

Analisis Sentimen Masyarakat Terhadap Sistem Pembayaran MyPertamina dengan Metode *Random Forest*, SVM, dan Naïve Bayes

Ayu Amelia^a, Lilis Nur Hayati^b, Herdianti Darwis^c

Universitas Muslim Indonesia, Makassar, Indonesia

^a13120190004@umi.ac.id; ^blilis.nurhayati@umi.ac.id; ^cherdianti.darwis@umi.ac.id

Received: 23-02-2024 | Revised: 08-03-2024 | Accepted: 15-03-2024 | Published: 29-03-2024

Abstrak

PT. Pertamina (PERSERO), sebagai perusahaan BUMN terkemuka di Indonesia di bidang perminyakan, memiliki peran vital dalam pengolahan dan pemasaran minyak bumi, terutama bahan bakar minyak (BBM). Penelitian ini menerapkan tiga metode analisis sentimen yaitu Random Forest, SVM, dan Naïve Bayes untuk mengevaluasi ulasan pengguna terhadap aplikasi MyPertamina. Dengan mengumpulkan data melalui web scraping dari Google Play Store sebanyak 3360 ulasan dianalisis dari 2018 hingga 01 Desember 2023. klasifikasi sentimen terbagi menjadi tiga kategori: positif, negatif, dan netral. Penggunaan Google Colab sebagai alat utama dalam pengolahan data dan implementasi model klasifikasi menawarkan efisiensi dalam eksperimen. Hasil evaluasi menunjukkan bahwa ketiga metode analisis sentimen Random Forest, SVM, dan Naïve Bayes mencapai akurasi tinggi pada evaluasi ulasan aplikasi MyPertamina. Random Forest menonjol dengan akurasi 99.77%, sementara SVM dan Naïve Bayes juga memberikan performa yang baik, masing-masing mencapai 99.31% dan 90.24%. Nilai Precision, Recall, dan F1-Score yang optimal pada ketiga metode mengindikasikan keefektifan mereka dalam menganalisis sentimen ulasan pengguna.

Kata kunci: MyPertamina, *Random Forest*, SVM, *Naïve Bayes*, *Google Play Store*.

Pendahuluan

PT.Pertamina (PERSERO) adalah salah satu perusahaan BUMN terbesar di Indonesia. Perusahaan milik negara itu beroperasi di industri perminyakan. Pertamina (PERSERO) bergerak dalam bidang usaha pengolahan dan pemasaran minyak bumi dari hulu ke hilir. Salah satu hasil olahan pertamina adalah bahan bakar minyak (BBM) [1]. BBM merupakan komoditas yang memegang peranan vital dalam semua aktivitas ekonomi. BBM juga menentukan faktor penting dalam menentukan perubahan harga bahan pokok atau inflasi di suatu negara [2]. Pemerintah melihat bisnis minyak dan gas alam sangat berperan penting dalam menghasilkan masukan yang nyata untuk pertumbuhan ekonomi Nasional BBM yang dikelola oleh pemerintah indonesia dengan UUD No. 22 tahun 2001 tentang minyak dan gas. Pemerintah mencoba pembelian bahan bakar minyak (BBM) bersubsidi yaitu pertalite dan solar dengan mengeluarkan program MyPertamina. Dengan tujuan agar penyaluran pertalite dan solar tepat sasaran agar nantinya aplikasi MyPertamina bisa membantu memudahkan kinerja pemerintah [3].

Aplikasi merupakan perangkat lunak yang dikembangkan untuk melakukan tugas tertentu yang bertujuan untuk memudahkan kinerja atau transaksi manusia dengan manusia lainnya. Mobile Payment merupakan hasil dari perkembangan teknologi, dimana mobile payment diciptakan untuk proses pembayaran dari tunai menjadi non tunai dengan menggunakan perangkat handphone atau smartphone pembayarannya menggunakan QR Code (quick response code) OTP (one time password). dan NFC (near field communication) [4]. Efek dari kemajuan teknologi sangat berdampak bagi kehidupan manusia, yang dari awalnya semua aktivitas transaksi dilakukan secara antarmuka, kini sudah bisa dilakukan dengan jarak yang jauh dan kapanpun selagi semua akses internet terhubung dari satu dengan yang lainnya.

MyPertamina adalah aplikasi layanan digital yang dikembangkan oleh perusahaan pertamina, melalui aplikasi MyPertamina diharapkan memudahkan masyarakat dalam pembelian BBM di Stasiun Pengisian Bahan Bakar Umum (SPBU) secara *cashless*. Aplikasi MyPertamina dirancang untuk membantu kebutuhan masyarakat agar transaksi menjadi lebih mudah [5]. Peluncuran atau uji coba aplikasi MyPertamina pertama kali pada tanggal 1 juli 2022 di beberapa wilayah dan kabupaten di Indonesia. Adapun wilayah dan kabupaten yang termasuk dalam uji coba aplikasi My Pertamina yaitu : Kota Bukittinggi, Kabupaten Agam, Kota padang panjang, Kabupaten Tanah Datar, Kota Banjarmasin, kota bandung, Kota Tasikmalaya, Kabupaten Ciamis, Kota Manado, Kota Yogyakarta dan Kota Sukabumi [6]. Sulawesi selatan resmi menerapkan program subsidi tepat untuk BBM jenis solar diseluruh SPBU pada tanggal 21 Maret 2023. Kendaraan roda 4 atau lebih wajib menggunakan *QR-Code* aplikasi mypertamina. Kendaraan roda empat yang terdaftar memiliki kuota 60 liter

BBM bersubsidi. Kemudian untuk kendaraan roda empat barang memiliki kuota 80 liter dan 200 liter untuk roda enam ke atas setiap harinya. Sementara untuk kendaraan yang belum terdaftar, maksimal akan mendapatkan 31 kuota hanya 20 liter perhari. Meskipun telah diwajibkan hingga saat ini masih banyak masyarakat yang tidak menggunakan aplikasi mypertamina dikarenakan masyarakat masih lebih memilih menggunakan pembayaran tunai dibandingkan *cashless* [7].

Berdasarkan penilaian yang dilakukan oleh pengguna aplikasi MyPertamina didasarkan dari pengalaman pengguna dalam menggunakan aplikasi MyPertamina belum memuaskan. Dilihat dari rating pada *google play store* yang memiliki rentang 1.0 sampai dengan 5.0, rating yang didapatkan aplikasi MyPertamina adalah 3.1 dengan ulasan 3360 ribu, rating tersebut didapatkan berdasarkan review dari pengalaman pengguna aplikasi MyPertamina.

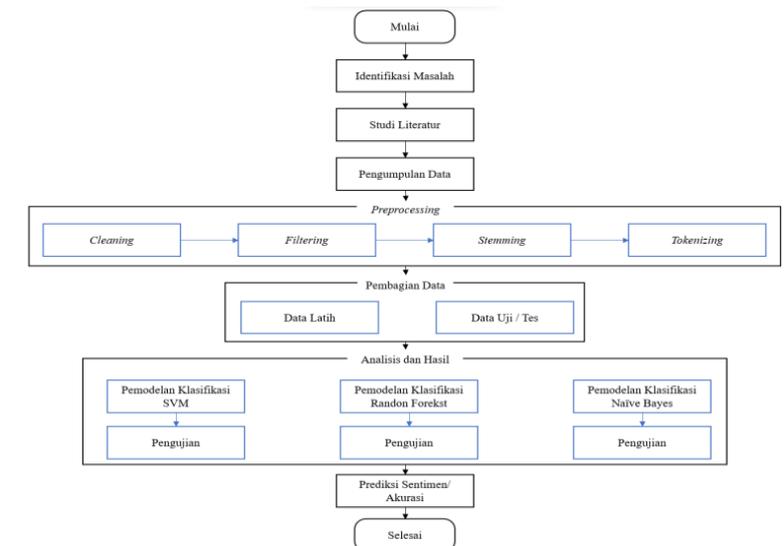
Ada beberapa kendala yang dirasakan pengguna aplikasi MyPertamina adalah sering error pada aplikasi, sering terjadi *bug* yang kemudian membuat pengguna menjadi malas menggunakan aplikasi MyPertamina dan juga kendalanya adalah petugas di lapangan kurang memahami penerapan penggunaan aplikasi MyPertamina [5]. Selain itu berdasarkan hasil wawancara secara langsung oleh konsumen SPBU Pertamina Racing Kota Makassar banyak yang lebih memilih membayar tunai dibandingkan *cashless* atau menggunakan aplikasi MyPertamina. Opini dan sentiment pengguna *review* aplikasi MyPertamina sangat membantu sebagai evaluasi program MyPertamina agar nantinya lebih baik dari sebelumnya, dikarenakan program MyPertamina harus mempertimbangkan berbagai masukan, diantaranya dengan melihat respon dan opini masyarakat.

Analisis sentimen adalah pendapat atau opini seseorang terhadap suatu barang atau objek yang biasanya hasil dari pendapat itu bisa berupa pendapat positif, negatif ataupun netral. Analisis sentiment bisa diartikan sebagai pendapat dari seseorang atau lebih terkait objek atau topik. Tujuannya agar nantinya bisa dijadikan sebagai keputusan untuk mengetahui apakah seseorang suka atau tidak suka dengan suatu objek. Biasanya nilai yang diberikan terkait analisis sentiment adalah negatif, netral dan positif [8]. Terdapat dua cara pendekatan dalam melakukan *sentiment analysis*, yaitu dengan *lexicon-based* (pendekatan berbasis leksikal) dan *learning-based* (pendekatan dengan machine learning) [9]. Pendekatan *lexicon-based* bergantung pada kamus opini (*lexicon*) untuk melakukan klasifikasi sedangkan *learning-based* menggunakan dataset yang telah diklasifikasi secara manual sebelumnya untuk sebagai data latih untuk memperoleh klasifikasi teks opini secara otomatis [10]. Model atau metode klasifikasi seperti *Naïve Bayes* [11], *Random Forest* [12], dan *Support Vectors Machine* namun diketahui bahwa metode *Random Forest* memiliki performa terbaik dan cocok digunakan untuk memproses data dalam jumlah besar.

Berdasarkan latar belakang tersebut, terdapat beberapa rumusan masalah, yakni bagaimana menganalisis sentimen masyarakat terhadap sistem pembayaran aplikasi MyPertamina dan implementasi metode *Random Forest*, SVM, dan *Naïve Bayes* dalam mengklasifikasi sentimen masyarakat terhadap penerapan aplikasi MyPertamina. Tujuan penelitian mencakup pemahaman terhadap sentimen positif, netral, dan negatif dalam ulasan masyarakat terkait sistem pembayaran pada aplikasi MyPertamina, serta mengklasifikasikan sentimen masyarakat terhadap aplikasi tersebut.

Metode

Gambar 1 menunjukkan tahapan penelitian dalam pengolahan data dan analisis sentimen, dimulai dari identifikasi masalah, studi literatur, dan pengumpulan data. Proses berikutnya adalah *preprocessing* yang meliputi *cleaning*, *filtering*, *stemming*, dan *tokenizing* data sebelum dibagi menjadi data latih dan data uji. Setelah pembagian data, dilakukan pemodelan klasifikasi menggunakan tiga metode berbeda yaitu *Support Vector Machine*, *Random Forest*, dan *Naïve Bayes*, yang masing-masing diuji keakuratannya. Pengujian ini bertujuan untuk menemukan model dengan prediksi sentimen atau akurasi terbaik. Proses diakhiri dengan prediksi sentimen/akurasi dan konfirmasi keberhasilan dari model klasifikasi yang telah dipilih.



Gambar 1. Tahapan Penelitian

A. Identifikasi Masalah

Melalui analisis komentar pengguna aplikasi MyPertamina di *Google Play Store*, penelitian ini bertujuan untuk mengidentifikasi permasalahan yang muncul dalam aplikasi tersebut. Fokus pada klasifikasi komentar menggunakan metode SVM, *Random Forest*, dan *Naive Bayes* untuk memahami variasi sentimen dan isu yang diungkapkan oleh pengguna. Tujuan akhirnya adalah mengukur akurasi dari masing-masing metode klasifikasi dengan menggunakan bahasa pemrograman *Python*, memberikan wawasan mendalam tentang sejauh mana aplikasi dapat memenuhi atau mengecewakan harapan pengguna berdasarkan *feedback* yang diberikan. Melalui pendekatan ini, ditemukan rekomendasi solusi meningkatkan kualitas dan kepuasan pengguna terhadap aplikasi MyPertamina.

B. Studi Literatur

Tahapan ini bertujuan untuk merinci teori-teori yang akan digunakan dalam analisis sentimen ulasan MyPertamina, dengan fokus pada metode klasifikasi seperti metode SVM, Random Forest, dan Naive Bayes. Pendekatan SVM, yang mirip dengan mencari jalur terbaik dalam labirin opini, akan digabungkan dengan keberagaman pohon keputusan dari Random Forest dan pendekatan probabilitas Naive Bayes. Dengan membangun landasan teoritis yang kokoh, penelitian ini tidak hanya mencari solusi teknis, tetapi juga menetapkan dasar referensi yang kuat bagi peneliti dalam menghadapi dinamika analisis sentimen di lingkungan digital. Diharapkan, hasil penelitian ini akan memberikan pemahaman mendalam tentang pandangan dan evaluasi pengguna terhadap MyPertamina, membuka jalan bagi perbaikan dan peningkatan yang lebih baik bagi aplikasi tersebut.

C. Pengumpulan Data

Jenis data yang diterapkan adalah data sekunder, khususnya ulasan dalam bahasa Indonesia terkait aplikasi MyPertamina di *Google Play Store*. Menggunakan teknik *web scraping* (pengambilan sebuah dokumen semi-terstruktur dari halaman web), data sebanyak 3360 ulasan dari tahun 2018 hingga 1 Desember 2023 berhasil dikumpulkan [13]. Memperoleh variabel penting seperti tanggal ulasan, rating, nama akun Google, dan isi ulasan pengguna. Pendekatan ini memberikan pandangan mendalam terhadap respons pengguna terhadap aplikasi MyPertamina selama periode yang diteliti. Klasifikasi sentimen dibagi menjadi tiga kategori, pengolahan data dan implementasi model dilakukan menggunakan *Google Colab*, dan data ulasan. Metode klasifikasi yang diterapkan melibatkan *Random Forest*, SVM, dan *Naive Bayes*.

D. Pre-Processing

Preprocessing merupakan kunci vital dalam proses mining, menyeimbangkan data yang seringkali tidak ideal dengan mengatasi masalah seperti *missing value*, *data redundant*, *outliers*, atau format tidak sesuai [14]. Melalui serangkaian langkah eliminasi permasalahan, preprocessing tidak hanya membersihkan data,

tetapi juga memastikan kesiapan optimal untuk analisis lanjut. Dengan pendekatan ini, kompleksitas data dipahami, dan proses mining dapat berjalan lebih lancar, menghasilkan hasil lebih akurat.

1. *Cleaning*

Pada tahap *Cleaning*, dilakukan beberapa proses untuk mempersiapkan data dengan cermat. Pertama, URL dihapus dari kolom konten untuk menghilangkan tautan yang mungkin tidak relevan. Kedua, teks diubah menjadi *lowercase* agar konsistensi dalam analisis. Ketiga, dilakukan penghapusan *mention* untuk menghilangkan referensi pengguna. Keempat, *hashtag* dihilangkan untuk fokus pada kata kunci utama. Kelima, karakter yang tidak diperlukan setelah proses sebelumnya dihapus, termasuk tanda baca yang dapat mempengaruhi interpretasi teks. Selanjutnya, tanda baca dihilangkan sepenuhnya agar kata-kata dapat diproses dengan lebih baik [15]. Akhirnya, *extra whitespace* dihapus untuk memastikan kebersihan dan konsistensi dalam representasi teks. Proses ini memastikan data digunakan dalam analisis sentimen lebih terstruktur dan siap tahapan berikutnya.

2. *Filtering*

Filtering adalah tahap mengambil kata-kata penting dari hasil token dengan menggunakan algoritma *stoplist* (membuang kata kurang penting) atau *wordlist* (menyimpan kata penting). Pembuangan kata tidak penting (*stopword*) dapat menghapus kata-kata umum yang tidak memberikan kontribusi signifikan terhadap analisis sentimen. Penghapusan tanda baca yang tidak diperlukan [16].

3. *Stemming*

Stemming adalah proses pemetaan dan penguraian bentuk dari suatu kata menjadi bentuk kata dasarnya. Untuk melakukan stemming bahasa Indonesia kita dapat menggunakan *library Python Sastrawi* yang sudah kita siapkan di awal. *Library Sastrawi* menerapkan Algoritma Nazief dan Adriani dalam melakukan stemming bahasa Indonesia. Pengubahan kata berimbuhan menjadi kata dasar: mengubah kata-kata dengan imbuhan agar fokus pada makna dasar.

4. *Tokenizing*

Pemisahan kata (*tokenizing*): memisahkan setiap kata berdasarkan spasi, mempersiapkan data untuk tahap analisis lebih lanjut [17].

E. Pembagian Data

Data yang telah melalui proses pelabelan, dilakukan pembagian data menjadi 2, yaitu data latih dan data uji. Untuk masing-masing pembagian data data latih dan uji ini penulis membagi kembali menjadi 2 yaitu : Pertama, data latih berjumlah berjumlah lebih kecil dari data uji - Kedua, data latih berjumlah lebih banyak dari data uji.

F. Analisis dan Hasil

Tahap analisis dan hasil penelitian ini, klasifikasi dilakukan menggunakan Metode SVM, *Random Forest*, dan *Naive Bayes*. Metode *Naive Bayes* merupakan bentuk klasifikasi probalistik yang mengandalkan *Teorema Bayes* dengan asumsi independensi yang kuat, membentuk suatu model yang dijelaskan sebagai "model yang memiliki ciri-ciri kebebasan".

G. Prediksi Sentimen / Akurasi

Analisis sentimen mencapai prediksi sentimen dengan mengukur akurasi menggunakan Metode SVM, *Random Forest*, dan *Naive Bayes*, memberikan gambaran seberapa baik model klasifikasi dapat mengidentifikasi dan memahami sentimen dari ulasan MyPertamina. Hasil akurasi ini menjadi parameter kritis untuk mengevaluasi keefektifan setiap metode dalam memprediksi respons pengguna terhadap aplikasi tersebut.

Perancangan

A. *Web Scrapping*

Gambar 2 menunjukkan data-data ulasan diambil menggunakan fungsi *reviews()* dari *google-play-scraper*, data ulasan aplikasi telah berhasil diambil dengan penentuan parameter yang kritis seperti *id*, *lang*, dan *country*. Pemilihan pengaturan *default* bahasa "id" (Indonesia) dan pengurutan berdasarkan tanggal terbaru memberikan hasil sebanyak 3360 ulasan mentah, mencakup periode satu bulan terakhir dari rentang waktu 2018 hingga 1 Desember 2023. Proses ini memberikan gambaran komprehensif tentang respons pengguna MyPertamina yang terus berkembang dalam jangka waktu tersebut.

Tabel 2. Contoh implementasi *Cleaning Data*

Proses	Sebelum	Sesudah
Menghapus URL dari Kolom Konten	Aplikasi MyPertamina sangat membantu, kunjungi https://www.mypertamina.com untuk info lebih lanjut	Aplikasi MyPertamina sangat membantu, kunjungi untuk info lebih lanjut
Menghapus Teks jadi Lowercase	Aplikasi MyPertamina sangat Mudah Digunakan	aplikasi mypertamina sangat mudah digunakan
Menghapus Menton	Terima kasih @MyPertamina, aplikasi kalian luar biasa!	Terima kasih, aplikasi kalian luar biasa!
Menghapus Hashtag	Pengalaman baru dengan #MyPertamina sangat memuaskan!	Pengalaman baru dengan MyPertamina sangat memuaskan!
Menghapus Next Karakter	Penting banget nih!!	Penting banget nih!!
Menghapus Tanda Baca	Aplikasi MyPertamina terbaik!!! 👍👍	Aplikasi MyPertamina terbaik
Menghapus <i>Extra Whitespace</i> dan <i>Cose Folding</i>	Aplikasi MyPertamina Luar biasa	aplikasi mypertamina luar biasa

2. *Filtering*

Penerapan teknik *Filtering* dalam analisis sentimen ulasan MyPertamina menggunakan *stopwords*. *Stopword* adalah kata umum yang biasanya muncul dalam jumlah besar dan dianggap tidak memiliki makna [18]. Contoh *stopword* dalam bahasa Indonesia adalah “yang”, “dan”, “di”, “dari”, dll. Makna di balik penggunaan *stopword* yaitu dengan menghapus kata-kata yang memiliki informasi rendah dari sebuah teks, kita dapat fokus pada kata-kata penting sebagai gantinya.

content	score	Year	Month	Day	sentiment	content_token
3329	untuk aplikasi sudah bagus sayang sekali unt...	4	2018	9	13	1 [aplikasi, bagus, sayang, struk, langsung, spb...
1319	tolong untuk kamera di aplikasinya bisa diupgr...	4	2018	9	17	1 [tolong, kamera, aplikasinya, diupgrade, tamba...
1634	susah sekali menscan struknya cuma bisa sekal...	3	2018	9	17	0 [susah, menscan, struknya, doank, habis, zonk...

Gambar 5. *Filtering data (Stopword removal)*

3. *Stemming*

Penerapan teknik *Stemming* menggunakan teknik pemrosesan bahasa alami yang menurunkan infleksi kata ke bentuk akarnya, sehingga membantu dalam pra-pemrosesan teks, kata, dan dokumen untuk normalisasi teks. Sastrawi *Python* adalah *library python* sederhana yang memungkinkan Anda untuk mereduksi kata-kata infleksi dalam Bahasa Indonesia (Bahasa Indonesia) ke bentuk dasarnya atau akarnya (*stem*). Gambar 6 menggambarkan *port Python* dari proyek Sastrawi asli yang ditulis dalam PHP (kredit diberikan kepada penulis asli dan kontributor Sastrawi PHP).

content	score	Year	Month	Day	sentiment	content_token	stemmed
3329	untuk aplikasi sudah bagus sayang sekali unt...	4	2018	9	13	1 [aplikasi, bagus, sayang, struk, langsung, spb...	[aplikasi, bagus, sayang, struk, langsung, spb...
1319	tolong untuk kamera di aplikasinya bisa diupgr...	4	2018	9	17	1 [tolong, kamera, aplikasinya, diupgrade, tamba...	[tolong, kamera, aplikasi, diupgrade, tambah...
1634	susah sekali menscan struknya cuma bisa sekal...	3	2018	9	17	0 [susah, menscan, struknya, doank, habis, zonk...	[susah, menscan, struknya, doank, habis, zonk...
2791	di suruh upgrade versi terbaru nya di play sto...	4	2018	9	18	1 [suruh, upgrade, versi, terbaru, nya, play, st...	[suruh, upgrade, versi, baru, nya, play, store...
2407	nyesal download apk ini cuma berguna bagi masy...	1	2018	9	18	-1 [nyesal, download, apk, berguna, masyarakat, j...	[nyesal, download, apk, guna, masyarakat, jaka...

Gambar 6. *Stemming sastrawi*

4. *Tokenizing*

Penerapan teknik *Tokenizing* dalam proses pemisahan teks menjadi potongan-potongan yang disebut sebagai token untuk kemudian di analisa. Kata, angka, simbol, tanda baca dan entitas penting lainnya dapat dianggap sebagai token. Gambar 7 menggambarkan proses NLP, token diartikan sebagai “kata” meskipun *tokenize* juga dapat dilakukan pada paragraf maupun kalimat.

content	score	Year	Month	Day	sentiment	content_token
3329	untuk aplikasi sudah bagus sayang sekali unt...	4	2018	9	13	1 [untuk, aplikasi, sudah, bagus, sayang, sekali...
1319	tolong untuk kamera di aplikasinya bisa diupgr...	4	2018	9	17	1 [tolong, untuk, kamera, di, aplikasinya, bisa...
1634	susah sekali menscan struknya cuma bisa sekal...	3	2018	9	17	0 [susah, sekali, menscan, struknya, cuma, bisa...

Gambar 7. *Tokenizing*

Pemodelan

A. Wordcloud

Wordcloud digunakan untuk memunculkan citraan visual dari sebuah teks tertulis kemudian dapat dijadikan sebagai sebuah alat bantu dalam melakukan analisis teks. kata-kata yang ada digambarkan ke dalam suatu visualisasi dimana besar kecilnya huruf bergantung pada frekuensi munculnya kata tersebut di dalam teks. Singkatnya adalah kata terbesar adalah kata yang paling sering muncul dalam suatu data.

1. *Wordcloud* sentimen positif

Visualisasi *wordcloud* sentimen positif pada ulasan MyPertamina menciptakan gambaran grafis yang menyoroti kata-kata kunci yang sering muncul dalam konteks positif, memberikan pandangan intuitif terhadap aspek positif yang diapresiasi oleh pengguna.



Gambar 8. *Wordcloud* sentimen positif

2. *Wordcloud* sentimen netral

Wordcloud sentimen netral memberikan gambaran visual atas kata-kata yang paling umum dalam ulasan dengan sentimen netral terhadap MyPertamina. Hal ini membantu mengidentifikasi aspek-aspek yang dianggap netral oleh pengguna tanpa afiliasi positif atau negatif yang signifikan.



Gambar 9. *Wordcloud* sentimen netral

3. *Wordcloud* sentimen negatif

Melalui *wordcloud* sentimen negatif, kata-kata yang dominan dalam ulasan dengan sentimen negatif terhadap MyPertamina dipresentasikan secara visual. Ini memberikan wawasan terhadap masalah atau ketidakpuasan yang mungkin perlu diperbaiki untuk meningkatkan kepuasan pengguna.



Gambar 10. *Wordcloud* sentimen negatif

B. Penerapan TF-IDF

TF-IDF. Fungsi *CountVectorizer()* digunakan untuk pembuatan fitur menggunakan metode *Bag-of-Worlds*, sedangkan metode pembobotan TF-IDF dilakukan menggunakan fungsi *TfidfVectorizer()* yang disediakan dalam *library sklearn* dengan jumlah *n-gram* yang digunakan adalah 1 (*unigram*).

1. Preparation data

	sentiment	stemmed
0	-1	['aplikasi', 'gimana', 'min', 'kamera', 'ngga'...
1	1	['tolong', 'kamera', 'aplikasi', 'diupgrade', ...
2	0	['susah', 'menscan', 'struknya', 'doank', 'hab'...
3	-1	['nyesal', 'download', 'apk', 'guna', 'masyara'...
4	-1	['versi', 'buka', 'nya', 'bagus', 'jelek', 'sc'...

Gambar 11. Preparation data

Jika kita cek, item pada kolom *stemmed* memiliki tipe data `<class str>`, kita perlu ubah *str*.

```
['bug', 'mana', 'tolong', 'baik', 'aplikasi', 'kalo', 'kelola', 'bantu', 'mudah', 'transaksi', 'bbm', 'error', 'orang', 'malas', 'guna', 'terimakasih']
type : <class 'list'>
```

2. Term Frequency (TF)

Frekuensi relatif suatu kata dalam dokumen. Berdasarkan gambar, tampaknya setiap term memiliki TF yang sama, yang tidak biasa dan mungkin menunjukkan bahwa ini adalah TF untuk satu term dalam dokumen atau rata-rata TF.

$$w_{i,j} = t_{f_{i,j}} \times \log\left(\frac{n}{df_i}\right) \tag{1}$$

Keterangan:

$t_{f_{i,j}}$ = Banyaknya kata-*i* pada dokumen ke-*j*.

N = Total dokumen

df = Banyaknya dokumen yang mengandung kata ke-*i*.

Document merupakan Pandas Series row *df_list* yang sebelumnya dibuat. Setelah itu kita *apply()* fungsi *.calc_TF()* kedalam *Pandas Series 'df_list'* dan simpan hasilnya pada *Series 'TF_dict'*.

```
# Check TF result
index = 90

print('%20s' % "term", "\t", "TF\n")
for key in df["TF_dict"][index]:
    print('%20s' % key, "\t", df["TF_dict"][index][key])
```

Gambar 12. Cek TF Result

Result pada google colab, *print Panda Series 'TF_dict'* untuk *index row Dataframe* ke-90.

term	TF
bug	0.0625
mana	0.0625
tolong	0.0625
baik	0.0625
aplikasi	0.0625
kalo	0.0625
kelola	0.0625
bantu	0.0625
mudah	0.0625
transaksi	0.0625
bbm	0.0625
error	0.0625
orang	0.0625
malas	0.0625
guna	0.0625
terimakasih	0.0625

Gambar 13. Output cek TF result

$$\begin{aligned}
 w_{aplikasi,d1} &= 1 \times \left(\log\left(\frac{7}{1}\right) \right) \\
 &= 1 \times 0.845 \\
 &= 0.845
 \end{aligned}$$

Berbeda dengan perhitungan menggunakan *library sklearn, feature_extraction* di *Python*, bobot TF-IDF dihitung berdasarkan rumus berikut.

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{n}{df_i}\right) + 1 \quad (2)$$

Menambahkan nilai 1 tersebut yaitu kata yang muncul di setiap dokumen tidak sepenuhnya diabaikan. Perhitungan *log* pada *library* TF-IDF di *sklearn* menggunakan *logaritma natural*, yaitu dapat ditulis *eloga* atau dengan kata lain *logaritma* dengan basis *e* dengan $e = 2.718281828459.....$ Sehingga perhitungan bobot TF-IDF di *Python* menjadi.

$$\begin{aligned}
 w_{aplikasi,d1} &= tf_{aplikasi,d1} \times \ln\left(\frac{N}{df_i}\right) + 1 \quad (3) \\
 &= 1 \times \ln\left[\ln\left(\frac{7}{1}\right) + 1\right] \\
 &= 1 \times \ln(8) \\
 &= 1 \times 2.079 \\
 &= 2.079
 \end{aligned}$$

Dalam *TfidfVectorizer* terdapat beberapa parameter untuk diatur, seperti *smooth_idf* dan *norm*. Jika kita pilih *smooth_idf = true*, maka perhitungan bobot sebagai berikut.

$$\begin{aligned}
 w_{aplikasi,d1} &= tf_{aplikasi,d1} \times \ln\left(\frac{N}{df_i}\right) + 1 \quad (4) \\
 &= 1 \times \ln\left[\ln\left(\frac{7+1}{1+1}\right) + 1\right] \\
 &= 1 \times \ln(4) + 1 \\
 &= 1 \times 1.386 + 1 \\
 &= 2.386
 \end{aligned}$$

Jadi berdasarkan perhitungan dengan *logaritma natural* : $w_{aplikasi,d1} = 2.079$ sedangkan *smoothing* (*smooth_idf = true*): $w_{aplikasi,d1} = 2.386$. Jika ada pertanyaan kenapa bukan $\ln(8)$? Maka ini merujuk pada *logaritma natural* dari 4, bukan perkalian angka 4. Untuk lebih jelasnya $\ln(4)$ dapat dihitung sebagai *logaritma natural* dari 4 dengan besar $e = 2.71828$, jadi $\ln(4) = 1.386$ jadi rumus yang diberikan:

$$\begin{aligned}
 w_{aplikasi,d1} &= 1 \times \ln\left(\frac{7+1}{1+1}\right) + 1 \quad (5) \\
 &= 1 \times \ln(4) + 1 \\
 &= 1 \times 1.386 + 1 \\
 &= 2.386
 \end{aligned}$$

Jadi, untuk pernyataan ini benar, dan $\ln(4)$ adalah sekitar 1.386, bukan 8. Tetapi, jika tetap menggunakan $\ln(8)$ hasil perhitungan akan menjadi sekitar = 3.079. Nilai ini tidak konsisten dengan definisi $\ln(8)$ sebagai *logaritma natural* dari 8. Sehingga konsistensi rumus akan tetap mempertahankan $\ln(4)$ atau sekitar 1.386.

C. Modelling

1. Pengaturan Model

Peneliti menggunakan tiga model klasifikasi, yaitu SVM, *Random Forest*, dan *Naïve Bayes*, dalam eksperimen ini. Data dibagi dengan rasio 80:20 untuk training dan testing. Penggunaan *Google Colab* sebagai alat eksperimen memungkinkan untuk mengimplementasikan model dengan efisien. Hasil eksperimen menunjukkan bahwa model *Random Forest* mencapai akurasi tertinggi pada semua percobaan dibandingkan dengan SVM dan *Naïve Bayes*.

2. Pengujian Model

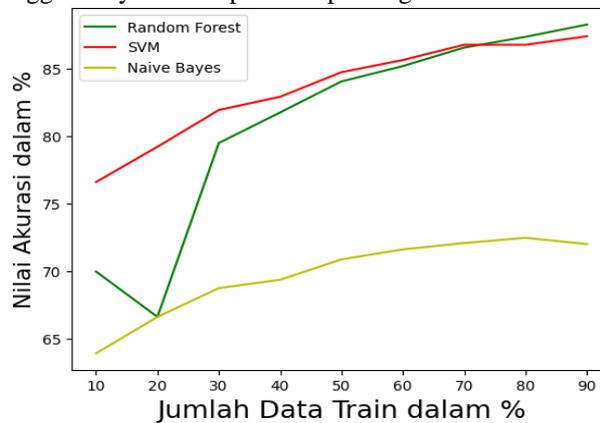
Pengujian model dilakukan dengan variasi rasio pembagian data training dan testing, yang diterapkan dalam uji tren dan uji tes. Pada uji tren, data dibagi dengan rasio yang berubah secara berurutan dari 90:10 hingga 10:90. Hasil menunjukkan bahwa akurasi model *Random Forest*, SVM, dan *Naïve Bayes* bergantung pada rasio tersebut. Model *Random Forest* menunjukkan konsistensi dalam mempertahankan akurasi tinggi pada berbagai rasio, sementara SVM dan *Naïve Bayes* cenderung

mengalami fluktuasi.

index	data_train	data_tes	random_forest	SVM	naive_bayes
0	90	10	88.28	87.41	71.99
1	80	20	87.37	86.78	72.46
2	70	30	86.57	86.78	72.07
3	60	40	85.2	85.65	71.6
4	50	50	84.06	84.75	70.86
5	40	60	81.76	82.92	69.35
6	30	70	79.49	81.94	68.73
7	20	80	66.59	79.21	66.59
8	10	90	69.96	76.6	63.89

Gambar 17. Pengujian *Modeling Random Forest, SVM, dan Naive Bayes*

Sementara itu, pada uji tes dengan rasio 80:20, 70:30, dan 60:40, model *Random Forest* tetap unggul dengan akurasi yang konsisten lebih tinggi daripada model *SVM* dan *Naive Bayes*. Uji tren dan uji tes ini memberikan wawasan mendalam tentang sejauh mana model-model tersebut dapat mempertahankan kinerjanya saat data dibagi dengan berbagai proporsi. Pemilihan model *Random Forest* sebagai model unggulan didukung oleh konsistensi kinerja tinggi pada skenario uji tren dan uji tes, menunjukkan ketangguhannya terhadap variasi pembagian data.



Gambar 18. Visualisasi Akurasi *Modeling*

3. Evaluasi Model

Tahapan evaluasi model, peneliti memanfaatkan dataset berisi 3360 ulasan dengan pembagian data train dan tes. Untuk mengukur performa model, peneliti menggunakan *Confusion Matrix Multi-Class Classification 3x3*, memungkinkan analisis yang terperinci terhadap hasil klasifikasi untuk setiap kelas. Dengan memanfaatkan *Confusion Matrix*, peneliti dapat menghasilkan metrik evaluasi seperti *Accuracy, Precision, Recall, dan F1-Score* pada masing-masing model.

Tabel 3. *Confusion Matrix Multi-Class Classification*

	<i>Prediksi -1</i>	<i>Prediksi 0</i>	<i>Prediksi 1</i>
<i>Actual -1</i>	<i>TP (True Positive)</i>	<i>FN (False Negative)</i>	<i>FP (False Positive)</i>
<i>Actual 0</i>	<i>FP (False Positive)</i>	<i>TP (True Positive)</i>	<i>FN (False Negative)</i>
<i>Actual 1</i>	<i>FP (False Positive)</i>	<i>FN (False Negative)</i>	<i>TP (True Positive)</i>

Keterangan:

- TP : Jumlah data positif yang terdeteksi dengan benar
- FP : Jumlah data negatif yang terdeteksi sebagai data positif
- FN : Jumlah data positif namun terdeteksi sebagai data negatif
- TN : Jumlah data negatif yang terdeteksi dengan benar

a. *Random Forest*

Hasil evaluasi model *Random Forest* menunjukkan tingkat akurasi yang sangat tinggi sebesar 99.77%, menandakan kemampuan model untuk melakukan klasifikasi yang sangat baik pada dataset yang digunakan. Nilai *precision, recall, dan f1-score* untuk setiap kelas (kelas -1, 0, dan 1) juga mencapai 1.00, menunjukkan ketepatan dan konsistensi model dalam mengidentifikasi setiap kelas dengan sangat baik. Evaluasi model ini memberikan indikasi bahwa *Random Forest* secara efektif dapat mengklasifikasikan data ke dalam tiga kelas dengan performa yang sangat

tinggi.

Tabel 4. Random Forest Evaluation

	Prediksi -1	Prediksi 0	Prediksi 1
Actual -1	291	0	1
Actual 0	1	283	0
Actual 1	0	0	295

1) Accuracy

$$\begin{aligned}
 Accuracy &= \frac{TP_{kelas-1} + TP_{kelas 0} + TP_{kelas1}}{jumlah\ data} \\
 &= \frac{291+283+295}{869} \\
 &= 0.997
 \end{aligned} \tag{6}$$

1) Precision

$$\begin{aligned}
 Precision &= \frac{TP}{TP + FP} \\
 &= kelas - 1 = \frac{291}{291+1} = 0.996 \\
 &= kelas 0 = \frac{283}{283+1} = 1.0000 \\
 &= kelas1 = \frac{295}{295+1} = 0.9966 \\
 Average\ Precision &= \frac{0.996+1.0000+0.9966}{3} = 0.9977
 \end{aligned} \tag{7}$$

2) Recall

$$\begin{aligned}
 Recall &= \frac{TP}{TP + FN} \\
 &= kelas - 1 = \frac{291}{291+1} = 0.996 \\
 &= kelas 0 = \frac{283}{283+1} = 0.9966 \\
 &= kelas1 = \frac{295}{295+0} = 1.0000 \\
 Average\ Recall &= \frac{0.996+1.0000+0.9966}{3} = 0.9977
 \end{aligned} \tag{8}$$

3) F1- Score

$$\begin{aligned}
 F1 - Score &= 2 x \frac{Precision x Recall}{Precision + Recall} \\
 &= kelas - 1 = 2 x \frac{0.9966 x 0.9966}{0.9966+0.9966} = 0.9966 \\
 &= kelas 0 = 2 x \frac{1.0000 x 0.9965}{1.0000+0.9965} = 0.9982 \\
 &= kelas1 = 2 x \frac{0.9966 x 1.0000}{0.9966+1.0000} = 0.9983 \\
 Average\ F1 - Score &= 2 x \frac{0.9966+0.9982+0.9983}{3} = 0.9977
 \end{aligned} \tag{9}$$

```

Random Forest Evaluation:
Accuracy: 0.9977037887485649
Precision: 0.9977076674837868
Recall: 0.9977037887485649
F1-Score: 0.9977037476342979
Confusion Matrix:
[[291  0  1]
 [ 1 283  0]
 [ 0  0 295]]
Classification Report:
              precision    recall  f1-score   support

-1         1.00         1.00         1.00         292
 0         1.00         1.00         1.00         284
 1         1.00         1.00         1.00         295

 accuracy
macro avg         1.00         1.00         1.00         871
weighted avg         1.00         1.00         1.00         871
    
```

Gambar 19. Output Confusion Matrix Random Forest

b. SVM

Hasil evaluasi model SVM menunjukkan tingkat akurasi yang sangat tinggi sebesar 99.31%, mencerminkan kemampuan model untuk melakukan klasifikasi yang sangat baik pada dataset yang digunakan. *Precision*, *recall*, dan *f1-score* untuk setiap kelas (kelas -1, 0, dan 1) juga mencapai nilai tinggi sekitar 0.99, menunjukkan ketepatan dan konsistensi model dalam mengidentifikasi setiap kelas dengan sangat baik. Evaluasi model ini menunjukkan bahwa SVM efektif dalam mengklasifikasikan data dalam tiga kelas dengan performa sangat tinggi.

Tabel 5. SVM *evaluation*

	<i>Prediksi -1</i>	<i>Prediksi 0</i>	<i>Prediksi 1</i>
<i>Actual -1</i>	290	0	2
<i>Actual 0</i>	0	284	0
<i>Actual 1</i>	4	0	291

1) *Accuracy*

$$\begin{aligned}
 Accuracy &= \frac{TP_{kelas-1} + TP_{kelas 0} + TP_{kelas1}}{jumlah\ data} \\
 &= \frac{290+284+291}{865} \\
 &= 0.9931
 \end{aligned} \tag{10}$$

2) *Precision*

$$\begin{aligned}
 Precision &= \frac{TP}{TP + FP} \\
 &= kelas - 1 = \frac{290}{290+4} = 0.9864 \\
 &= kelas 0 = \frac{284}{284+0} = 1.0000 \\
 &= kelas1 = \frac{291}{291+2} = 0.9932 \\
 Average\ Precision &= \frac{0.9864+1.0000+0.9932}{3} = 0.9932
 \end{aligned} \tag{11}$$

3) *Recall*

$$\begin{aligned}
 Recall &= \frac{TP}{TP + FN} \\
 &= kelas - 1 = \frac{291}{291+1} = 0.996 \\
 &= kelas 0 = \frac{283}{283+1} = 0.9966 \\
 &= kelas1 = \frac{295}{295+0} = 1.0000 \\
 Average\ Recall &= \frac{0.996+1.0000+0.9966}{3} = 0.9977
 \end{aligned} \tag{12}$$

4) *F1 - Score*

$$\begin{aligned}
 F1 - Score &= 2 x \frac{Precision \times Recall}{Precision + Recall} \\
 &= kelas - 1 = 2 x \frac{0.9864 \times 0.9932}{0.9864+0.9932} = 0.9966 \\
 &= kelas 0 = 2 x \frac{1.0000 \times 1.0000}{0.9864+0.9832} = 0.9982 \\
 &= kelas1 = 2 x \frac{0.9932 \times 0.9864}{0.9932+0.9864} = 0.9983 \\
 Average\ F1 - Score &= 2 x \frac{0.9966+0.9982+0.9983}{3} = 0.9932
 \end{aligned} \tag{13}$$

```
SVM Evaluation:
Accuracy: 0.9931113662456946
Precision: 0.9931269334181638
Recall: 0.9931113662456946
F1-Score: 0.9931114062298705
Confusion Matrix:
[[290  0  2]
 [ 0 284  0]
 [ 4  0 291]]
Classification Report:
      precision    recall  f1-score   support

-1      0.99      0.99      0.99         292
 0      1.00      1.00      1.00         284
 1      0.99      0.99      0.99         295

 accuracy          0.99          0.99          0.99          871
 macro avg          0.99          0.99          0.99          871
 weighted avg          0.99          0.99          0.99          871
```

Gambar 20. Output Confusion Matrix SVM

c. *Naïve Bayes*

Hasil evaluasi model *Naive Bayes* menunjukkan tingkat akurasi sebesar 90.24%, mencerminkan kinerja yang baik dalam mengklasifikasikan data pada dataset yang digunakan. *Precision*, *recall*, dan *f1-score* untuk setiap kelas (kelas -1, 0, dan 1) memiliki nilai yang cukup seimbang, menunjukkan kemampuan model dalam mengidentifikasi setiap kelas dengan baik. Evaluasi model ini mengindikasikan bahwa *Naive Bayes* mampu memberikan performa yang baik dalam mengklasifikasikan data dengan seimbang antara ketepatan dan kecocokan.

Tabel 6. *Naive Bayes Evaluation*

	<i>Prediksi -1</i>	<i>Prediksi 0</i>	<i>Prediksi 1</i>
<i>Actual -1</i>	228	39	25
<i>Actual 0</i>	0	284	0
<i>Actual 1</i>	10	11	274

1) *Accuracy*

$$Accuracy = \frac{TP_{kelas-1} + TP_{kelas 0} + TP_{kelas1}}{jumlah\ data} = \frac{228+284+271}{865} = 0.9024 \tag{14}$$

2) *Precision*

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

$$= kelas - 1 = \frac{228}{228+39+25} = 0.9579$$

$$= kelas 0 = \frac{284}{284+0+0} = 1.0000$$

$$= kelas1 = \frac{274}{274+10+11} = 0.9163$$

$$Average\ Precision = \frac{0.9579+1.0000+0.9163}{3} = 0.9082 \tag{15}$$

3) *Recall*

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

$$= kelas - 1 = \frac{291}{291+1} = 0.996$$

$$= kelas 0 = \frac{283}{283+1} = 0.9966$$

$$= kelas1 = \frac{295}{295+0} = 1.0000$$

$$Average\ Recall = \frac{0.996+1.0000+0.9966}{3} = 0.9977 \tag{16}$$

4) *F1-Score*

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{17}$$

$$\begin{aligned}
 &= \text{kelas} - 1 = 2 \times \frac{0.9579 \times 0.7808}{0.9579+0.7808} = 0.8603 \\
 &= \text{kelas} 0 = 2 \times \frac{0.8502 \times 1.0000}{0.8502+1.0000} = 0.9190 \\
 &= \text{kelas} 1 = 2 \times \frac{0.9163 \times 0.9288}{0.9163+0.9288} = 0.9225 \\
 \text{Average F1 - Score} &= 2 \times \frac{0.8603+0.9190+0.9225}{3} = 0.9006
 \end{aligned}$$

```

Naive Bayes Evaluation:
Accuracy: 0.9024110218140069
Precision: 0.9087836630742016
Recall: 0.9024110218140069
F1-Score: 0.9005829214188408
Confusion Matrix:
[[228 39 25]
 [ 0 284  0]
 [ 10 11 274]]
Classification Report:
              precision    recall  f1-score   support

-1           0.96         0.78         0.86         292
 0           0.85         1.00         0.92         284
 1           0.92         0.93         0.92         295

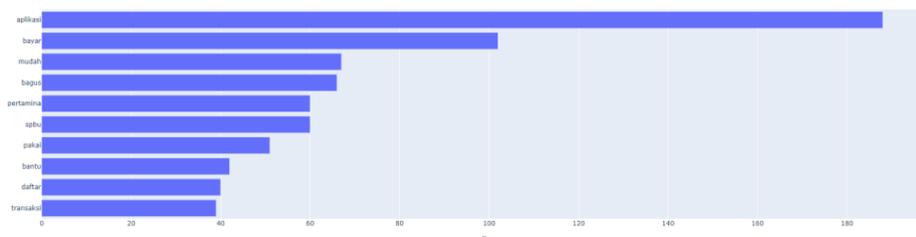
 accuracy          0.90         0.90         0.90         871
 macro avg         0.91         0.90         0.90         871
 weighted avg      0.91         0.90         0.90         871
    
```

Gambar 21. Output Confusion Matrix Naive Bayes

4. Most Common Words

Tabel 7. Most common words

No	Kata	Frekuensi
1.	Aplikasi	188
2.	Bayar	102
3.	Mudah	67
4.	Bagus	66
5.	Pertamina	60
6.	SPBU	60
7.	Pakai	51
8.	Bantu	42
9.	Daftar	40
10.	Transaksi	39



Gambar 22. Visualisasi daftar 10 istilah dengan nilai TF-IDF terbesar

Tabel 8 dan Gambar 21 menampilkan daftar sepuluh kata yang paling sering muncul dalam sebuah *dataset*, dengan "Aplikasi" menjadi yang teratas dengan *frekuensi* 188 kali. Grafik batang di bawahnya memvisualisasikan data yang sama, memudahkan pemahaman visual terhadap distribusi frekuensi kata-kata tersebut, dengan "Bayar" dan "Mudah" mengikuti setelah "Aplikasi" sebagai kata yang sering digunakan. Visualisasi ini menggunakan metode TF-IDF, yang membantu dalam mengidentifikasi kepentingan kata dalam dokumen yang relatif terhadap corpus (kumpulan dokumen) yang dianalisis.

Kesimpulan

Berdasarkan analisis yang telah dijelaskan sebelumnya terkait Analisis Sentimen Masyarakat Terhadap Sistem Pembayaran MyPertamina dengan Metode *Random Forest*, SVM dan *Naive Bayes* sebagai yaitu pertama, penelitian ini menggunakan tiga metode analisis sentimen, yaitu *Random Forest*, SVM, dan *Naive Bayes*, berhasil diimplementasikan untuk mengevaluasi ulasan pengguna terhadap aplikasi MyPertamina. Data ulasan

aplikasi MyPertamina berhasil dikumpulkan secara eksklusif melalui teknik *web scraping* dari *Google Play Store*, dengan total 3360 ulasan dianalisis dari periode tahun 2018 hingga Desember 2023. Kedua, klasifikasi sentimen pada analisis dibagi menjadi tiga kategori, yakni positif, negatif, dan netral, dan penggunaan metode *Random Forest* menunjukkan akurasi tinggi dalam mengklasifikasikan sentimen pengguna. Keputusan menggunakan *Google Colab* sebagai alat utama dalam pengolahan data dan implementasi model klasifikasi memberikan efisiensi dan aksesibilitas dalam menjalankan eksperimen. Ketiga, Hasil evaluasi menunjukkan bahwa ketiga metode analisis sentimen, yaitu *Random Forest*, *SVM*, dan *Naïve Bayes*, menghasilkan akurasi yang tinggi pada evaluasi ulasan aplikasi MyPertamina. *Random Forest* mencapai tingkat keakuratan yang paling memuaskan dengan akurasi, *precision*, *recall*, dan *F1-Score* mencapai 99.77%. *SVM* dan *Naïve Bayes* juga menunjukkan performa yang sangat baik, masing-masing dengan akurasi 99.31% dan 90.24%. Nilai *Precision*, *Recall*, dan *F1-Score* yang optimal pada ketiga metode mengindikasikan keefektifan mereka dalam menganalisis sentimen ulasan pengguna.

Daftar Pustaka

- [1] M. Andriani, “Analisis Pengaruh Kualitas Pelayanan Terhadap keputusan Pembelian pada PT.Pertamina (Studi Pada PT.Pertamina Cemara Asri, Medan,” *J. Stindo Prof.*, vol. VI, pp. 3–7, 2020.
- [2] G. Rozy Hrp and N. Aslami, “Analisis Dampak Kebijakan Perubahan Publik Harga BBM terhadap Perekonomian Rakyat Indonesia,” *J. Ilmu Komputer, Ekon. dan Manaj.*, vol. 2, no. 1, pp. 1464–1474, 2022.
- [3] A. A. Fitriani, “Efektivitas Penggunaan Aplikasi My Pertamina di Era Kenaikan BBM Bersubsidi,” *J. Pros. Mateandrau*, vol. 1, no. 2, 2022.
- [4] E. O. Safitri, Y. T. Musityo, and N. H. Wardhani, “Analisis Perilaku Penggunaan Mobile Payment Aplikasi OVO menggunakan Technology Acceptance Model (TAM) Termodifikasi,” *Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 8, pp. 8184–8189, 2019.
- [5] A. . Sinurat, “Aplikasi MyPertamina Untuk Meningkatkan Keterlibatan Konsumen,” *J. Internasioanl Tinj. Bisnis*, vol. 5, no. 1, pp. 65–73, 2022.
- [6] N. K. Hikmawati, “Analisis Kualitas Layanan My Pertamina Menggunakan Pendekatan e-GovQual pada Beberapa Kota Percobaan,” *J. Manaj. Inform.*, vol. 12, no. 2, pp. 100–111, 2022, doi: 10.34010/jamika.v12i2.7977.
- [7] B. Marewa, “Mulai 21 Maret Pembelian Solar di Sulsel Wajib Pakai QR-Code My Pertamina,” *RADARSELATAN.FAJAR.CO.ID*, 2023. .
- [9] A. L. Fairuz, R. D. Ramadhani, and N. A. F. Tanjung, “Analisis Sentimen Masyarakat Terhadap COVID-19 Pada Media Sosial Twitter,” *J. Dinda Data Sci. Inf. Technol. Data Anal.*, vol. 1, no. 1, pp. 42–51, 2021, doi: 10.20895/dinda.v1i1.180.
- [10] C. Suardi, A. N. Handayani, R. A. Asmara, A. P. Wibawa, L. N. Hayati, and H. Azis, “Design of Sign Language Recognition Using E-CNN,” *3rd 2021 East Indones. Conf. Comput. Inf. Technol. EIconCIT 2021*, pp. 166–170, 2021, doi: 10.1109/EIconCIT50028.2021.9431877.
- [11] D. Musfiroh, U. Khaira, P. E. P. Utomo, and T. Suratno, “Analisis Sentimen terhadap Perkuliahan Daring di Indonesia dari Twitter Dataset Menggunakan InSet Lexicon,” *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 1, no. 1, pp. 24–33, 2021, doi: 10.57152/malcom.v1i1.20.
- [12] L. N. H. Dian Azurah, Purnawansyah, Herdianti Darwis, “Klasifikasi Penyakit Bawang Merah Menggunakan Naïve Bayes dan Convolutional Neural Network.” 2023.
- [13] M. F. Banjar, I. Irawati, F. Umar, and L. N. Hayati, “Analysis of Stroke Classification Using Random Forest Method,” *Ilk. J. Ilm.*, vol. 14, no. 3, pp. 186–193, 2022, doi: 10.33096/ilkom.v14i3.1252.186-193.
- [14] P. Pertamina(Persero), *MyPertamina*. 2018, p. <https://play.google.com/store/search?q=mypertamina>.
- [15] E. Hasibuan and E. Allistair, “Analisis Sentimen pada Ulasan Aplikasi Amazon Shopping di Google Ply Store Menggunakan Naive Bayes Classifier,” vol. 1, no. 3, pp. 13–24, 2022.
- [16] N. Bayes, “Analisis Sentimen Dengan Naive Bayes Terhadap Komentar Aplikasi Tokopedia,” vol. 6, no. 1, 2019.
- [17] U. Kulsum, M. Jajuli, and N. Sulistiyowati, “Analisis Sentimen Aplikasi WETV di Google Play Store Menggunakan Algoritma Support Vector Machine,” vol. 6, no. 2, pp. 205–212, 2022.
- [18] M. Afdal and R. Novita, “Sentiment Analysis of PLN Mobile Application Review Using Naïve Bayes Classifier and K-Nearest Neighbor Algorithm Analisis Sentimen Ulasan Aplikasi PLN Mobile Menggunakan Algoritma Naïve Bayes Classifier dan K-Nearest Neighbor,” vol. 4, no. January, pp. 10–19, 2024.
- [19] C. Maulida, T. Yunanda, M. Hanafi, W. Mega, and P. Duhita, “Sentiment Analysis on TikTok Shop

Reviews Using Long Short-Term Memory Method to Find Business Opportunity,” *Inf. J. Ilm. Bid. Teknol. Inf. dan Komun.*, vol. 9, no. 1, pp. 1–7, 2024, [Online]. Available: <https://doi.org/10.25139/inform.v9i1.6524>.