
Analisis Sentimen Review Aplikasi Jogja Smart Service pada Google Play Store Menggunakan Metode SVM

Akbar Sidiq Kurniawan^{*1}, Irfan Pratama²

^{1,2}) Program Studi Sistem Informasi, Fakultas Teknologi Informasi, Universitas Mercu Buana

Email: ^{*1}akbarsidiq2000@gmail.com, ²irfan@mercubuana-yogya.ac.id

(Naskah masuk: 31 Jui 2024, diterima untuk diterbitkan: 20 Januari 2025)

Abstrak: Perkembangan Teknologi Informasi dan Komunikasi (TIK) telah mengubah cara informasi disampaikan dan mempermudah akses informasi bagi masyarakat. Salah satu implementasi smart city adalah layanan publik berbasis TIK, yang bertujuan mengatasi berbagai masalah masyarakat. Dengan informasi TIK terintegrasi, pelayanan publik menjadi lebih efektif, efisien, dan tepat sasaran. salah satunya pelayanan publik adalah aplikasi Jogja Smart Service yang bisa didownload di google play store. Saat ini aplikasi Jogja Smart Service telah didownload sebanyak 100 ribu kali dengan rating 4.7. Pada Google play store bisa dilihat di kolom komentar mengenai ulasan yang telah tulis oleh pengguna aplikasi Jogja Smart Service ini. SVM adalah metode yang memberikan hasil yang baik dalam klasifikasi, karena kemampuannya menemukan hyperplane terbaik untuk memisahkan kelas yang berbeda. Dalam penelitian ini, analisis sentimen terhadap ulasan aplikasi Jogja Smart Service dilakukan menggunakan metode SVM dengan variasi kernel: linear, polynomial, RBF, dan Sigmoid. Hasil pengujian menunjukkan bahwa SVM mampu melakukan analisis sentimen ulasan pengguna aplikasi Jogja Smart Service dengan baik, terutama menggunakan RBF kernel yang mencapai akurasi 90%, lebih tinggi dibandingkan dengan linear, polynomial, dan Sigmoid kernel.

Kata kunci - Jogja Smart Service; SVM; play store; sentiment analisis; text mining

Sentiment Analysis of Jogja Smart Service Application Reviews On Google Play Store Using SVM

Abstract: The development of Information and Communication Technology (ICT) has transformed the way information is delivered and has made it easier for the public to access information. One implementation of the smart city concept is ICT-based public services, aimed at addressing various societal issues. With integrated ICT information, public services become more effective, efficient, and targeted. One such public service is the Jogja Smart Service application, which can be downloaded from the Google Play Store. Currently, the Jogja Smart Service application has been downloaded 100,000 times with a rating of 4.7. In the Google Play Store, user reviews of the Jogja Smart Service application can be found in the comments section. SVM is a method that provides good results in classification due to its ability to find the best hyperplane to separate different classes. In this study, sentiment analysis of the Jogja Smart Service application reviews was conducted using the SVM method with kernel variations: linear, polynomial, RBF, and Sigmoid. The test results showed that SVM is capable of performing sentiment analysis of the Jogja Smart Service application reviews well, especially using the RBF kernel, which achieved an accuracy of 90.5%, higher than the linear, polynomial, and Sigmoid kernels.

Keywords - Jogja Smart Service; SVM; play store; sentiment analysis; text mining

1. PENDAHULUAN

Perkembangan Teknologi Informasi dan Komunikasi (TIK) telah mengubah cara penyampaian informasi dan mempermudah akses masyarakat terhadap berbagai informasi. Salah satu implementasi dari konsep *smart city* adalah layanan publik berbasis TIK, yang bertujuan untuk mengatasi berbagai masalah di masyarakat. Dengan adanya informasi yang berbasis TIK terintegrasi, pelayanan publik menjadi lebih efektif, efisien, dan tepat sasaran bagi pengguna[1]. Untuk menerapkan konsep smart city, diperlukan pemanfaatan internet. Internet bukan lagi sesuatu

yang baru bagi masyarakat. Saat ini, penggunaan internet telah meluas ke hampir semua aspek kehidupan, termasuk bidang kesehatan, pendidikan, industri, perdagangan, serta pemerintahan[2].

Penggunaan teknologi dalam birokrasi pemerintah daerah juga dapat meningkatkan fungsi manajemen kota. Salah satu kebijakan yang muncul seiring dengan perkembangan teknologi adalah kebijakan mengenai Konsep *Smart City*[3]. Konsep *smart city*, yang juga dikenal sebagai kota pintar, telah banyak diterapkan di negara-negara maju sejak awal perkembangan teknologi komunikasi dan informasi. Konsep ini muncul akibat pertumbuhan penduduk perkotaan dan perubahan signifikan dalam pola perilaku dan kebiasaan. Hal ini mendorong masyarakat, perencana kota, sektor swasta, dan pemerintah untuk menciptakan visi baru yaitu kota pintar. Definisi *smart city* terus berkembang dan sangat bervariasi [4].

Namun, keberhasilan suatu aplikasi tidak hanya bergantung pada fitur atau fungsinya tetapi juga pada reaksi dan pengalaman pengguna. Ulasan atau review yang diberikan oleh pengguna melalui *platform* seperti *Google Play Store* mencerminkan persepsi dan kepuasan mereka terhadap aplikasi. Oleh karena itu, penting untuk menganalisis ulasan ini untuk lebih memahami bagaimana pengguna mengevaluasi kualitas, keandalan, dan kegunaan aplikasi *Jogja Smart Service*.

Pada penelitian yang dilakukan oleh Kaburuan yang mengenai review aplikasi digital korlantas pada *google play store*, penelitian tersebut menggunakan metode *Support Vector Machine*, Hasil penelitian menunjukkan bahwa penggunaan metode SVM menghasilkan kinerja yang baik pada skenario rasio data 90:10, dengan akurasi mencapai 0.82. Namun, model SVM menunjukkan kinerja yang lebih rendah pada skenario rasio data 80:20 dan 60:40, dengan akurasi sebesar 0.74.[5]

Pada penelitian yang dilakukan oleh Muhammadin yang menganalisis Sentimen Pada Ulasan Aplikasi Kredivo Dengan Algoritma SVM Dan NBC, tujuan dari penelitian ini adalah untuk membandingkan kedua metode tersebut anatar SVM dan NBC mana yang lebih baik, dari penelitian tersebut menghasilkan Tingkat keseluruhan dapat dilihat dari nilai akurasi dengan algoritma SVM lebih tinggi dibanding *Naive Bayes Classifier*. *Support Vector Machine* menghasilkan akurasi 83,3% Kemudian untuk algoritma *Naive Bayes Classifier* sendiri menghasilkan nilai akurasi sebesar 80,8%[6].

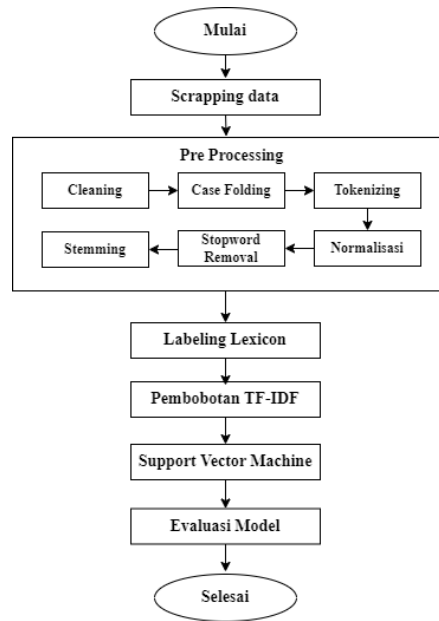
Dari penelitian sebelumnya dapat disimpulkan bahwa metode SVM merupakan salah satu cara efektif untuk menganalisis sentimen dari teks ulasan. SVM banyak digunakan dalam berbagai aplikasi pemrosesan bahasa alami, termasuk klasifikasi sentimen. SVM memungkinkan pengklasifikasian ulasan ke dalam kategori positif, negative maupun netral sehingga memberikan pemahaman yang lebih komprehensif tentang pandangan pengguna terhadap aplikasi *Jogja Smart Service*.

Dalam penelitian ini *labelling* ulasan menggunakan fitur *Lexicon* yang mana lexicon dalam analisis sentimen, mengacu pada kumpulan kata atau frasa yang telah diberi label dengan nilai sentimen tertentu. *Lexicon* ini digunakan untuk mengidentifikasi dan mengukur sentimen dalam teks. Setiap kata dalam *lexicon* biasanya diberi nilai positif, negatif, atau netral yang mencerminkan emosinya.

Tujuan penelitian ini adalah untuk melakukan analisis sentimen terhadap ulasan aplikasi *Jogja Smart Service* di *Google Play Store* menggunakan metode SVM. Diharapkan, hasil penelitian ini dapat memberikan pemahaman yang lebih mendalam tentang persepsi dan kepuasan pengguna terhadap aplikasi *Jogja Smart City*, serta memberikan rekomendasi perbaikan atau peningkatan yang mungkin diperlukan untuk meningkatkan kualitas layanan publik di Kota Yogyakarta.

2. METODE PENELITIAN

Metode Penelitian ini bertujuan untuk menganalisis ulasan pengguna pada aplikasi *Jogja Smart Service* di situs *Google Play* dengan memanfaatkan algoritma *Support Vector Machine*. Proses penelitian ini melibatkan beberapa tahapan yang dapat dilihat secara rinci pada Gambar 1.



Gambar 1. Tahap penelitian

2.1. Scrapping data

Scraping data merupakan proses pengambilan atau ekstraksi informasi dari situs web secara otomatis dengan menggunakan skrip atau program. Proses ini melibatkan pengunduhan halaman web dan pemrosesan kontennya untuk mengekstrak data yang diperlukan. Teknik ini sering digunakan untuk mengumpulkan data dalam jumlah besar dari berbagai sumber online, seperti ulasan pengguna, harga produk, atau informasi lain yang tersedia di situs web.

2.2. Preprocessing

Proses prapemrosesan data dilakukan karena banyak data yang masih menggunakan bahasa yang tidak sesuai dengan ejaan atau mengandung beberapa atribut yang tidak diperlukan untuk penelitian ini. Pada tahap ini, digunakan metode text mining, yaitu proses analisis teks untuk mencari informasi penting yang relevan dengan hasil tertentu. Prinsip dan cara yang digunakan dalam data mining diterapkan untuk mengidentifikasi pola dalam teks[7]. Preprocessing data terdiri dari

1. Cleaning Text

Pada tahap ini dilakukan beberapa proses, diantaranya penghapusan missing values data, menghapus simbol seperti `!"#$%&\'+,-./:;<=>? @[\]^_`{|}~'`, karakter spesial, hingga angka. Data duplikat pun dihapus[8].

2. Case Folding

Pada langkah ini, dilakukan untuk penyeragaman kata dengan mengonversi semua huruf menjadi huruf kecil. Langkah ini diambil untuk menghindari masalah di mana kata yang sama tidak terdeteksi akibat perbedaan kapitalisasi. Selain itu, ini juga membantu dalam mengelompokkan istilah yang memiliki makna yang sama[8].

3. Tokenizing

Tokenizing adalah tahap penting dalam pemrosesan teks yang bertujuan untuk memisahkan setiap kata dalam sebuah kalimat. Proses ini bertujuan untuk memecah kalimat menjadi unit-unit kata yang lebih kecil, sehingga setiap kata dapat dianalisis secara individu[9]. Jika karakter ke-*i* bukan tanda pemisah seperti titik (`.`), koma (`,`), spasi, atau pemisah lainnya, maka karakter tersebut akan digabungkan dengan karakter berikutnya[6].

4. Normalisasi

Normalisasi kalimat, yaitu memperbaiki ejaan kata yang disingkat agar sesuai dengan KBBI sehingga teks yang dihasilkan sesuai dengan standar ejaan yang berlaku.

5. *Stopword Removal*

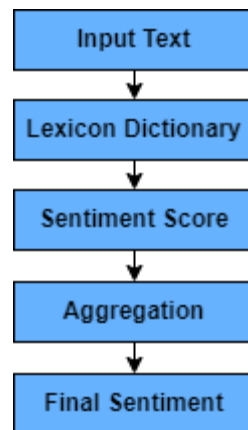
Pada tahap ini, diterapkan dalam tahap penghapusan kata-kata yang tidak memberikan kontribusi pada kalimat namun tidak mengurangi informasi penting dari kalimat tersebut. Melalui penghapusan kata-kata yang tidak bermakna ini, diharapkan dapat mengoptimalkan waktu pengolahan data dengan mengurangi isi dari setiap kalimat dalam dataset [8].

6. *Stemming*

Stemming adalah proses yang bertujuan untuk mengubah kata menjadi bentuk dasarnya atau bentuk asli dari kata tersebut. Proses ini menghapus imbuhan atau akhiran dari kata, sehingga kata tersebut kembali ke bentuk asalnya. Dengan melakukan stemming, kita dapat menyederhanakan variasi kata yang memiliki makna serupa, sehingga analisis teks menjadi lebih efisien dan konsisten[10].

2.3. *Lexicon*

Metode berbasis leksikon adalah metode ilmiah yang sering digunakan dalam penelitian analisis sentimen. Hasil analisis dengan metode ini berupa klasifikasi sentimen positif, negatif, dan netral. Metode ini termasuk dalam machine learning yang bersifat unsupervised. Penggunaan metode ini cukup mudah, di mana bobot kata yang ditemukan oleh sistem akan dihitung dan hasilnya akan diklasifikasikan ke dalam sentimen tertentu[11]. Dalam penerapannya, terdapat dua kamus yang digunakan: satu berisi kumpulan kata dengan sentimen positif dan satu lagi berisi kumpulan kata dengan sentimen negatif. Kedua kamus ini digunakan sebagai daftar kata (wordlist)[12]. Berikut adalah ilustrasi dari proses Lexicon



Gambar 2. Proses Lexicon

Input Text merupakan proses dimulai dengan teks input yang akan dianalisis. Lexicon Dictionary adalah kata-kata atau frasa tersebut kemudian dicocokkan dengan entri dalam kamus lexicon, yang berisi kata-kata dan nilai sentimen yang terkait. Sentiment Scores merupakan Setiap kata atau frasa yang cocok dengan entri dalam kamus lexicon diberikan skor sentimen sesuai dengan nilai yang ada di kamus. Score Aggregation adalah Skor-skor sentimen dari semua kata atau frasa dijumlahkan atau dirata-ratakan untuk mendapatkan skor sentimen keseluruhan dari teks. Final Sentiment adalah Hasil akhir berupa sentimen keseluruhan dari teks input, yang bisa berupa positif, negatif, atau netral.

2.4. *Term Frequency -Inverse Document Frequency (TF-IDF)*

TF-IDF adalah salah satu teknik yang digunakan untuk menghitung bobot setiap kata setelah proses ekstraksi fitur. Metode ini membantu menentukan pentingnya kata-kata dalam dokumen dengan memberikan nilai atau bobot tertentu pada masing-masing kata[13]. Pemberian bobot pada setiap kata bertujuan untuk mengukur seberapa sering kata tersebut muncul dalam dokumen. Term Frequency (TF) mengacu pada frekuensi kemunculan kata dalam dokumen tertentu, sedangkan

Inverse Document Frequency (IDF) menghitung bagaimana kata tersebut tersebar di seluruh koleksi dokumen[14]. Berikut adalah rumus dari TF-IDF

$$TF - IDF(t, d) = TF(t, d) \times IDF(t) \quad (1)$$

$$\text{Dimana : } IDF(t) = \log\left(\frac{N}{DF(t)}\right)$$

t	= Kata yang dihitung
d	= bobot kalimat(d)
$TF-IDF(t,d)$	= Kalimat bobot (d) terhadap kata (t)
$TF(t,d)$	= Term Frequency
$IDF(t)$	= Inverse Document Frequency
N	= Jumlah Kalimat
DFt	= Jumlah kata yang diulang

2.5. Lexicon

Support Vector Machine (SVM) adalah salah satu metode klasifikasi yang menggunakan *supervised learning* untuk memprediksi kelas berdasarkan pola dari hasil proses pelatihan. Klasifikasi dilakukan dengan menggunakan *hyperplane* yang memisahkan antara kelas positif dan negatif. *Hyperplane* yang optimal adalah yang memiliki jarak terbesar ke titik data pelatihan terdekat dari masing-masing kelas, karena umumnya, semakin besar margin, semakin rendah *error* generalisasi dari pemisah. *Hyperplane* pemisah terbaik antara dua kelas dapat diperoleh dengan mengukur margin *hyperplane* dan mencari titik maksimalnya[15].

Cara kerja SVM adalah dengan mencari *vektor* pendukung untuk setiap kelas. Sampel kelas yang paling dekat dengan sampel dari kelas lain disebut *support vector*. Setelah mendapatkan *support vector*, SVM menghitung margin. Dengan menggunakan *support vector*, margin dapat dianggap sebagai jalan yang memisahkan dua kelas. *Support vector* berfungsi sebagai batas tepi jalan, juga disebut bahu jalan, dan SVM mencari margin terbesar atau jalan terlebar yang dapat memisahkan dua kelas tersebut[16]. Di dalam metode SVM terdapat empat jenis kernel yang sering digunakan yaitu kernel *linear*, *polynomial*, *radial basis function* (RBF), dan *sigmoid*. Masing-masing kernel memiliki karakteristik dan cara kerja yang berbeda. Berikut adalah penjelasan mengenai masing-masing *kernel*.

1. Kernel Linear

Kernel linear memetakan data ke dalam dimensi fitur yang sama tanpa transformasi *nonlinier*. Digunakan saat data dapat dipisahkan secara *linear* dalam dimensi fitur asli. Parameter bias (C) mengatur posisi batas keputusan SVM, mempengaruhi margin dan letak *hyperplane* yang memisahkan kelas data[17].

2. Kernel Polynomial

Kernel *polynomial* memetakan data ke ruang dimensi yang lebih tinggi untuk mengatasi data yang tidak dapat dipisahkan secara *linear*. Cocok untuk *hard margin* dan *soft margin* dalam menangani pola *nonlinier*. Parameter penting termasuk derajat *polynomial* (d), kemiringan (γ), dan konstanta (r)[17].

3. Kernel RBF (Radial Basis Function)

Kernel RBF menggunakan parameter γ untuk mengatur sebaran kernel, memetakan data ke ruang dimensi tinggi secara *nonlinier*. Penyetelan γ harus hati-hati karena berdampak besar pada kinerja. Sensitif terhadap gangguan dalam data pelatihan jika tidak diperhatikan dengan baik[17].

4. Kernel Sigmoid

Kernel *sigmoid* memungkinkan transformasi data ke bentuk *nonlinier* menggunakan kurva *sigmoid*. Berguna ketika data tidak dapat dipisahkan secara *linear* dalam dimensi fitur asli. Parameter γ (*gamma*) dan konstanta (*r*) perlu diatur secara optimal untuk hasil klasifikasi yang baik. Pengaturan parameter ini sangat mempengaruhi kinerja SVM[17].

Masing-masing kernel akan digunakan untuk menguji performa metode SVM dalam klasifikasi sentimen ulasan teks. Hal tersebut dilakukan untuk mengetahui kernel mana yang optimal digunakan untuk sentimen analisis.

3. HASIL DAN PEMBAHASAN

3.1. Pengumpulan data

Pada tahapan *scrapping* data ini akan melalui proses pengumpulan dataset yang dibutuhkan untuk penelitian. Dataset yang diambil berasal dari ulasan aplikasi *Jogja Smart Service* yang terdapat dari situs resmi *Google Play Store*. Proses dalam mengumpulkan data ini dengan metode *web scrapping* menggunakan bahasa pemrograman berupa *python*. Hasil dari *scrapping* data menghasilkan jumlah ulasan sebanyak 1193 data yang diambil berdasarkan data yang paling relevan. Berikut adalah *output* dari hasil *scraping* dari *google play store* disimpan dalam bentuk *CSV*, berikut adalah sampel data hasil dapat dilihat pada tabel 1.

Tabel 1. Sample Data

Content	Score	At
Aplikasi yang baik untuk memudahkan administrasi warga Kota Yogyakarta dengan Pemerintah Kota Yogyakarta.	5	2020-07-02 17:56:35
Mosok warga luar kota jogja, ndak bisa mendaftarkan ortu yang tinggal di kota mau periksa di puskesmas kota	1	2021-12-01 02:34:19
Sangat membantu warga yang ingin mengurus atau mengetahui banyak hal tentang kota Yogyakarta	5	2020-07-01 00:46:35
Cara mengatasi anda belum melakukan aktivasi jss gimana	1	2022-06-12 17:07:13
Lama proses aktivasinya sudah lebih dari 1 jam gk selesai ² apa jangan ² petugasnya lagii bobok atau apa?	1	2022-10-20 23:42:43
Kurang bagus, jelek, tidak bisa reset password	2	2023-07-15 23:48:10

Dari beberapa atribut yang terdapat pada data ulasan aplikasi *Jogja Smart Service*, hanya 1 atribut yang digunakan dalam penelitian ini yaitu atribut "content" yang nantinya akan digunakan untuk *preprocessing*. Berikut beberapa sampel hasil data atribut "content", dapat dilihat pada gambar 3.

	content
0	Aplikasinya sangat bermanfaat, tapi sayangnya ...
1	Saya sering mengalami trouble saat memasukan e...
2	Sebenarnya aplikasi yang bagus dan sangat berm...
3	Ini sangat tidak efektif sekali, bahkan membua...
4	Keren, ringan di HP lawas juga fungsinya yang ...
...	...
1189	Bagus
1190	Bici
1191	Bagus
1192	Sukses
1193	Mantul

1194 rows x 1 columns

Gambar 3. Atribut *content*

1. Cleaning Text

Pada tahap data cleaning hal pertama yang dilakukan adalah menghapus kolom, simbol. *Emoticon*, dan tanda baca pada komentar, hasil dari *cleaning* data bisa dilihat pada tabel 2 sebagai berikut.

Tabel 2. Proses *cleaning* data

Sebelum <i>Cleanning</i> data	Sesudah <i>Cleanning</i> data
mantap sekali aplikasinya, bisa ngurus surat tanpa harus ke kantor dinas.. bisa dilakukan jarak jauh, semoga update lebih bagus lagi dengan tambah fitur lainnya 👍 👍	mantap sekali aplikasinya bisa ngurus surat tanpa harus ke kantor dinas bisa dilakukan jarak jauh semoga update lebih bagus lagi dengan tambah fitur lainnya

2. Case Folding

Setelah melakukan tahap *cleaning* data, kemudian melakukan tahap *case folding*. Hasil dari tahap *case folding* di tuliskan pada tabel 3.

Tabel 3. Proses *case folding*

Sebelum <i>case folding</i>	Sesudah <i>case folding</i>
mantap sekali aplikasinya bisa ngurus surat tanpa harus ke kantor dinas bisa dilakukan jarak jauh semoga update lebih bagus lagi dengan tambah fitur lainnya	mantap sekali aplikasinya bisa ngurus surat tanpa harus ke kantor dinas bisa dilakukan jarak jauh semoga update lebih bagus lagi dengan tambah fitur lainnya

3. Tokenizing

Setelah melakukan tahap *case folding* kemudian melakukan tahap *Tokenizing*. Hasil dari proses ini ditunjukkan pada tabel 4.

Tabel 4. Proses tokenizing

Sebelum Tokenizing	Sesudah Tokenizing
mantap sekali aplikasinya bisa ngurus surat tanpa harus ke kantor dinas bisa dilakukan jarak jauh semoga update lebih bagus lagi dengan tambah fitur lainnya	'mantap', 'sekali', 'aplikasinya', 'bisa', 'ngurus', 'surat', 'tanpa', 'harus', 'ke', 'kantor', 'dinas', 'bisa', 'dilakukan', 'jarak', 'jauh', 'semoga', 'update', 'lebih', 'bagus', 'lagi', 'dengan', 'tambah', 'fitur', 'lainnya

4. Normalisasi

Setelah melakukan tahap tokenizing kemudian melakukan tahap normalisasi. Hasil tahap normalisasi dapat dilihat pada tabel 5.

Tabel 5. Proses *normalisasi*

Sebelum normalisasi	Sesudah normalisasi
'mantap', 'sekali', 'aplikasinya', 'bisa', 'ngurus', 'surat', 'tanpa', 'harus', 'ke', 'kantor', 'dinas', 'bisa', 'dilakukan', 'jarak', 'jauh', 'semoga', 'update', 'lebih', 'bagus', 'lagi', 'dengan', 'tambah', 'fitur', 'lainnya	'mantap', 'sekali', 'aplikasinya', 'bisa', 'ngurus', 'surat', 'tanpa', 'harus', 'ke', 'kantor', 'dinas', 'bisa', 'dilakukan', 'jarak', 'jauh', 'semoga', 'update', 'lebih', 'bagus', 'lagi', 'dengan', 'tambah', 'fitur', 'lainnya

5. Stopword Removal

Setelah melalui tahap *Normalisasi*, kemudian dilakukan tahap *stopword removal* yang bertujuan untuk menghilangkan kata kata yang tidak bermakna. Pada tahap ini menggunakan *library NLTK* untuk menyediakan kata kata *stopword* dalam bahasa Indonesia dan menentukan *stopword* tambahan secara manual dengan menyesuaikan dengan data. Berikut adalah hasil dari *stopword* ditunjukkan pada tabel 6.

Table 6. Proses *stopword*

Sebelum <i>stopword</i>	Sesudah <i>Stopword</i>
'mantap', 'sekali', 'aplikasinya', 'bisa', 'ngurus', 'surat', 'tanpa', 'harus', 'ke', 'kantor', 'dinas', 'bisa', 'dilakukan', 'jarak', 'jauh', 'semoga', 'update', 'lebih', 'bagus', 'lagi', 'dengan', 'tambah', 'fitur', 'lainnya	'mantap', 'sekali', 'aplikasinya', 'bisa', 'ngurus', 'surat', 'tanpa', 'harus', 'ke', 'kantor', 'dinas', 'bisa', 'dilakukan', 'jarak', 'jauh', 'semoga', 'update', 'lebih', 'bagus', 'lagi', 'dengan', 'tambah', 'fitur', 'lainnya

6. Stemming

Setelah melalui tahap *stopword removal*, melakukan tahap *stemming*. *Stemming* digunakan untuk merubah kata menjadi kata dasar menggunakan bantuan *library Sastrawi*. Berikut adalah hasil dari beberapa yang sudah melalui tahap *stemming* seperti pada tabel 7.

Tabel 7. Proses stemming

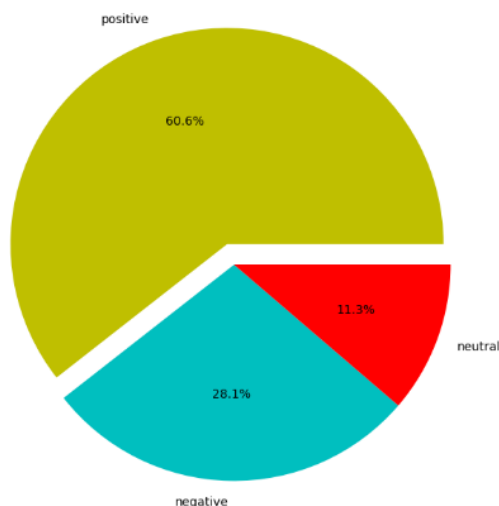
Sebelum stemming	Sesudah stemming
'mantap', 'sekali', 'aplikasinya', 'bisa', 'ngurus', 'surat', 'tanpa', 'harus', 'ke', 'kantor', 'dinas', 'bisa', 'dilakukan', 'jarak', 'jauh', 'semoga', 'update', 'lebih', 'bagus', 'lagi', 'dengan', 'tambah', 'fitur', 'lainnya	mantap', 'sekali', 'aplikasi', 'bisa', 'urus', 'surat', 'tanpa', 'harus', 'ke', 'kantor', 'dinas', 'bisa', 'laku', 'jarak', 'jauh', 'moga', 'update', 'lebih', 'bagus', 'lagi', 'dengan', 'tambah', 'fitur', 'lain'

3.2. Labelling Lexicon

Dalam proses pembobotan, setiap kata dalam kalimat akan diklasifikasikan berdasarkan kamus *leksikon*. Kata-kata dengan sentimen positif akan diberikan skor (+), sedangkan kata-kata dengan sentimen negatif akan diberikan skor (-). Kalimat yang sudah melalui proses pembobotan akan memiliki *polarity score* sentimen dari kata-katanya yang dijumlahkan. Jika total skornya > 0, kalimat tersebut akan diberi label positif. Sebaliknya, jika total skornya < 0, kalimat tersebut akan diberi label negatif. Dan untuk kalimat yang memiliki nilai 0 akan diberi label netral. Berikut adalah hasil dari pembobotan terdapat pada gambar 4

content_preprocessing	polarity_score	polarity
[aplikasi, 'sangat', 'manfaat', 'tapi', 'say...	17	positive
[saya, 'sering', 'alami', 'masalah', 'saat',...	11	positive
['benar', 'aplikasi', 'yang', 'bagus', 'dan', ...	10	positive
['ini', 'sangat', 'tidak', 'efektif', 'sekali'...	-10	negative
['keren', 'ringan', 'di', 'hp', 'lawas', 'juga...	6	positive
['sudah', 'coba', 'isi', 'dan', 'upload', 'ber...	-20	negative
['tiba', 'keluar', 'dan', 'masuk', 'lagi', 'ti...	-2	negative
['dan', 'pada', 'inti', 'untuk', 'urus', 'sura...	-24	negative
['aplikasi', 'kurang', 'protektif', 'untuk', '...	-39	negative
['aplikasi', 'bagus', 'tapi', 'tolong', 'buat'...	17	positive

Gambar 4. Hasil labeling *lexicon*



Gambar 5. Presentase label

Skor hasil pelabelan kelas opini tiap komentar dijumlahkan berdasarkan label positif, negatif, atau netral. Persentase dari hasil penjumlahan tersebut kemudian dihitung. Seperti yang terlihat pada grafik di atas, nilai persentase untuk label positif adalah 60,6%, negatif 28,1%, dan netral 11,3%.

3.3. Pembobotan TF – IDF

Pada tahap ini penerapakan analisis TF-IDF untuk mengidentifikasi istilah-istilah yang paling signifikan dalam setiap dokumen di corpus penelitian. TF-IDF adalah metode yang mengukur seberapa penting suatu kata dalam dokumen tertentu dengan mempertimbangkan frekuensi kemunculannya di dokumen tersebut (Term Frequency) serta kelangkaannya di seluruh corpus (Inverse Document Frequency).

Proses Penghitungan TF-IDF meliputi beberapa tahap. Pertama, Term Frequency (TF) mengukur frekuensi kemunculan sebuah istilah dalam sebuah dokumen. Kedua, Inverse Document Frequency (IDF) mengukur kelangkaan suatu istilah di seluruh dokumen dalam corpus. Skor TF-IDF kemudian dihitung dengan mengalikan nilai TF dengan IDF untuk setiap kata dalam setiap dokumen. Kata-kata dengan skor TF-IDF tinggi dianggap lebih penting karena mereka tidak hanya sering muncul dalam dokumen tertentu tetapi juga jarang muncul dalam dokumen lain dalam corpus. Berikut adalah hasil pembobotan TF-IDF.

Tabel 8. Hasil TF-IDF

kata	score
sekali	0,638522
bantu	0,492119
sangat	0,447812
aplikasi	0,386746
orang	0,000000
password	0,000000
pas	0,000000
parah	0,000000
pakai	0,000000
padahal	0,000000

3.4. Support Vector Machine

Untuk mengetahui performa dari metode *Support Vector Machine* dengan menggunakan 4 kernel yaitu *linier*, *polynomial*, *RBF*, dan *sigmoid*, maka dari itu dilakukan pengujian terhadap model. Kemudian untuk menghitung nilai *precision*, *recall*, dan *F1-score* pada masing masing *kernel*. Performa metode klasifikasi dari setiap kernel dapat dievaluasi menggunakan nilai *precision*, *recall*, dan *F1-score* untuk setiap kelasnya. Nilai *precision*, *recall*, dan *F1-score* berkisar antara 0 hingga 1. Semakin tinggi nilai yang mendekati 1, semakin baik performanya. Hasil keseluruhan dari proses evaluasi model dapat dilihat pada gambar berikut.

1. Kernel Linier

Kernel linear menunjukkan hasil yang cukup baik dengan akurasi sebesar 89%. *Precision* dan *recall* masing-masing mencapai 91% dan 84% dan *F1-score* mencapai 87%. Hal ini menunjukkan bahwa kernel *linear* cukup efektif dalam menangani dataset ulasan aplikasi *Jogja Smart Service* di *Google Play Store*.

2. Kernel Polynomial

Kernel Polynomial menunjukkan hasil yang paling rendah di antara keempat kernel, dengan akurasi sebesar 81%. *Precision* dan *recall* masing-masing mencapai 89% dan 68% dan *f1-score*

mencapai 77%. Hasil ini menunjukkan bahwa kernel *polynomial* kurang efektif untuk dataset ulasan produk yang digunakan dalam penelitian ini. Berikut Gambar 7 menunjukkan hasil evaluasi kernel *Polynomial*.

3. Kernel RBF

Kernel RBF menunjukkan kinerja terbaik di antara keempat kernel yang diuji. Dengan akurasi mencapai 90%, *precision* sebesar 90%, dan *recall* sebesar 90%, dan *f1-score* mencapai 90%, kernel *RBF* menunjukkan kemampuan yang sangat baik dalam menangani data ulasan aplikasi *Jogja Smart Service*.

4. Kernel Sigmoid

Berdasarkan evaluasi yang telah dilakukan, kernel sigmoid memiliki Tingkat akurasi paling rendah diantara kernel lainnya, kernel sigmoid memperoleh nilai akurasi 83%, *precision* 85%, *recall* 78% dan *f1-score* 81%.

3.5. Evaluasi Model

Dalam pengujian metode klasifikasi menggunakan Support Vector Machine (SVM), empat kernel diuji yaitu *linear*, *radial basis function (RBF)*, *polynomial*, dan *sigmoid*. Hasil pengujian masing-masing kernel dievaluasi berdasarkan metrik performa utama yaitu *accuracy*, *precision*, *recall*, dan *F1-score*. Berikut adalah hasil pengujian dalam bentuk tabel

Tabel 9. Hasil Evaluasi

Kernel	Accuracy	Precision	Recall	F1-score
Linier	89%	91%	84%	87%
RBF	90%	90%	90%	90%
Polynomial	81%	89%	68%	77%
Sigmoid	83%	85%	78%	81%

Berdasarkan hasil pengujian ini, RBF sebagai kernel terbaik dengan performa yang paling konsisten dan seimbang dalam hal akurasi, *presisi*, *recall*, dan *F1-score*. Linear kernel juga menunjukkan performa yang baik dan bisa menjadi pilihan yang layak. Kernel *polynomial* dan *sigmoid*, meskipun memiliki nilai presisi yang cukup tinggi, cenderung kurang dalam hal *recall*, yang menyebabkan performa keseluruhan yang lebih rendah. Oleh karena itu, untuk aplikasi praktis, RBF kernel merupakan kernel yang paling baik untuk model klasifikasi ini.

4. KESIMPULAN

Pada penelitian ini, algoritma *Support Vector Machine (SVM)* digunakan untuk analisis sentimen dengan berbagai kernel yang diuji, menghasilkan hasil berikut:

1. Sistem analisis sentimen dalam penelitian ini telah sudah cukup baik dalam mengidentifikasi apakah data tersebut sentimen berlabel positif atau negatif ataupun netral.
2. Pada penelitian ini, algoritma menggunakan SVM pada kernel Linier menghasilkan akurasi sebesar 89%, kernel *Polynomial* menghasilkan akurasi sebesar 81%, kernel RBF menghasilkan akurasi sebesar 90%, kernel Sigmoid menghasilkan akurasi sebesar 82%, Dengan hasil diatas maka dapat disimpulkan kernel *RBF* memiliki akurasi yang lebih baik.

DAFTAR PUSTAKA

- [1] M. G. Gumilar, "Inovasi Pemerintah Daerah Jogja Smart Service Dalam Menciptakan Smart And Liveable City Di Kota Yogyakarta," *Jurnal Gama Societa*, vol. 3, no. 1, p. 19, Apr. 2020, doi: 10.22146/jgs.50012.

- [2] N. W. Indriyani, R. P. Nurdiarti, and M. Nastain, "Aksesibilitas dan Pemanfaatan Aplikasi 'Jogja Smart Service' untuk Mewujudkan Good Governance," *Jurnal Riset Public Relations*, pp. 75–84, Dec. 2022, doi: 10.29313/jrpr.vi.1352.
- [3] N. Nur'aini *et al.*, "Efektivitas Aplikasi JSS Sebagai Implementasi E-Government Kota Yogyakarta," *Gunung Djati Conference Series*, vol. 39, 2024, doi: 10.15575/gdcs.v39i.
- [4] R. Ramadhani, E. P. Purnomo, and A. N. Kasiwi, "E-Government Assessment pada Kualitas Aplikasi Jogja Smart Service (JSS) di Kota Yogyakarta," *Jurnal Pemerintahan dan Politik*, vol. 5, no. 2, Jun. 2020, doi: 10.36982/jpg.v5i2.1031.
- [5] E. R. Kaburuan and N. R. Setiawan, "Sentimen Analisis Review Aplikasi Digital Korlantas Pada Google Play Store Menggunakan Metode SVM," *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 12, no. 1, pp. 105–116, Mar. 2023, doi: 10.32736/sisfokom.v12i1.1614.
- [6] A. Muhammadin and I. A. Sobari, "Analisis Sentimen Pada Ulasan Aplikasi Kredivo Dengan Algoritma SVM Dan NBC," *Jurnal Rekayasa Perangkat Lunak*, vol. 2, no. 2, 2021, [Online]. Available: <http://jurnal.bsi.ac.id/index.php/reputasi>
- [7] P. Aditiya, U. Enri, and I. Maulana, "Analisis Sentimen Ulasan Pengguna Aplikasi Myim3 Pada Situs Google Play Menggunakan Support Vector Machine," *JURIKOM (Jurnal Riset Komputer)*, vol. 9, no. 4, p. 1020, Aug. 2022, doi: 10.30865/jurikom.v9i4.4673.
- [8] U. Kulsum, M. Jajuli, and N. Sulistiyowati, "Analisis Sentimen Aplikasi WETV di Google Play Store Menggunakan Algoritma Support Vector Machine," 2022. [Online]. Available: <http://jurnal.polibatam.ac.id/index.php/JAIC>
- [9] S. Fide, "Analisis Sentimen Ulasan Aplikasi Tiktok Di Google Play Menggunakan Metode Support Vector Machine (Svm) Dan Asosiasi," vol. 10, no. 3, pp. 346–358, 2021, [Online]. Available: <https://ejournal3.undip.ac.id/index.php/gaussian/>
- [10] C. Matulesy and A. Purnama, "Analisis Sentimen Terhadap Review Pengguna Indrive Di Google Playstore Menggunakan Algoritma Support Vector Machine," *Jurnal Darma Agung*, vol. 31, no. 1, p. 1015, Apr. 2023, doi: 10.46930/ojsuda.v31i1.3078.
- [11] R. Mahendrajaya, G. A. Buntoro, and M. B. Setyawan, "url: <http://studentjournal.umpo.ac.id/index.php/komputek> Analisis Sentimen Pengguna Gopay Menggunakan Metode Lexicon Based Dan Support Vector Machine," 2019. [Online]. Available: <http://studentjournal.umpo.ac.id/index.php/komputek>
- [12] F. Amaliah and I. K. Dwi Nuryana, "Perbandingan Akurasi Metode Lexicon Based Dan Naive Bayes Classifier Pada Analisis Sentimen Pendapat Masyarakat Terhadap Aplikasi Investasi Pada Media Twitter," *Journal of Informatics and Computer Science (JINACS)*, vol. 3, no. 03, pp. 384–393, Apr. 2022, doi: 10.26740/jinacs.v3n03.p384-393.
- [13] J. E. Br Sinulingga and H. C. K. Sitorus, "Analisis Sentimen Opini Masyarakat terhadap Film Horor Indonesia Menggunakan Metode SVM dan TF-IDF," *Jurnal Manajemen Informatika (JAMIKA)*, vol. 14, no. 1, pp. 42–53, Feb. 2024, doi: 10.34010/jamika.v14i1.11946.
- [14] S. Khairunnisa, A. Adiwijaya, and S. Al Faraby, "Pengaruh Text Preprocessing terhadap Analisis Sentimen Komentar Masyarakat pada Media Sosial Twitter (Studi Kasus Pandemi COVID-19)," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 5, no. 2, p. 406, Apr. 2021, doi: 10.30865/mib.v5i2.2835.
- [15] F. Romadoni, Y. Umaidah, and B. N. Sari, "Text Mining Untuk Analisis Sentimen Pelanggan Terhadap Layanan Uang Elektronik Menggunakan Algoritma Support Vector Machine," *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 9, no. 2, pp. 247–253, Jul. 2020, doi: 10.32736/sisfokom.v9i2.903.
- [16] S. Ernawati and R. Wati, "Evaluasi Performa Kernel SVM dalam Analisis Sentimen Review Aplikasi ChatGPT Menggunakan Hyperparameter dan VADER Lexicon," 2024.
- [17] S. D. Wahyuni and R. H. Kusumodestoni, "Optimalisasi Algoritma Support Vector Machine (SVM) Dalam Klasifikasi Kejadian Data Stunting," *Bulletin of Information Technology (BIT)*, vol. 5, no. 2, pp. 56–64, 2024, doi: 10.47065/bit.v5i2.1247.