dan tempatnya bersih.

scrapped_data_shellasia

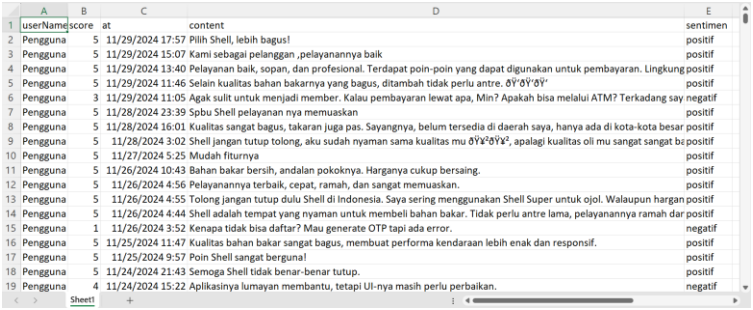
Gambar 4. 8 Data Ulasan yang Disimpan kedalam File CSV

B. Pelebelan pada Data Ulasan

Setelah data ulasan terkumpul, langkah berikutnya adalah memberi label sentimen pada setiap komentar untuk mengategorikannya sebagai positif atau negatif. Peneliti akan memberikan label positif jika ulasan didominasi oleh pujian atau kesan baik terhadap aplikasi, seperti "Aplikasi tidak pernah problem" atau "Ok banyak promonya". Sebaliknya, label negatif diberikan untuk ulasan yang lebih banyak mengandung kritik atau keluhan, misalnya "agx sulit untuk menjadi member, klo pembayaran lewat apa min atm kah? terkadang saya klo beli shell adanya uang cash, terkadang ditanya punya member gx tapi saya gx tau cara daftarnya +pembayarannya".

Dalam penelitian ini, proses pelabelan dilakukan secara manual dengan bantuan seorang mahasiswa

jurusan bahasa yang berpengalaman dalam analisis teks. Namun, karena jumlah datanya sangat banyak (mencapai ribuan ulasan), proses ini memakan waktu cukup lama. Untuk memastikan hasil yang akurat, setiap ulasan harus dibaca dan dinilai satu per satu, terutama yang mengandung kalimat ambigu atau campuran antara pujian dan kritik. Berikut hasil pelebelan manual disajikan pada gambar 4.9



	A	B	C	D	E
1	userName	score	at	content	sentimen
2	Pengguna	5	11/29/2024 17:57	Pilih Shell, lebih bagus!	positif
3	Pengguna	5	11/29/2024 15:07	Kami sebagai pelanggan ,pelayanannya baik	positif
4	Pengguna	5	11/29/2024 13:40	Pelayanan baik, sopan, dan profesional. Terdapat poin-poin yang dapat digunakan untuk pembayaran. Lingkung	positif
5	Pengguna	5	11/29/2024 11:46	Selain kualitas bahan bakarnya yang bagus, ditambah tidak perlu antre. ðŸ‘ðŸ‘	positif
6	Pengguna	3	11/29/2024 11:05	Agak sulit untuk menjadi member. Kalau pembayaran lewat apa, Min? Apakah bisa melalui ATM? Terkadang say	negatif
7	Pengguna	5	11/28/2024 23:39	Spbu Shell pelayanan nya memuaskan	positif
8	Pengguna	5	11/28/2024 16:01	Kualitas sangat bagus, takaran juga pas. Sayangnya, belum tersedia di daerah saya, hanya ada di kota-kota besar	positif
9	Pengguna	5	11/28/2024 3:02	Shell jangan tutup tolong, aku sudah nyaman sama kualitas mu ðŸ¥²ðŸ¥², apalagi kualitas oli mu sangat sangat ba	positif
10	Pengguna	5	11/27/2024 5:25	Mudah fiturnya	positif
11	Pengguna	5	11/26/2024 10:43	Bahan bakar bersih, andalan pokoknya. Harganya cukup bersaing.	positif
12	Pengguna	5	11/26/2024 4:56	Pelayanannya terbaik, cepat, ramah, dan sangat memuaskan.	positif
13	Pengguna	5	11/26/2024 4:55	Tolong jangan tutup dulu Shell di Indonesia. Saya sering menggunakan Shell Super untuk ojol. Walaupun hargan	positif
14	Pengguna	5	11/26/2024 4:44	Shell adalah tempat yang nyaman untuk membeli bahan bakar. Tidak perlu antre lama, pelayanannya ramah dar	positif
15	Pengguna	1	11/26/2024 3:52	Kenapa tidak bisa daftar? Mau generate OTP tapi ada error.	negatif
16	Pengguna	5	11/25/2024 11:47	Kualitas bahan bakar sangat bagus, membuat performa kendaraan lebih enak dan responsif.	positif
17	Pengguna	5	11/25/2024 9:57	Poin Shell sangat berguna!	positif
18	Pengguna	5	11/24/2024 21:43	Semoga Shell tidak benar-benar tutup.	positif
19	Pengguna	4	11/24/2024 15:22	Aplikasinya lumayan membantu, tetapi UI-nya masih perlu perbaikan.	negatif

Sheet1

Gambar 4. 9 Data Ulasan yang Sudah diberi Label Sentimen

C. Remove Duplicate

Remove duplicate adalah menghapus data yang memiliki kesamaan sehingga menjadi satu data saja. Oleh karena itu, nantinya jumlah datanya juga akan berkurang setelah proses penghapusan duplicate. Berikut code program *remove duplicate* disajikan pada gambar 4.10 :

```
#Remove Duplicate
df = df.drop_duplicates(subset=['content'])
df.shape
```

Gambar 4. 10 Source Code Jumlah Data Duplikat

Dari code program setelah dilakukan remove duplicate data awal yang berjumlah 3000 data menjadi 2336 data. Dapat disimpulkan bahwa jumlah dari data yang duplikasi sebanyak 661 data.

D. Preprocessing Data

Sebelum data dapat dianalisis lebih mendalam, penting untuk melakukan tahap persiapan data terlebih dahulu. Proses ini kita sebut sebagai preprocessing, yang bertujuan untuk membersihkan data, menyusun data agar terstruktur sehingga hasil penelitian bisa lebih akurat . Berikut tahapan pada prepocessing pada penelitian ini:

1. *Cleaning data*

   Cleaning data adalah untuk membersihkan data ulasan yang tidak berguna seperti emoji, tanda baca, URL, hastag dan sebagainya. Berikut hasil dari codep program *cleaning data* disajikan pada gambar 4.11.

   Parameter `clean_text(text)` berfungsi untuk membersihkan dan memproses text sesuai dengan yang inginkan. `text = re.sub(r'#\w+', '', text)` berfungsi untuk menghapus hastag. `text = re.sub(r'https?://\S+', '', text)` Berfungsi mengahpus `URL`. `text =`

`re.sub(number_pattern, '', text)` berfungsi menghapus angka. `text = re.sub(r'(?<=\w)\.(?=\w)', ' ', text)` berfungsi menghapus titik antara kata

```
#Cleaning Data
import re
def clean_text(text):
    number pattern = r'\d+'
    text = re.sub(r'#\w+', '', text)
    text = re.sub(r'https?://\S+', '', text)
    text = re.sub(number_pattern, '', text)
    text = re.sub(r'(?<=\w)\.(?=\w)', ' ', text)
    text = re.sub(r'[^\w\s]', '', text)
    text = re.sub(r'\s+', ' ', text)
    text = text.lower()
    return text
def replaceTOM(text):
    pola = re.compile(r'(.)\1{2,}', re.DOTALL)
    return pola.sub(r'\1', text)
def clear_emoji(text):
    return text.encode('ascii',
'ignore').decode('ascii')
def preprocess_text(text):
    text = replaceTOM(text)
    text = clear_emoji(text)
    text = clean_text(text)
    return text
df['clean_text'] =
df['content'].apply(preprocess_text)
df
```

Gambar 4. 11 Source Code Cleaning

`text = re.sub(r'[^\w\s]', '', text)` berfungsi menghapus karkter non-alfanumerik. `text = re.sub(r'\s+', ' ', text)` mengganti banyak spasi dengan satu spasi saja. `replaceTOM(text)` berfungsi menghapus karakter yang berulang menjadi satu karakter saja pada data ulasan. `clear_emoji(text)` berfungsi menghapus

emoji dengan encoding non-ASCII pada `text.preprocess_text(text)` : menjalankan fungsi dari pengahpusan karkter berulang, menghapus emoji, dan membersihkan text keseluruhan.

2. *Case folding*

Tahap *case folding* adalah merubah setiap kata menjadi huruf kecil semua, yang bertujuan untuk mempermudah menganalisis data. Berikut hasil dari codep program case folding disajikan pada gambar 4.12:

```
#Case Folding
    text = text.lower()
    return text
```

Gambar 4. 12 Source Code Case folding

`text.lower()` mengubah text menjadi huruf kecil keseluruhan pada parameter text. return memproses text yang sudah diubah menjadi huruf kecil. Berikut hasil dari code program *cleaning data* dan *case folding* disajikan pada gambar 4.13.

	content	sentimen	clean_text
0	Pilih Shell, lebih bagus!	positif	pilih shell lebih bagus
1	Kami sebagai pelanggan ,pelayanannya baik	positif	kami sebagai pelanggan pelayanannya baik
2	Pelayanan baik, sopan, dan profesional. Terdap...	positif	pelayanan baik sopan dan profesional terdapat ...
3	Selain kualitas bahan bakarnya yang bagus, dit...	positif	selain kualitas bahan bakarnya yang bagus dita...
4	Agak sulit untuk menjadi member. Kalau pembaya...	negatif	agak sulit untuk menjadi member kalau pembayar...
...	...	...	...
2992	The best lah	positif	the best lah
2993	syukak	positif	syukak

Gambar 4. 13 Hasil Proses Case folding dan Cleansing

3. *Normalization*

Tahap *Normalization* adalah membenarkan kata-kata yang tidak baku seperti kata gaul, singkatan menjadi kata baku. Code program *Normalization* ditampilkan pada gambar 4.14.

```
#Normalisasi Data
norm = {
        'mantapp': 'mantap', 'agx':
        'lumayan', 'bang': 'abang', 'apk':
        'aplikasi', 'dpet': 'dapat',
        'pdhl': 'padahal', 'utk': 'untuk',
        'ga': 'tidak', 'guna': 'berguna',
        'bs':
        'rek': 'teman', 'ttp':
        'tutup',    'mannntappp': 'mantap',
        'dlo': 'dulu', 'tiap': 'setiap',
        'pakek': 'pakai', 'cuman': 'cuma',
        'mantul': 'mantap',
        'markotop': 'baik', 'naek'
 }
def normalisasi(str_text):
    for key, value in norm.items():
        str_text = re.sub(rf'\b{key}\b',
value, str_text)
    return str_text

df['normalisasi_text'] =
df['clean_text'].apply(lambda x:
normalisasi(x))
df
```

Gambar 4. 14 Source Code Normalization

Kamus normalisasi `norm` berisi kata yang berpasangan kata baku dan kata yang tidak baku. `normalisasi(str_text)` melakukan proses penerjemahan atau normalisasi. Berikut hasil dari codep program normalisasi disajikan pada gambar 4.14.

	content	sentiment	clean_text	normalisasi_text
0	Pilih Shell, lebih bagus!	positif	pilih shell lebih bagus	pilih shell lebih bagus
1	Kami sebagai pelanggan ,pelayanannya baik	positif	kami sebagai pelanggan pelayanannya baik	kami sebagai pelanggan pelayanannya baik
2	Pelayanan baik, sopan, dan profesional. Terdap...	positif	pelayanan baik sopan dan profesional terdapat ...	pelayanan baik sopan dan profesional terdapat ...
3	Selain kualitas bahan bakarnya yang bagus, dit...	positif	selain kualitas bahan bakarnya yang bagus dita...	selain kualitas bahan bakarnya yang bagus dita...
4	Agak sulit untuk menjadi member. Kalau pembaya...	negatif	agak sulit untuk menjadi member kalau pembayar...	lumayan sulit untuk menjadi member kalau pemba...
...	...	...	...	...
2992	The best lah	positif	the best lah	the best lah
2993	syukak	positif	syukak	suka
2994	Mantap gan,,,,	positif	mantap gan	mantap gan

Gambar 4. 15 Hasil Proses Normalization

4. Tokenization

Tokenization merupakan proses pemecahan teks menjadi unit-unit terkecil berupa kata-kata individual. Tahap ini mengubah rangkaian kalimat menjadi kumpulan kata terpisah guna memudahkan proses analisis lebih lanjut. Berikut code program tokenization disajikan pada gambar 4.16:

```
#Tokenize
import nltk.corpus
nltk.download('punkt')
from nltk.tokenize import sent_tokenize,
word_tokenize
df['tokenized_text'] =
df['normalisasi_text'].apply(lambda x:
x.split())
df
```

Gambar 4. 16 Source Code Tokenization

Modul "`sent_tokenize`" berfungsi untuk memisahkan text menjadi kalimat. "`word_tokenize`" berfungsi untuk memisahkan text menjadi kata-kata. Berikut hasil dari code program tokenization disajikan pada gambar 4.19.

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

	content	sentimen	clean_text	normalisasi_text	tokenized_text
0	Pilih Shell, lebih bagus!	positif	pilih shell lebih bagus	pilih shell lebih bagus	[pilih, shell, lebih, bagus]
1	Kami sebagai pelanggan ,pelayanannya baik	positif	kami sebagai pelanggan pelayanannya baik	kami sebagai pelanggan pelayanannya baik	[kami, sebagai, pelanggan, pelayanannya, baik]
2	Pelayanan baik, sopan, dan profesional. Terdap...	positif	pelayanan baik sopan dan profesional terdapat ...	pelayanan baik sopan dan profesional terdapat ...	[pelayanan, baik, sopan, dan, profesional, ter...
3	Selain kualitas bahan bakarnya yang bagus, dit...	positif	selain kualitas bahan bakarnya yang bagus dita...	selain kualitas bahan bakarnya yang bagus dita...	[selain, kualitas, bahan, bakarnya, yang, bagu...
4	Agak sulit untuk menjadi member. Kalau pembaya...	negatif	agak sulit untuk menjadi member kalau pembayar...	lumayan sulit untuk menjadi member kalau pemba...	[lumayan, sulit, untuk, menjadi, member, kalau...
...	...	...	...	...	...
2992	The best lah	positif	the best lah	the best lah	[the, best, lah]
2993	syukak	positif	syukak	suka	[suka]

Gambar 4. 17 Hasil Proses Tokenization

5. *Stemming*

Tahap *stemming* merupakan tahap untuk menanggalkan imbuhan pada suatu kata sehingga diperoleh bentuk dasar kata tersebut. Dalam pelaksanaannya, diperlukan pemasangan pustaka Sastrawi terlebih dahulu. Berikut code program untuk menginstall sastrawi dari stimming disajikan pada gambar 4.18 :

```
!pip install Sastrawi swifter
```

Gambar 4. 18 Install Library Sastrawi

Setelah menginstall *library* paket *Python* "`Sastrawi`" dengan bahasa Indonesia. Selanjutnya mengimpor kelas "`StemmerFactory`" dari kelas sastrawi. Fungsi dari "`stemmer`" objek utama dalam proses *stemming*. "`stemmed_wrapper(term)`" untuk menerima parameter "term" kata yang akan di stem. term_dict membuat kamus kosong untuk menyimpan kata yang akan diproses. "`term_dict[term] = stemmed_wrapper(term)`"

melakukan *stemming* pada kata dan hasil yang disimpan pada "`term_dict.get_stemmed_term(document)`" menerima satu parameter "`document`" yaitu daftar kata pada sistem dokumen dan disimpan pada kolom baru "stemmed_text". Berikut code program dari stimming pada gambar 4.19.

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

def stemmed_wrapper(term):
    return stemmer.stem(term)
term_dict = {}
hitung=0

for document in df['tokenized_text']:
    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]
df['stemmed_text'] =
df['tokenized_text'].apply(lambda x:'
'.join(get_stemmed_term(x)))
df
```

Gambar 4. 19 Source Code Stemming

Berikut hasil dari code program *stemming* yang

disajikan pada gambar 4.20.

```
3558 : bergembira : gembira
3559 : gan : gan
{'pilih': 'pilih', 'shell': 'shell', 'lebih': 'lebih', 'bagus': 'bagus', 'kami': 'kami', 'sebagai': 'bagai', 'pelanggan': 'langgan', 'pelayanann
----------------------
```

	content	sentimen	clean_text	normalisasi_text	tokenized_text	stemmed_text
0	Pilih Shell, lebih bagus!	positif	pilih shell lebih bagus	pilih shell lebih bagus	[pilih, shell, lebih, bagus]	pilih shell lebih bagus
1	Kami sebagai pelanggan ,pelayanannya baik	positif	kami sebagai pelanggan pelayanannya baik	kami sebagai pelanggan pelayanannya baik	[kami, sebagai, pelanggan, pelayanannya, baik]	kami bagai langgan layan baik
2	Pelayanan baik, sopan, dan profesional. Terdap...	positif	pelayanan baik sopan dan profesional terdapat ...	pelayanan baik sopan dan profesional terdapat ...	[pelayanan, baik, sopan, dan, profesional, ter...	layan baik sopan dan profesional dapat poinpoi...
3	Selain kualitas bahan bakarnya yang bagus, dit...	positif	selain kualitas bahan bakarnya yang bagus dita...	selain kualitas bahan bakarnya yang bagus dita...	[selain, kualitas, bahan, bakarnya, yang, bagu...	selain kualitas bahan bakar yang bagus tambah ...
4	Agak sulit untuk menjadi member. Kalau pembaya...	negatif	agak sulit untuk menjadi member kalau pembayar...	lumayan sulit untuk menjadi member kalau pemba...	[lumayan, sulit, untuk, menjadi, member, kalau...	lumayan sulit untuk jadi member kalau bayar le...
...	...	...	...	...	...	...
2992	The best lah	positif	the best lah	the best lah	[the, best, lah]	the best lah
2993	syukak	positif	syukak	suka	[suka]	suka

Gambar 4. 20 Hasil Proses Stemming

E. *Split Validation Data*

Split Validation Data adalah membagi data ulasan menjadi data training dan data testing. Dalam penelitian ini perbandingan 80:20, yang dimana data testing 20% dari keseluruhan. Pembagian ini dilakukan secara acak untuk memastikan distribusi data yang merata. Berikut code program dari *Split Validation Data* disajikan pada gambar 2.21:

```
#Tahapan split validation
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,
y, test_size=0.2, random_state=42)
print(X_train.shape)
print(y train.shape)
print(X_test.shape)
print(y_test.shape)
```

Gambar 4. 21 Source Code Split Validation Data

Mengimport *library* `'sklearn'`. `'model_selection'` fungsinya untuk membagi dataset

menjadi kelas `train_test_split`.

F. TF-IDF

TF-IDF (*Term Frequency*-Invers *Document Frequency*) adalah proses merubah kata menjadi vector. Pada tahapan TF-IDF ini memanfaatkan `TfidfVectorizer` dari *library* *`sklearn`*. Parameter `vectorizer.fit(X_train)` Mempelajari kata-kata unik pada data train dan menghitung bobot kata pada keseluruhan dokumen IDF. Proses ini menghasilkan representasi numerik dimana kata-kata yang lebih penting akan memiliki bobot yang lebih tinggi. Contoh dari tahapan mengubah text menjadi vector disajikan pada gambar 2.23:

```
#Tahapan merubah data menjadi vector
from sklearn.feature_extraction.text import
TfidfVectorizer
vectorizer = TfidfVectorizer()
vectorizer.fit(X_train)
```

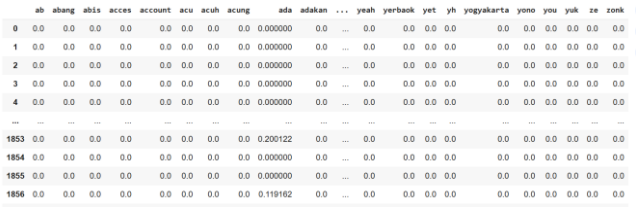
Gambar 4. 22 Source Code Mengubah Data menjadi Vector

Selanjutnya, data training dan testing diubah menjadi matriks numerik TF-IDF fungsi `transform()` lalu di konversi ke bentuk tabel `toarray()`. Fungsi utama dalam code ini yaitu mengubah data text menjadi angka supaya bisa memproses untuk analisis. Code program tahapan TF-IDF disajikan pada gambar 2.23:

```
#Tahapan TF-IDF
X_train_TFIDF = vectorizer.transform(X_train).toarray()
X_test_TFIDF = vectorizer.transform(X_test).toarray()
X_all_TFIDF =
vectorizer.transform(df["stemmed_text"]).toarray()
kolom = vectorizer.get_feature_names_out()
train_tf_idf = pd.DataFrame(X_train_TFIDF,
columns=kolom)
test_tf_idf = pd.DataFrame(X_test_TFIDF, columns=kolom)
train_tf_idf
```

Gambar 4. 23 Source Code Pembobotan TF-IDF

Setelah dilakukan pembobotan pada text, berikut hasil TF-IDF disajikan pada gambar 4.6.



	ab	abang	abis	acces	account	acu	acuh	acung	ada	adakan	...	yeah	yerbaok	yet	yh	yogyakarta	yono	you	yuk	ze	zonk
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	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	0.000000	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	0.000000	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	0.000000	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	0.000000	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1853	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.200122	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1854	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1855	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1856	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.119162	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Gambar 4. 24 Hasil Proses Pembobotan TF-IDF

Dalam penelitian ini, contoh perhitungan TF-IDF secara manual sebagai berikut :

(Doc 1) = “Pilih Shell, lebih bagus!”

(Doc 2) = “Kami sebagai pelanggan, pelayanannya baik”

(Doc 3) = “Spbu Shell pelayanannya memuaskan”

Setelah dilakukan text preprocessing maka data menjadi seperti :

(Doc 1) = [“pilih”, “shell”, ”lebih”, “bagus”]

(Doc 2) = [“kami”, “bagai”, “langgan”, “layan”, “baik”]

(Doc 3) = [“spbu”, “shell”, “layan”, “puas”]

Tahap selanjutnya yaitu perhitungan TF-IDF menggunakan word vector. *Term Frequency* (TF) yang mengukur frekuensi kemunculan suatu kata dalam dokumen tertentu. Berikut perhitungan TF pada tabel disajikan pada tabel 4.1.

Tabel 4. 1 Contoh Perhitungan TF

Kata	*Term Frequency*		
	D1	D2	D3
pilih	1	0	0
shell	1	0	1
lebih	1	0	0
bagus	1	0	0
kami	0	1	0
bagai	0	1	0
langgan	0	1	0
layan	0	1	1
baik	0	1	0
spbu	0	0	1
puas	0	0	1

Setelah didapatkan nilai TF, supaya dapat masuk ke perhitungan IDF maka harus menentukan DF (*Document Frequency*) terlebih dahulu. DF sendiri merupakan banyaknya jumlah dokumen yang mengandung kata tersebut. Berikut perhitungan DF disajikan pada tabel 4.2.

Tabel 4. 2 Contoh Perhitungan DF

Kata	*Term Frequency*			DF
	D1	D2	D3	
pilih	1	0	0	1
shell	1	0	1	2
lebih	1	0	0	1

Kata	D1	D2	D3	DF
bagus	1	0	0	1
kami	0	1	0	1
bagai	0	1	0	1
langgan	0	1	0	1
layan	0	1	1	2
baik	0	1	0	1
spbu	0	0	1	1
puas	0	0	1	1

Setelah dihitungnya nilai DF, langkah selanjutnya adalah menghitung nilai IDF. Adapun perhitungan IDF disajikan pada tebel 4.3.

Tabel 4. 3 Contoh Perhitungan IDF

Kata	DF	D/DF D=3	IDF (Log(D/DF)	IDF+1
pilih	1	3	0.477	1.477
shell	2	1.5	0.176	0.176
lebih	1	3	0.477	1.477
bagus	1	3	0.477	1.477
kami	1	3	0.477	1.477
bagai	1	3	0.477	1.477
langgan	1	3	0.477	1.477
layan	2	1.5	0.176	0.176
baik	1	3	0.477	1.477
spbu	1	3	0.477	1.477
puas	1	3	0.477	1.477

Setelah diketahui nilai TF dan IDF, langkah selanjutnya yaitu menghitung nilai TF-IDF. Adapun perhitungan TF-IDF disajikan pada tabel 4.4.

Tabel 4. 4 Hasil Perhitungan TF-IDF

Kata	DF	D/DF D=3	IDF (Log(D /DF)	IDF+1	D1	D2	D3
pilih	1	3	0.477	1.477	1.477	0	0

Kata	DF	D/DF D=3	IDF (Log(D /DF)	IDF+1	D1	D2	D3
shell	2	1.5	0.176	0.176	0.176	0	0.176
lebih	1	3	0.477	1.477	1.477	0	0
bagus	1	3	0.477	1.477	1.477	0	0
kami	1	3	0.477	1.477	0	1.477	0
bagai	1	3	0.477	1.477	0	1.477	0
langgan	1	3	0.477	1.477	0	1.477	0
layan	2	1.5	0.176	0.176	0	0.176	0.176
baik	1	3	0.477	1.477	0	1.477	0
spbu	1	3	0.477	1.477	0	0	1.477
puas	1	3	0.477	1.477	0	0	1.477

Sehingga apabila hasil TFIDF dituliskan dalam bentuk baris kolom menjadi seperti dibawah ini :

Tabel 4. 5 Tabel train_tf_idf

	pilih	shell	lebih	bagus	...	layan	baik	spbu	puas
D1	1,477	1,176	1,477	1.477	...	0	0	0	0
D2	0	0	0	0	...	1,176	1.477	0	0
D3	0	1,176	0	0	...	1,176	0	1.477	1.477

Setiap baris dalam array tersebut merepresentasikan nilai TF-IDF untuk setiap kata dalam dokumen. Nilai TF (*Term Frequency*) mencerminkan frekuensi kemunculan suatu kata dalam dokumen tertentu, di mana semakin sering sebuah kata muncul, semakin tinggi nilai TF-nya. Sementara itu, IDF (*Inverse Document Frequency*) mengukur seberapa umum suatu kata muncul di seluruh dokumen. IDF memiliki sifat terbalik - kata yang sering muncul di banyak dokumen akan memiliki nilai IDF kecil.

G. Klasifikasi *Naive Bayes*

Setelah dilakukan preprocessing, split data, dan pembobotan TF-IDF langkah selanjutnya yaitu klasifikasi menggunakan algoritma *Naive Bayes*. Dalam algoritma klasifikasi *Naive Bayes*, analisis sentimen dilakukan dengan menghitung probabilitas untuk setiap kategori, misalnya positif dan negatif. Kategori dengan nilai probabilitas yang lebih tinggi akan dipilih sebagai label sentimen dari suatu komentar. Contoh proses perhitungan sentimen dalam algoritma naïve bayes secara manual sebagai berikut:

(Doc|Negatif) : ["registrasi", "susah", "banget", "tunggu", "sms", "otp", "lama", "sekali", "akhir", "saya", "nyerah", "dan", "uninstall", "aplikasi", "shell", "mohon", "baik"]

(Doc|Positif) : ["aplikasi", "yang", "bantu", "karena", "setiap", "beli", "di", "shell", "dapat", "poin", "dan", "bisa", "tukar", "dengan", "uang", "atau", "barang", "lain"]

Data yang disajikan adalah sata yang telah diproses pada tahapan preprocessing. Lalu setelah ini akan dihitung term yang beradapa pada sentiment negatif dan negatif. Berikut perhitungan term disajikan pada tabel 4.6

Tabel 4. 6 Term class frequency

Term	P(X|Negatif)	P(X|Positif)
registrasi	1	0
susah	1	0

Term	P(X|Negatif)	P(X|Positif)
banget	1	0
tunggu	1	0
sms	1	0
otp	1	0
lama	1	0
sekali	1	0
akhir	1	0
saya	1	0
nyerah	1	0
uninstall	1	0
aplikasi	1	1
mohon	1	0
baik	1	0
yang	0	1
bantu	0	1
karena	0	1
tiap	0	1
beli	0	1
di	0	1
shell	1	1
dapat	0	1
point	0	1
dan	1	1
bisa	0	1
tukar	0	1
dengan	0	1
uang	0	1
barang	0	1
lain	0	1
Term	**17**	**17**

Diperoleh jumlah P(X|Positif) 17, P(X|Negatif) 17 dengan jumlah kata 31.

1. Perhitungan probabilitas prior :

   a. Jumlah prior positif (N_{pos}) = 1

   b. Jumlah prior negatif (N_{neg}) = 1

   c. Probabilitan prior positif

   $$P\ (Positif) = \frac{N_{pos}}{N_{pos}+N_{neg}} = \frac{1}{2} = 0.5$$

   d. Probabilitas prior negatif

   $$P\ (Negatif) = \frac{N_{neg}}{N_{pos}+N_{neg}} = \frac{1}{2} = 0.5$$

2. Menghitung probabilitas likehood untuk sentiment positif dan negatif, seperti yang ditunjukan pada tabel:

Tabel 4. 7 Likelihood Probability

Term	**Probabilitas**	
	P(X|Negatif)	**P(X|Positif)**
registrasi	0.0417	0.0208
susah	0.0417	0.0208
banget	0.0417	0.0208
tunggu	0.0417	0.0208
sms	0.0417	0.0208
otp	0.0417	0.0208
lama	0.0417	0.0208
sekali	0.0417	0.0208
akhir	0.0417	0.0208
saya	0.0417	0.0208

Term	Probabilitas	
	P(X|Negatif)	P(X|Positif)
nyerah	0.0417	0.0208
uninstall	0.0417	0.0208
aplikasi	0.0417	0.0417
mohon	0.0208	0.0208
baik	0.0208	0.0208
yang	0.0208	0.0417
bantu	0.0208	0.0417
karena	0.0208	0.0417
tiap	0.0208	0.0417
beli	0.0208	0.0417
di	0.0208	0.0417
shell	0.0417	0.0417
dapat	0.0208	0.0417
point	0.0208	0.0417
dan	0.0417	0.0417
bisa	0.0208	0.0417
tukar	0.0208	0.0417
dengan	0.0208	0.0417
uang	0.0208	0.0417
barang	0.0208	0.0417
lain	0.0208	0.0417

Untuk memudahkan pemahaman, berikut contoh perhitungan manual likelihood pada kata ["registasi", "susah", "aplikasi", "shell"]

a. P (“registrasi”|pos) = $\frac{N_a}{N_{pos}} = \frac{0+1}{17+31} = \frac{1}{48} = 0.0208$

b. P (“regristrasi”|neg) = $\frac{N_a}{N_{neg}} = \frac{1+1}{17+31} = \frac{2}{48} = 0.0417$

c. P (“susah”|pos) = $\frac{N_a}{N_{pos}} = \frac{0+1}{17+31} = \frac{1}{48} = 0.0208$

d. P (“susah”|neg) = $\frac{N_a}{N_{neg}} = \frac{2+1}{17+31} = \frac{1}{48} = 0.417$

e. P (“aplikasi”|pos) = $\frac{N_a}{N_{pos}} = \frac{1+1}{17+31} = \frac{2}{48} = 0.0417$

f. P (“aplikasi”|neg) = $\frac{N_a}{N_{neg}} = \frac{1+1}{17+31} = \frac{2}{48} = 0.0417$

g. P (“shell”|pos) = $\frac{N_a}{N_{pos}} = \frac{1+1}{17+31} = \frac{2}{48} = 0.0417$

h. P (“shell”|neg) = $\frac{N_a}{N_{neg}} = \frac{1+1}{17+31} = \frac{2}{48} = 0.0417$

Setelah memperoleh nilai probabilitas untuk setiap kata langkah berikutnya adalah melakukan pengujian pada data uji menggunakan model *Naive Bayes*. Berikut data yang akan di uji "saya mau registrasi aplikasi tidak bisa kode otp tidak masuk"

a. Sentimen Positif

P(Positif|Doc) $\propto$ P(“saya”) × P("registrasi"|Positif) × P("aplikasi"|Positif) × P("otp"|Positif)= $0.5 \times 0.0208 \times 0.0208 \times 0.0417 \times 0.0208 = 1.87627315E-7$

b. Sentimen Negatif

P(Negatif|Doc) $\propto$ P(“saya”) × P("registrasi"|Negatif) × P("aplikasi"|Negatif) × P("otp"|Negatif)= $0.5 \times 0.0417 \times 0.0417 \times 0.0417 \times 0.0417 = 0.00000151$

Berdasarkan perhitungan di atas, dapat dilihat bahwa P(negatif|Uji) > P(positif|Uji) dengan nilai probabilitas tertinggi sebesar 0.00000151. Oleh karena itu, sampel data uji tersebut diklasifikasikan sebagai negatif

Selain itu, penggunaan `TfidfVectorizer` dapat membantu mengonversi teks menjadi representasi numerik berbobot TF-IDF, sehingga model *Naive Bayes* dapat memprosesnya secara efektif. Hasil pelatihan model kemudian dievaluasi menggunakan metrik seperti *akurasi, presisi, recall,* dan *F1-score* untuk memastikan kinerja optimal dalam klasifikasi teks

Selanjutnya penelitian ini untuk melakukan klasifikasi teks menggunakan algoritma *Naive Bayes*. Beberapa komponen penting dari *sklearn* yang diperlukan seperti *MultinomialNB* untuk algoritma *naïve bayes*, *accuracy*_score, *precision*_score, *recall*_score, f1_score untuk matrik evaluasi dan *classification_report, confusion_matrix.*

Dalam pemrosesan teks dengan Python, kita dapat menganalisis tingkat kepentingan setiap fitur (kata) menggunakan algoritma Naive Bayes dari library scikit-learn. Membuat sebuah objek dari kelas `MultinomialNB()`.Kemudian, `clf.fit(X_train_TF IDF,y_train)`melatih model ini dengan data latih.

Berikut code program untuk mengimport *library sklearn* dan proses klasifikasi menggunakan algoritma *Naive Bayes* yang disajikan pada gambar 4.27

```
# Tahap perhitungan model Naive Bayes
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.model_selection import train_test_split
from sklearn.metrics import classification_report,
confusion_matrix

clf = MultinomialNB()
clf.fit(X_train_TFIDF, y_train)
predicted = clf.predict(X_test_TFIDF)

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("MultinomialNB Accuracy:", accuracy_score(y_test,
predicted))
print(f'\nConfusion Matrix: \n{confusion_matrix(y_test,
predicted)}')
print('=====================================================\
n')
print(classification_report(y_test, predicted,
zero division=0))
```

Gambar 4. 25 Source Code klasifikasi dengan Metode Naive Bayes

H. Uji Model

Tahap ini bertujuan untuk mengevaluasi akurasi dan ketepatan model dalam pengklasifikasian serta mendapatkan hasil klasifikasi yang ditunjukkan dalam tabel 2.1, yang merupakan *confusion matrix*. Tabel ini mencakup dua kelas, yaitu kelas prediksi *(predicted class)* dan kelas yang sebenarnya *(true class).* Penelitian ini menggunakan

Confusion matrix karena terdapat dua kelas yang dianalisis, yaitu kelas negatif dan positif. Diketauhi sebelumnya dari tahapan uji model didapatkan hasil *Confusion matrix* 2x2 ditunjukan pada tabel:

Tabel 4. 8 Confusion matrix Dua Kelas

		Predicted Class	
		Negative	Positive
True Class	Negative	(TN) True Negative	(FP) False Positive
	Positive	(FN) False Negative	(TP) True Positive

Untuk mengukur tingkat keakuratan model dalam melakukan klasifikasi, perlu dilakukan perhitungan nilai akurasi. Caranya adalah dengan menjumlahkan seluruh data yang berhasil diprediksi dengan data sebenarnya, kemudian membaginya dengan jumlah total data uji yang digunakan. Rumus perhitungan akurasi dapat dilihat pada persamaan berikut:

$$Akurasi = \frac{TN + TP}{Total\ data\ yang\ diuji} \times 100\%$$

Setelah dilakukan tahapan uji model, sehinggal didapatkan hasil nilai akurasi dan *Confusion matrix* 2x2 yang disajikan pada gambar 4.26

```
MultinomialNB Accuracy: 0.9139784946236559

Confusion Matrix:
[[149  24]
 [ 16 276]]
```

Gambar 4. 26 Hasil Uji Model

Pada perhitungan klasifikasi menggunakan code tersebut, hasil dari klasifikasi Naïve Bayes untuk akurasi bernilai 0,9139. Namun, menghitung nilai akurasi saja tidaklah cukup karena jumlah data tiap kelas pada data latih tidaklah seimbang, maka perlu dilakukan evaluasi menggunakan metrik lain seperti precision, recall, dan F1-score yang lebih mampu mencerminkan performa model pada data tidak seimbang..

I. Evaluasi Model

Tahap evaluasi model dilakukan untuk mengukur kinerja model yang telah dikembangkan. Dalam penelitian ini, dilakukan perhitungan performa menggunakan beberapa metrik evaluasi, yaitu akurasi, presisi, recall, dan F1-score. Nilai-nilai tersebut diperoleh melalui analisis Confusion matrix berukuran 2×2. Berikut tabel Confusion matrix disajikan pada tabel 4.7.

Tabel 4. 9 Hasil Matrix Confused

		Predicted Class	
		Negative	Positive
True Class	Negative	149	24
	Positive	16	276

Peneliti melakukan perhitungan manual untuk mengevaluasi performa model *Naive Bayes* dengan mengukur nilai akurasi berdasarkan tabel *Confusion matrix* yang telah disusun. Proses perhitungan akurasi secara manual dapat diuraikan melalui langkah-langkah berikut :

$$Akurasi = \frac{TN + TP}{Total\ data\ yang\ diuji} \times 100\%$$

$$Akurasi = \frac{276 + 149}{276 + 149 + 24 + 16} \times 100\%$$

$$Akurasi = \frac{425}{465} \times 100\%$$

$$Akurasi = 0.9139\ \times 100\ \%$$

$$Akurasi = 91\%$$

Untuk mengevaluasi performa model pada metrik presisi, *recall*, dan *F1-score*, diperlukan penentuan nilai *true positive (TP), true negative (TN), false positive (FP),* dan *false negative (FN)* untuk setiap kelas dalam *confusion matrix*. Namun, pada matriks berukuran 2×2, proses identifikasi nilai-nilai ini menjadi cukup kompleks karena keterbatasan representasi kelas. Oleh karena itu, untuk mempermudah perhitungan, dilakukan transformasi matriks menjadi format 2×2 terlebih dahulu untuk setiap kelas yang dievaluasi.

1. Kelas Negatif

Tabel 4. 10 Perhitungan Kelas Negatif

True/Pred	*Negatif*	*Bukan*
Negatif	TP = 149	FN = 24
Bukan	FP = 16	TN = 276

2. Kelas Positif

Tabel 4. 11 Perhitungan Kelas Positif

True/Pred	*Positif*	*Bukan*
Positif	TP= 276	FN = 16
Bukan	FP= 24	TN = 149

Setelah diperoleh nilai *true positive (TP), true negative (TN), false positive (FP), dan false negative (FN),* maka perhitungan metrik evaluasi dapat dilakukan. Berdasarkan rumus yang telah dijelaskan pada bab sebelumnya, berikut perhitungan nilai *precision*, *recall*, dan *F1-score* secara manual

1. Perhitungan Kelas Negatif

$$Precision = \frac{TP}{TP + FP} = \frac{149}{149 + 16}$$
$$= \frac{149}{165} = 0.90$$
$$Recall = \frac{TP}{TP + FN} = \frac{149}{149 + 24}$$
$$= \frac{149}{173} = 0.86$$
$$f1 - Scrore = 2 \times \frac{Precision * Recall}{Precision + Recall}$$
$$= 2 \times \frac{0.90 \times 0.86}{0.90 + 0.86} = 0.88$$

2. Perhitungan Kelas Positif

$$Precision = \frac{TP}{TP + FP} = \frac{276}{276 + 24}$$

$$= \frac{276}{300} = 0.92$$

$$Recall = \frac{TP}{TP + FN} = \frac{276}{276 + 16}$$

$$= \frac{276}{292} = 0.95$$

$$f1 - Scrore = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

$$= 2 \times \frac{0.92 \times 0.95}{0.92 + 0.95} = 0.93$$

Berdasarkan perhitungan manual diatas dapat disimpulkan bahwa nilai precision dihitung pada nilai negatif sebesar 90%, nilai positif sebesar 92%. Untuk nilai recall dapat dihitung pada nilai negatif sebesar 86%, nilai positif sebesar 95%. Sedangkan untuk nilai f1 score pada nilai negatif sebesar 88% dan nilai positif sebesar 93%. Untuk mengimplementasikan perhitungan nilai akurasi, *precision, recall*, dan *F1-Score* pada suatu sistem secara otomatis, Anda dapat menggunakan kode *Python* dengan *library scikit-learn* seperti yang ditunjukkan pada Gambar 4.29. Kode tersebut akan mempermudah evaluasi model dengan menghasilkan metrik-metrik tersebut secara akurat dan efisien, sekaligus memastikan konsistensi hasil

antara perhitungan manual dan komputasi otomatis.

```
# Tahap perhitungan model Naive Bayes
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.model_selection import train_test_split
from sklearn.metrics import classification_report,
confusion_matrix

clf = MultinomialNB()
clf.fit(X_train_TFIDF, y_train)
predicted = clf.predict(X_test_TFIDF)

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("MultinomialNB Accuracy:", accuracy_score(y_test,
predicted))
print(f'\nConfusion Matrix: \n{confusion_matrix(y_test,
predicted)}')
print('==================================================
==\n')
```

Gambar 4. 27 Source Code Perhitungan performa Naive Bayes

Langkah berikutnya adalah menghitung nilai akurasi, *precision*, *recall*, dan *f1-score* menggunakan kode *Python* yang telah diimplementasikan pada Gambar Kode tersebut menghasilkan evaluasi performa model *Naive Bayes* untuk setiap kelas melalui ketiga metrik utama (*precision*, *recall*, dan *f1-score*).

Hasil pengukuran tersebut dapat diperoleh performa metode *Naive Bayes* untuk setiap kelas melalui evaluasi metrik *precision, recall,* dan *F1-score*. Nilai-nilai

yang dihasilkan berupa bilangan desimal dengan skala 0.0 hingga 1.0, dimana semakin mendekati angka 1 menunjukkan hasil yang semakin optimal untuk masing-masing metrik evaluasi tersebut (Zidan, 2022). Berikut hasil code program dalam proses evaluasi model ditunjukan pada gambar 4.30 :

```
MultinomialNB Precision: 0.9136865428478332
MultinomialNB Recall: 0.9139784946236559
MultinomialNB f1_score: 0.9135417150305487
MultinomialNB Accuracy: 0.9139784946236559

Confusion Matrix:
[[149  24]
 [ 16 276]]
=====================================================

              precision    recall  f1-score   support

     negatif       0.90      0.86      0.88       173
     positif       0.92      0.95      0.93       292

    accuracy                           0.91       465
   macro avg       0.91      0.90      0.91       465
weighted avg       0.91      0.91      0.91       465
```

Gambar 4. 28 Hasil Pengukuran Evaluasi Performa

Hasil evaluasi model menunjukkan bahwa nilai precision, recall, dan F1-score pada setiap kelas sentimen menunjukkan performa yang baik dalam model Naive Bayes. Akurasi menggambarkan kemampuan model dalam memprediksi secara benar baik sentimen positif maupun negatif dibandingkan dengan seluruh data. Berdasarkan hasil perhitungan menggunakan algoritma Naive Bayes,

diperoleh akurasi sebesar 91% yang mengindikasikan kemampuan klasifikasi yang baik secara keseluruhan. *Precision*, yang mengukur ketepatan model dalam mengklasifikasikan sentimen, menunjukkan nilai 90% untuk kelas negatif dan 92% untuk kelas positif. Hal ini mengindikasikan bahwa model sedikit lebih akurat (2%) dalam memprediksi sentimen positif dibandingkan negatif. Sementara itu, *recall* yang mengukur kemampuan model dalam mengidentifikasi seluruh sentimen yang relevan, mencapai 86% untuk kelas negatif dan 95% untuk kelas positif. Artinya, model lebih efektif dalam mendeteksi sentimen positif dibandingkan negatif, dengan selisih keberhasilan sebesar 9%. Hasil evaluasi model menunjukkan bahwa nilai *precision, recall, dan F1-score* pada setiap kelas sentimen dengan F1-score 88% untuk negatif dan 93% untuk positif mengkonfirmasi bahwa model Naive Bayes ini memiliki kemampuan klasifikasi yang baik secara keseluruhan.

J. Testing Prediksi Sentimen

Setelah menyelesaikan tahap klasifikasi dan penilaian kinerja model Naive Bayes, langkah berikutnya adalah menguji kemampuan prediksi label sentimen. Tujuan pengujian ini adalah untuk mengevaluasi sejauh mana model dapat mengenali dan memprediksi sentimen

dari ulasan yang diberikan pengguna. Sistem akan memproses setiap kalimat yang dimasukkan pengguna dan memberikan output berupa prediksi sentimen. Berikut code program prediksi sentimen :

```
# Input teks baru
new_text = ["Aplikasi shell bantu banget, dan pelayanan
sangat baik"]

new_text_tfidf = vectorizer.transform(new_text)
sentiment = clf.predict(new_text_tfidf)
print("Sentimen Prediksi:", sentiment[0])
```

Gambar 4. 29 Code program prediksi sentimen

Contoh pengujian dilakukan dengan menganalisis komentar "aplikasi shell sangat membantu, dan pelayanan sangat baik". Pada pengecekan manual, kalimat ini jelas tergolong positif, dan model berhasil memprediksi dengan tepat bahwa ini merupakan ulasan bernada positif, seperti terlihat dalam gambar 4.30

```
Sentimen Prediksi: positif
```

Gambar 4. 30 Hasil prediksi sentimen

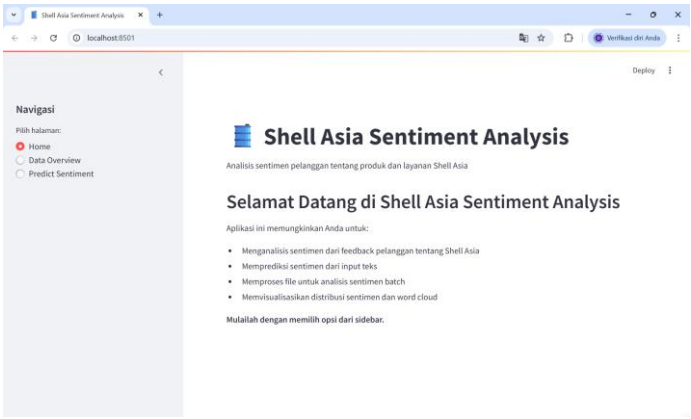
K. Visualisasi

Visualisasi hasil analisis sentimen tidak hanya ditampilkan dalam bentuk diagram batang dan *wordcloud* di *Google Colab*, tetapi juga diintegrasikan ke dalam sebuah website sederhana menggunakan *framework Streamlit*. Diagram batang digunakan untuk menampilkan distribusi persentase sentimen dan negatif sehingga

pembaca dapat langsung melihat dominasi respon pengguna aplikasi Shell Asia. Ukuran kata pada *word cloud* menunjukkan frekuensi kemunculannya semakin besar kata, semakin sering dimuncul pada sentimen positif dan juga sentimen negatif. Berikut tampilan dari website analisis sesntimen yang akan dijelaskan di bawah ini.

1. Halaman Home

   Tampilan menu utama aplikasi analisis sentimen Shell Asia dengan berbagai pilihan fitur, termasuk analisis data, pelatihan model, dan tampilan visual word cloud yang interaktif dalam antarmuka yang sederhana dan informatif. Berikut halaman home disajikan pada gambar 4.31.

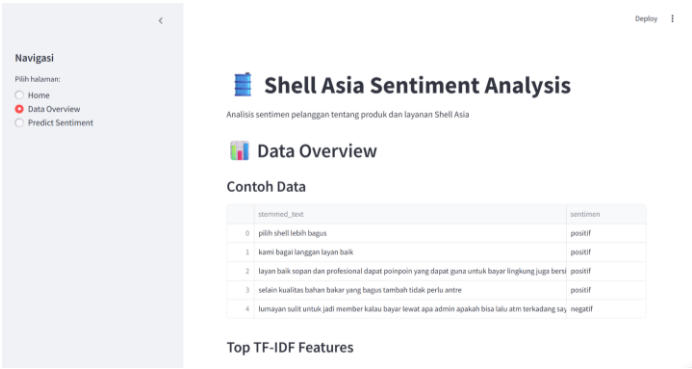


Gambar 4. 31 Halaman Home

2. Halaman Overview

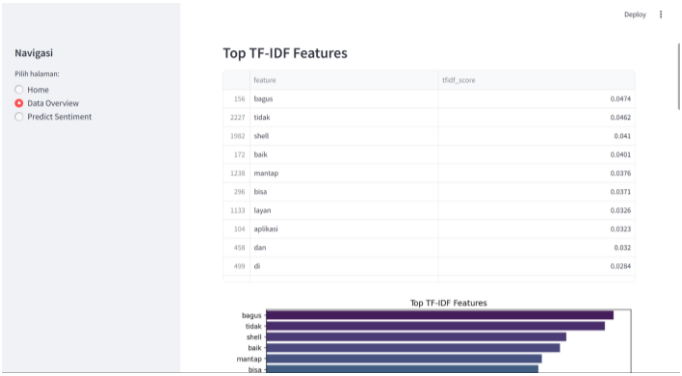
Pada menu Data Overview, sistem menampilkan contoh data teks yang telah diproses *(stemming)* dan fitur TF-IDF untuk analisis sentimen, termasuk 20 kata dengan skor TF-IDF tertinggi, serta visualisasi grafis, diagram batang dan juga *wordcloud*. Berikut halaman Overview disajikan pada gambar 4.32.

a. Tampilan Halaman Data Overview



Gambar 4. 32 Halaman Data Overview

b. Tabel untuk skor tertinggi dalam fitur TF-IDF



Gambar 4. 33 Tabel Top TF-IDF Features

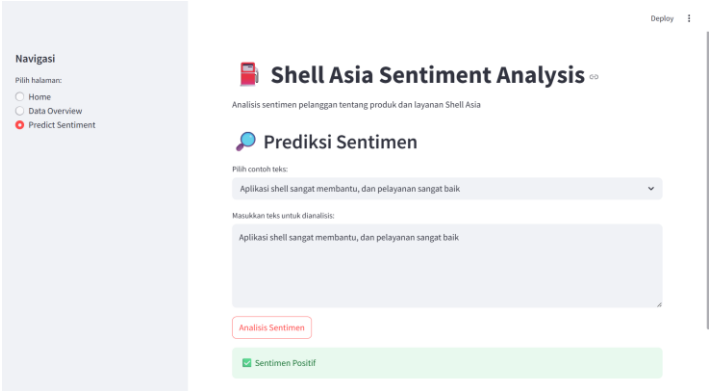
c. Visualisasi Top TF-IDF Features

Gambar 4. 34 Viasualisasi Top TF-IDF Features

d. Tampilan Confusion Matrix dan Matrix Performa

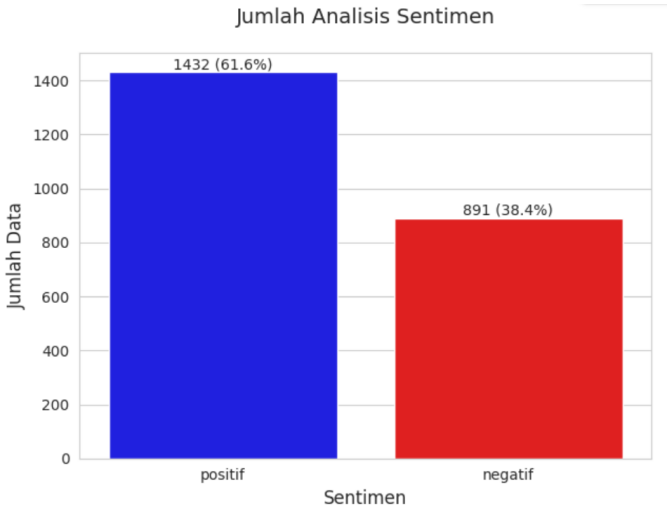
Gambar 4. 35 Confusion Matrix dan Performa Model

3. Tampilan ini untuk menganalisis sentimen teks secara instan, baik dengan memilih contoh teks yang telah disediakan maupun memasukkan teks secara manual. Sistem akan menampilkan hasil analisis berupa klasifikasi sentimen positif atau negatif. Berikut halaman prediksi disajikan pada gambar 4.33



Gambar 4. 36 Halaman Prediksi Sentimen

Diketauhi hasil akhir setelah dilakukan tahapan prepprocessing, pembobotan TF-IDF dan uji model menghasilkan data sebanyak 2.323 komentar bernilai negatif yaitu 891 (38.4%), komentar bernilai positif yaitu 1.432 (61.6%).



Gambar 4. 37 Jumlah Sentimen

Untuk mengetahui kata-kata yang paling banyak muncul pada pengguna aplikasi Shell Asia di *Google Play Store* yang bernilai sentimen positif seperti ‘shell’, ‘bagus’, ‘aplikasi’, ‘layan’ dan lain-lain. Untuk worldcloud dari sentiment positif ditunjukan ditunjukan pada gambar 4.32

Gambar 4. 38 Wordcloud pada sentimen positif

Selanjutnya untuk mengetahui kata-kata yang paling banyak muncul pada pengguna aplikasi Shell Asia di *Google Play Store* yang bernilai sentimen negatif seperti ‘aplikasi’, ‘tidak’, ‘poin’, ‘shell’ dan lain-lain. Untuk worldcloud dari sentiment negatif ditunjukan ditunjukan pada gambar 4.33. Visualisasi *wordcloud* memudahkan identifikasi kata kunci dominan yang mencerminkan sentimen pengguna, seperti kata bagus’ dan ‘layan’ yang mendominasi ulasan positif, atau 'tidak' dan 'poin' yang sering muncul dalam ulasan negatif. Pola ini membantu memahami aspek yang perlu ditingkatkan atau

dipertahankan dalam pengembangan aplikasi Shell Asia

Gambar 4. 39 Wordcloud pada sentimen negatif

BAB V
KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan penelitian yang dilakukan, maka dapat disimpulkan bahwa:

1. Pada pengimplemenntasian model *Naive Bayes* pada analisis sentimen ulasan pengguna aplikasi Shell Asia di *Google Play Store* memiliki hasil yang baik. Proses ini dimulai dengan pelabelan, dilanjutkan dengan preprocessing, Split Data 80:20, pembobotan menggunakan TF-IDF. Setelah itu, klasifikasi dilakukan dengan metode *Naive Bayes*, diakhiri dengan pengujian dan evaluasi. Data yang diambil pada ulasan pengguna aplikasi berjumlah 3.000 data, namun setelah dilakukan preprocessing dan pembobotan TF-IDF menjadi 2.323 data dengan rincian data yang memiliki sentiment positif sebanyak 1432 atau 61.6%, sedangkan data yang memiliki sentiment negatif terdapat 891 atau 38.4%.
2. Berdasarkan hasil evaluasi, algoritma Naive Bayes menunjukkan performa yang baik dengan tingkat akurasi sebesar 91%, presisi kelas negatif sebesar 90% dan kelas positif 92%, serta recall kelas

negatif 86% dan kelas positif 95%. Nilai F1-score untuk kelas negatif mencapai 88% dan kelas positif 93%. Secara keseluruhan, Naive Bayes dapat dipertimbangkan sebagai solusi klasifikasi dalam penelitian ini, terutama jika fokus utama adalah akurasi prediksi kelas positif.

B. Saran

Berdasarkan penelitian yang dilaksanakan, penulis berharap kepada peneliti selanjutnya untuk dapat dikembangkan dan terdapat beberapa saran-saran, yaitu:

1. Menggunakan metode klasifikasi yang berbeda seperti *Support Vector Machine (SVM), K-NN, Decision Tree* dengan begitu dapat membandingkan hasil performa yang dilakukan untuk mencari metode klasifikasi yang terbaik.
2. Pada penelitian ini, data diambil dari *Google Play Store* pada penelitian berikutnya diharapkan data dapat diperoleh dari media sosial lainnya seperti twitter atau X, youtube, facebook, instagram atau tik tok dll.

DAFTAR PUSTAKA

Andre Septian, J., Maulana Fahrudin, T., & Nugroho, A. (2019). Analisis Sentimen Pengguna Twitter Terhadap Polemik Persepakbolaan Indonesia Menggunakan Pembobotan TF-IDF dan K-Nearest Neighbor. *JOURNAL OF INTELLIGENT SYSTEMS AND COMPUTATION*, *1*(1), 43–49.

Aryanti, P. G., & Santoso, I. (2023). ANALISIS SENTIMEN PADA TWITTER TERHADAP MOBIL LISTRIK MENGGUNAKAN ALGORITMA NAIVE BAYES. *IKRA-ITH Informatika: Jurnal Komputer Dan Informatika*, *7*(2), 133–137. https://journals.upi-yai.ac.id/index.php/ikraith-informatika/issue/archive

Darwis, D., Siskawati, N., & Abidin, Z. (2021). Penerapan Algoritma Naive Bayes untuk Analisis Sentimen Review Data Twitter BMKG Nasional. *Jurnal TEKNO KOMPAK* , *15*(1), 131–145.

Fauziyyah, A. K., & Gautama, D. H. (2020). ANALISIS SENTIMEN PANDEMI COVID19 PADA STREAMING TWITTER DENGAN TEXT MINING PYTHON. *Jurnal Ilmiah SINUS*, *18*(2), 31. https://doi.org/10.30646/sinus.v18i2.491

Feldman, R., & Sanger, J. (2007). *The text mining handbook : advanced approaches in analyzing unstructured data*. Cambridge University Press.

Fikri, M. I., Sabrila, T. S., Azhar, Y., & Malang, U. M. (2020). Perbandingan Metode Naïve Bayes dan Support Vector Machine pada Analisis Sentimen Twitter. *SMATIKA Jurnal: STIKI Informatika Jurnal*, *10*(02), 71–76.

Fitri, E., Yuliani, Y., Rosyida, S., & Gata, W. (2020). Analisis Sentimen Terhadap Aplikasi Ruangguru Menggunakan Algoritma Naive Bayes, Random Forest Dan Support

Vector Machine. *JURNAL TRANSFORMTIKA*, *18*(1), 71–80. www.nusamandiri.ac.id,

Ghojogh, B., Samad, M. N., Mashhadi, S. A., Kapoor, T., Ali, W., Karray, F., & Crowley, M. (2019). *Feature Selection and Feature Extraction in Pattern Analysis: A Literature Review*. http://arxiv.org/abs/1905.02845

Hasibuan, E., & Heriyanto, E. A. (2022). ANALISIS SENTIMEN PADA ULASAN APLIKASI AMAZON SHOPPING DI GOOGLE PLAY STORE MENGGUNAKAN NAIVE BAYES CLASSIFIER. *JTS*, *1*(3), 13–24.

Ilayah, N. (2023). *Perbandingan Kinerja Metode Klasifikasi Naïve Bayes Dan Random Forest Dalam Analisis Sentimen Kasus Narkoba di Indonesia Pada Komentar YouTube*. UIN Walisongo Semarang.

Khder, M. A. (2021). Web scraping or web crawling: State of art, techniques, approaches and application. *International Journal of Advances in Soft Computing and Its Applications*, *13*(3), 144–168. https://doi.org/10.15849/ijasca.211128.11

Liu, B. (2016). Book Review Sentiment Analysis: Mining Opinions, Sentiments, and Emotions. *Computational Linguistics*. https://doi.org/10.1162/COLI

Luqman Adi, I. (2023). *ANALISIS SENTIMENT REVIEW APLIKASI MyPERTAMINA MENGGUNAKAN FASTTEXT DAN DECISION TREE*.

Maria, R., Umayah, R. U., Mahardinny, S., Kalana, D. N., & Saputra, D. D. (2023). Analisis Sentimen Persepsi Masyarakat Terhadap Penggunaan Aplikasi My Pertamina Pada Media Sosial Twitter Menggunakan Metode Naïve Bayes

Classifier. *Jurnal Komputer Antartika*, *1*. https://ejournal.mediaantartika.id/index.php/jka

Mas Pintoko, B., & Muslim L, K. (2018). *Analisis Sentimen Jasa Transportasi Online pada Twitter Menggunakan Metode Naïve Bayes Classifier*.

Maulidy, A. M., & Suharto, B. (2024). *Analisis Kampanye Digital dan Strategi Pemasaran PT Shell Indonesia di Pasar Ritel Bahan Bakar*. *9*(1). https://www.shell.co.id/

Ndapamuri, A. M., Manongga, D., & Iriani, A. (2023). Analisis Sentimen Ulasan Aplikasi Tripadvisor Dengan Metode Support Vector Machine, K-Nearest Neighbor, Dan Naive Bayes. *JURNAL INOVTEK POLBENG - SERI INFORMATIKA*, *8*(1), 127–140.

Putri Santoso, D., & 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).

Rachman Chajannah, D. A. D., Goejantoro, R., & Tisna Amija, F. D. (2020). Implementasi Text Mining Pengelompokkan Dokumen Skripsi Menggunakan Metode K-Means Clustering Implementation Of Text Mining For Grouping Thesis Documents Using K-Means Clustering. *Jurnal EKSPONENSIAL*, *11*(2), 167–174.

Rivki, M., & Mukharil Bachtiar, A. (2017). IMPLEMENTASI ALGORITMA K-NEAREST NEIGHBOR DALAM PENGKLASIFIKASIAN FOLLOWER TWITTER YANG MENGGUNAKAN BAHASA INDONESIA. *Jurnal Sistem Informasi (Journal of Information System)*, *13*(1). https://doi.org/10.21609/jsi.v13i1.500

Riziq sirfatullah Alfarizi, M., Zidan Al-farish, M., Taufiqurrahman, M., Ardiansah, G., & Elgar, M. (2023). PENGGUNAAN PYTHON SEBAGAI BAHASA PEMROGRAMAN UNTUK MACHINE LEARNING DAN DEEP LEARNING. In *Karimah Tauhid* (Vol. 2, Issue 1).

Shell Indonesia Luncurkan Program Loyalitas Berbasis Aplikasi Shell Go+. (2022, February 22). Shell Indonesia. https://www.shell.co.id/in_id/ruang-media/news-and-media-releases/news-2022/shell-indonesia-luncurkan-program-loyalitas-berbasis-aplikasi-shell-go.html

Shell Indonesia meluncurkan Shell Flagship pertama sebagai one-stop destination. (2024, January 24). Shell Indonesia. https://www.shell.co.id/in_id/ruang-media/news-and-media-releases/news-2024/shell-indonesia-flagship-pertama-one-stop-destination.html

Shell Tutup 9 SPBU di Sumut Mulai 1 Juni, Ini Alasannya. (2024, May 31). CNN Indonesia. https://www.cnnindonesia.com/ekonomi/20240531165554-85-1104363/shell-tutup-9-spbu-di-sumut-mulai-1-juni-ini-alasannya

Sujadi, H., Fajar, S., & Roni, C. (2022). ANALISIS SENTIMEN PENGGUNA MEDIA SOSIAL TWITTER TERHADAP WABAH COVID-19 DENGAN METODE NAIVE BAYES CLASSIFIER DAN SUPPORT VECTOR MACHINE. *INFOTECH Journal*, *8*(1), 22–27. https://doi.org/10.31949/infotech.v8i1.1883

Sujjada, A., Nurfazri Novianti, J., & Griha Tofik Isa, I. (2023). ANALISIS SENTIMEN TERHADAP REVIEW BANK DIGITAL PADA GOOGLE PLAY STORE MENGGUNAKAN METODE SUPPORT VECTOR MACHINE (SVM). *Jurnal Rekayasa*

Teknologi Nusa Putra, *9*(2), 122–135. https://rekayasa.nusaputra.ac.id/index

Suryono, S., Utami, E., & Luthfi, E. T. (2018). KLASIFIKASI SENTIM EN PADA TW ITTER DENGAN NAIVE BAYES CLASSIFIER. *Angkasa, Jurnal Ilmiah Bidang Teknologi*, *10*(1), 89–95.

Tripathy, A., Agrawal, A., & Rath, S. K. (2015). Classification of Sentimental Reviews Using Machine Learning Techniques. *Procedia Computer Science*, *57*, 821–829. https://doi.org/10.1016/j.procs.2015.07.523

Wenty Dwi Yuniarti. (2019). *Dasar-Dasar Pemograman dengan Python*.

Wibawa, A. P., Guntur, M., Purnama, A., Fathony Akbar, M., & Dwiyanto, F. A. (2018). Metode-metode Klasifikasi. *Prosiding Seminar Ilmu Komputer Dan Teknologi Informasi*, *3*(1).

Yoga Pratama, A., Umaidah, Y., Voutama, A., Informatika, T., Ilmu Komputer, F., Singaperbangsa Karawang Ds Paseurjaya, U., Telukjambe Timur, K., Karawang, K., & Barat, J. (2021). Analisis Sentimen Media Sosial Twitter Dengan Algoritma K-Nearest Neighbor Dan Seleksi Fitur Chi-Square (Kasus Omnibus Law Cipta Kerja). *Jurnal Sains Komputer & Informatika (J-SAKTI)*, *5*(2), 897–910.

Yutika, C. H., Adiwijaya, A., & Faraby, S. Al. (2021). Analisis Sentimen Berbasis Aspek pada Review Female Daily Menggunakan TF-IDF dan Naïve Bayes. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, *5*(2), 422. https://doi.org/10.30865/mib.v5i2.2845

Zain, M. R. A. F., & Kamayani, M. (2023). Analisis Sentimen Ulasan Pelanggan Online Ubi Madu Cilembu Abah Nana

Menggunakan Algoritma Naïve Bayes. *Jurnal Sistem Komputer Dan Informatika (JSON)*, *5*(1), 11–21. https://doi.org/10.30865/json.v5i1.6646

Zidan, M. (2022). *Analisis Sentimen Kenaikan Harga Bahan Bakar Minyak (BBM) Berdasarkan Respon Pengguna Media Sosial Twitter Di Indonesia Menggunakan Metode Naive Bayes*. UNIVERSITAS ISLAM NEGERI WALISONGO SEMARANG.

DAFTAR LAMPIRAN

Lampiran 1: Lembar Bimbingan Proposal

PERSETUJUAN PEMBIMBING

Proposal Skripsi ini telah disetujui oleh Pembimbing untuk dilaksanakan. Disetujui pada:

Hari : Jum'at

Tanggal : 20 Desember 2024

Pembimbing I	Pembimbing II
Dr. Wenty Dwi Yuniarti S.Pd., M.Kom	Mokhamad Iklil Mustofa M.Kom.
NIP. 19770622200604 2 005	NIP. 19880807201903 1 010

Mengetahui,

Ketua Jurusan Teknologi Informasi

Dr. Khothibul Umam S.T., M. Kom.

NIP. 19790827 201101 1 007

Lampiran 2: Lembar Pengesahan Proposal

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

PENGESAHAN UJIAN KOMPREHENSIF

Naskah proposal skripsi berikut ini:

Judul	:	Analisis Sentimen pada Ulasan Aplikasi Shell Asia Di Google Play Store Menggunakan Metode Niave Bayes
Penulis	:	Desmi Salsabila
NIM	:	2108096028
Jurusan	:	Teknologi Informasi

Telah diuji dalam seminar proposal skipsi oleh Dewan Penguji Fakultas Sains dan Teknologi UIN Walisongo Semarang dan dapat diterima sebagai salah satu syarat memperoleh gelar sarjana dalam bidang Ilmu Teknologi Informasi.

Semarang, 5 Febuari 2024

DEWAN PENGUJI

Penguji I

Dr. Khotibul Umam, M.Kom
NIP. 197908272011011007

Penguji II

Dr. Wenty Dwi Yuniarti, S.Pd.,M.Kom.
NIP. 197706272006042005

Penguji III

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

Penguji IV

Siti Nurâ€™aini, M.Kom.
NIP. 198401312018012001

Lampiran 3: Contoh Dokumen Hasil Scraping Data Ulasan Aplikasi Shell Asia

No.	Data Ulasan Aplikasi Shell Asia
1.	Pilih Shell, lebih bagus!
2	Kami sebagai pelanggan ,pelayanannya baik
3	Pelayanan baik, sopan, dan profesional. Terdapat poin-poin yang dapat digunakan untuk pembayaran. Lingkungannya juga bersih
4	Selain kualitas bahan bakarnya yang bagus, ditambah tidak perlu antre. ÃƒÂ°Ã…Â¸Ã¢â‚¬Ëœ,Â ÃƒÂ°Ã…Â¸Ã¢â‚¬Ëœ,Â ÃƒÂ°Ã…Â¸Ã¢â‚¬Ëœ,Â
5	Agak sulit untuk menjadi member. Kalau pembayaran lewat apa, Min? Apakah bisa melalui ATM? Terkadang saya hanya membawa uang tunai ketika membeli di Shell. Saya juga sering ditanya, 'Punya member atau tidak?' tapi saya belum tahu cara daftarnya serta metode pembayarannya
6	Spbu Shell pelayanan nya memuaskan
7	Kualitas sangat bagus, takaran juga pas. Sayangnya, belum tersedia di daerah saya, hanya ada di kota-kota besar saja.
8	Kenapa tidak bisa daftar? Mau generate OTP tapi ada error.
9	Setiap kali saya mau menukar voucher, selalu tidak bisa. Tulisannya kadaluarsa terus, padahal belum dibuka. Karyawan Shell juga tidak membantu.

....

2996	Ok
2997	Pelayanan bagus
2998	Mantap
2999	Ok mudah dan cepat
3000	nyaman aman

Lampiran 4: Contoh Dokumen yang Sudah Diberi Label

No.	Data Ulasan Aplikasi Shell Asia	Sentimen
1.	Pilih Shell, lebih bagus!	Positif
2	Kami sebagai pelanggan ,pelayanannya baik	Positif
3	Pelayanan baik, sopan, dan profesional. Terdapat poin-poin yang dapat digunakan untuk pembayaran. Lingkungannya juga bersih	Positif
4	Selain kualitas bahan bakarnya yang bagus, ditambah tidak perlu antre. ÃƒÂ°Ã…Â¸Ã¢â‚¬ËœÃ‚Â ÃƒÂ°Ã…Â¸Ã¢â‚¬Ëœ ÃƒÂ°Ã…Â¸Ã¢â‚¬ËœÃ‚Â	Positif
5	Agak sulit untuk menjadi member. Kalau pembayaran lewat apa, Min? Apakah bisa melalui ATM? Terkadang saya hanya membawa uang tunai ketika membeli di Shell. Saya juga sering ditanya, 'Punya member atau tidak?' tapi saya belum tahu cara daftarnya serta metode pembayarannya	Negatif
6	Spbu Shell pelayanan nya memuaskan	Positif
....		
2996	Ok	Positif
2997	Pelayanan bagus	Positif
2998	Mantap	Positif
2999	Ok mudah dan cepat	Positif
3000	nyaman aman	Positif

Lampiran 5: Source Code Scrapping Data

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

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

```
[ ] from google_play_scraper import Sort, reviews
    result, continuation_token = reviews(
        'com.shell.sitibv.shellgoplusindia',
        lang='id',
        country='id',
        sort=Sort.MOST_RELEVANT,
        count=3000,
        filter_score_with=None
    )
```

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

```
df_busu[['userName', 'score','at', 'content']].head()
```

```
[ ] my_df = sorted_df[['userName', 'score','at', 'content']]
```

```
[ ] my_df.head()
```

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

Lampiran 6 : Source Code Preprocessing dan Model

```
[1]  #Upload file scraping
     from google.colab import files
     uploaded = files.upload()
```

```
[2]  #Import libabry
     import pandas as pd
     import re
     import numpy as np
```

```
     ## Membaca file csv dan menampilakn di dataframe
     df = pd.read_csv("Scrapped_Data_Shellasia.csv", encoding='latin1')
     df
```

```
[4]  #Remove Duplicate
     df = df.drop_duplicates(subset=['content'])
```

```
[5]  df.shape
```

```
[6]  df = df[['content', 'sentimen']]
     df
```

```
     #Cleaning Data
     import re
     def clean_text(text):
         number_pattern = r'\d+'
         text = re.sub(r'#\w+', '', text)
         text = re.sub(r'https?://\S+', '', text)
         text = re.sub(number_pattern, '', text)
         text = re.sub(r'(?<=\w)\.(?=\w)', ' ', text)
         text = re.sub(r'[^\w\s]', '', text)
         text = re.sub(r'\s+', ' ', text)
```

```
     #Case Folding
         text = text.lower()
         return text
     def replaceTOM(text):
         pola = re.compile(r'(.)\1{2,}', re.DOTALL)
         return pola.sub(r'\1', text)
     def clear_emoji(text):
         return text.encode('ascii', 'ignore').decode('ascii')
     def preprocess_text(text):
         text = replaceTOM(text)
         text = clear_emoji(text)
         text = clean_text(text)
         return text
     df['clean_text'] = df['content'].apply(preprocess_text)
     df
```

```
#Normalisasi Data
norm = {
    'mantapp': 'mantap', 'agx': 'lumayan', 'bang': 'abang', 'apk': 'aplikasi', 'dpet': 'dapat',
    'okegas': 'lanjut', 'x': 'kali', 'telp': 'telepon', 'tp': 'tapi', 'banyqk': 'banyak', 'manta
    'emng': 'memang', 'lgi': 'lagi', 'bgt': 'banget', 'oc': 'bagus', 'kwalitas': 'kualitas', 'ka
    'markotop': 'baik', 'naek': 'naik', 'markotop': 'sekali', 'lg': 'lagi', 'trims': 'terimakasi
    'sprt': 'seperti', 'kmrin': 'kemarin', 'mantaaap': 'mantap', 'apps': 'aplikasi', 'gw': 'saya
    'regis': 'registrasi', 'komplen': 'komplain', 'gua': 'saya', 'klo': 'kalo', 'gnti': 'ganti',
    'sblmnya': 'sebelumnya', 'thank': 'thanks', 'mempernudah': 'mempermudah', 'sheel': 'shell',
    'skalian': 'sekalian', 'uninstal': 'uninstall', 'apl': 'aplikasi', 'ngk': 'engga', 'gtu': 'g
    'voucernya': 'vouchernya', 'gimanaaa': 'gimana', 'bangke': 'bangkai', 'hanguuuuuuuus': 'hang
    'maximal': 'maksimal', 'gmna': 'gimana', 'ama': 'sama', 'prtmn': 'pertamina', 'mnta': 'minta
    'syg': 'sayang', 'kacaw': 'kecewa', 'nomkr': 'nomor', 'sllu': 'selalu', 'dafra': 'daftar', '
    'tlpn': 'telepon', 'tgl': 'tanggal', 'blm': 'belum', 'iriiiiiit': 'irit', 'nga': 'tidak', 's
    'mantep': 'mantap', 'job': 'sekali', 'y': 'ya', 'nie': 'ini', 'ga': 'tidak', 'kyk': 'kaya',
     'dpt': 'dapat', 'rp': 'rupiah', 'eror': 'error', 'kl': 'kalau', 'jl': 'jalan', 'aplikas': '
    }
def normalisasi(str_text):
    for key, value in norm.items():
        str_text = re.sub(rf'\b{key}\b', value, str_text)
    return str_text

df['normalisasi_text'] = df['clean_text'].apply(lambda x: normalisasi(x))
df
```

```
#Tokenize
import nltk.corpus
nltk.download('punkt')
from nltk.tokenize import sent_tokenize, word_tokenize
df['tokenized_text'] = df['normalisasi_text'].apply(lambda x: x.split())
df
```

```
!pip install Sastrawi swifter
```

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

def stemmed_wrapper(term):
    return stemmer.stem(term)
term_dict = {}
hitung=0

for document in df['tokenized_text']:
    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]
df['stemmed_text'] = df['tokenized_text'].apply(lambda x:' '.join(get_stemmed_term(x)))
df
```

```
[ ]  #Menyimpan file yang sudah di preprocessing
     df.to_csv("StemmingData-Hasil.csv" ,index=False)
```

```
     #Mengimport library yang dibutuhkan
     import re
     import pandas as pd
     import seaborn as sns
     import matplotlib.pyplot as plt
```

```
[ ]  df = pd.read_csv("StemmingData-Hasil.csv")
     df
```

```
     df = df[['stemmed_text', 'sentimen']]
     df
```

```
     #Mengecek baris yang bernilai NaN
     df.isnull().sum()
```

```
     # Menghapus baris dengan NaN
     df = df.dropna(subset=['stemmed_text'])
```

```
[ ]  X = df['stemmed_text']
     y = df['sentimen']
```

```
[ ]  #Tahapan split validation
     from sklearn.model_selection import train_test_split
     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
     print(X_train.shape)
     print(y_train.shape)
     print(X_test.shape)
     print(y_test.shape)
```

```
     #Tahapan merubah data menjadi vector
     from sklearn.feature_extraction.text import TfidfVectorizer
     vectorizer = TfidfVectorizer()
     vectorizer.fit(X_train)
```

```
[ ]  #Tahapan TF-IDF
     X_train_TFIDF = vectorizer.transform(X_train).toarray()
     X_test_TFIDF = vectorizer.transform(X_test).toarray()
     X_all_TFIDF = vectorizer.transform(df["stemmed_text"]).toarray()
     kolom = vectorizer.get_feature_names_out()
     train_tf_idf = pd.DataFrame(X_train_TFIDF, columns=kolom)
     test_tf_idf = pd.DataFrame(X_test_TFIDF, columns=kolom)
     train_tf_idf
```

```
# Tahap perhitungan model Naive Bayes
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.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix

clf = MultinomialNB()
clf.fit(X_train_TFIDF, y_train)
predicted = clf.predict(X_test_TFIDF)

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("MultinomialNB Accuracy:", accuracy_score(y_test, predicted))
print(f'\nConfusion Matrix: \n{confusion_matrix(y_test, predicted)}')
print('=======================================================\n')
print(classification_report(y_test, predicted, zero_division=0))
```

```
[ ]  # Input teks baru
     new_text = ["Aplikasi shell sangat membantu, dan pelayanan sangat baik"]

     new_text_tfidf = vectorizer.transform(new_text)
     sentiment = clf.predict(new_text_tfidf)
     print("Sentimen Prediksi:", sentiment[0])
```

```
#Diagram Batang
data = df
# Menghapus nilai kosong jika ada
data = data.dropna(subset=['sentimen'])
# Menghitung jumlah setiap kelas sentimen
sentiment_count = data['sentimen'].value_counts()
# Menghitung persentase
total_count = sentiment_count.sum()
sentiment_percentage = (sentiment_count / total_count) * 100
# Mengatur gaya plot
sns.set_style('whitegrid')
# Membuat daftar warna berdasarkan kelas sentimen
colors = ['blue', 'red']
# Membuat plot
fig, ax = plt.subplots(figsize=(7, 5))
ax = sns.barplot(x=sentiment_count.index, y=sentiment_count.values, palette=colors)

# Menambahkan judul dan label sumbu
plt.title('Jumlah Analisis Sentimen', fontsize=14, pad=20)
plt.xlabel('Sentimen', fontsize=12)
plt.ylabel('Jumlah Data', fontsize=12)
# Menambahkan label di atas setiap bar (jumlah dan persentase)
for i, (count, percentage) in enumerate(zip(sentiment_count.values, sentiment_percentage)):
    ax.text(i, count + 0.10, f'{count} ({percentage:.1f}%)', ha='center', va='bottom')
plt.show()
```

```python
from wordcloud import WordCloud
import matplotlib.pyplot as plt
# Filter dataframe untuk hanya memuat keseluruhan text
positive_text = ' '.join(df['stemmed_text'].tolist())
# Buat objek WordCloud
wordcloud = WordCloud(width=800, height=400, background_color='white').generate(positive_text)
# Tampilkan visualisasi kata-kata yang paling sering muncul dalam pelabelan positif
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
# Menambahkan tulisan "Sentimen Positif" di atas WordCloud
plt.title('Sentimen Analisis', fontsize=16, pad=20)
plt.show()
```

```python
from wordcloud import WordCloud
import matplotlib.pyplot as plt

# Filter dataframe untuk hanya memuat teks dengan pelabelan positif
positive_text = ' '.join(df[df['sentimen'] == 'positif']['stemmed_text'].tolist())

# Buat objek WordCloud
wordcloud = WordCloud(width=800, height=400, background_color='white').generate(positive_text)

# Tampilkan visualisasi kata-kata yang paling sering muncul dalam pelabelan positif
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')

# Menambahkan tulisan "Sentimen Positif" di atas WordCloud
plt.title('Sentimen Positif', fontsize=16, pad=20)
plt.show()
```

```python
from wordcloud import WordCloud
import matplotlib.pyplot as plt

# Filter dataframe untuk hanya memuat teks dengan pelabelan negatif
negative_text = ' '.join(df[df['sentimen'] == 'negatif']['stemmed_text'].tolist())

# Buat objek WordCloud
wordcloud = WordCloud(width=800, height=400, background_color='white').generate(negative_text)

# Tampilkan visualisasi kata-kata yang paling sering muncul dalam pelabelan negatif
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')

# Menambahkan tulisan "Sentimen Positif" di atas WordCloud
plt.title('Sentimen Negatif', fontsize=16, pad=20)
plt.show()
```

Lembar 7: Daftar Riwayat Hidup

A. Identitas Diri

Nama Lengkap	:	Desmi Salsabila
Tempat & Tanggal Lahir	:	Jakarta, 21 Desember 2002
Alamat Rumah	:	Jl. Danau Tondano Gg. Sekolah No. 5 Rt/Rw. 020/004 Kel. Bendungan Hilir Kec. Tanah Abang Jakarta Pusat
No. HP	:	085747265157
E-Mail	:	desmisalsa12@gmail.com

B. Riwayat Pendidikan

Pendidikan Fromal

a. Sekolah Dasar Negeri 05 Bendungan Hilir

b. Sekolah Menengah Pertama Negeri 40 Jakarta

c. Sekolag Menengah Atas Negeri 1 Weru

Semarang, 23 Juni 2025

Desmi Salsabila
NIM. 2108096028