



ANALISIS SENTIMEN APLIKASI POLRI *SUPER APP* MENGUNAKAN ALGORITMA *RANDOM FOREST*

Vicky Fransisco¹, Dionisia Bhisetya Rarasati²

Universitas Bunda Mulia

Jl. Jalur Sutera Barat Kav 7-9, Alam Sutera, Tangerang, Banten

Email: ¹vickyfrans1@gmail.com, ²dionisia.rarasati@gmail.com

Abstract

Polri Super App is an integrated mobile app that can do a lot of things related to the police. Police service covers very important aspects of social life, such as handling criminal cases, regulating traffic, overseeing community activities, and so on. In order to understand the level of satisfaction and expectations of the users of the application, the sentiment analysis of the user of the app is necessary. By analyzing the reviews from the users, it is obtained the sentiment of the community that is expected to be useful for the organizers of the service Polri Super App in order to continue to improve the quality and service of the applications. Random Forest algorithms are used to classify against app reviews. Test results using the confusion matrix showed that the model formed had an accuracy of 95.87%.

Keywords: *Confusion Matrix; Polri Super App; Random Forest; Sentiment Analysis*

PENDAHULUAN

Kemajuan teknologi informasi dan komunikasi telah mengalami pertumbuhan cepat dan mendorong perubahan dalam berbagai sektor, termasuk pelayanan publik. Penggunaan teknologi informasi dan komunikasi dalam layanan publik bertujuan untuk meningkatkan kinerja secara lebih efisien, efektif, transparan, dan akuntabel. Salah satu sektor yang mengalami transformasi signifikan adalah pelayanan yang disediakan oleh institusi kepolisian. Pelayanan kepolisian mencakup berbagai aspek yang sangat penting dalam kehidupan bermasyarakat, seperti menangani kasus kriminal, mengatur lalu lintas, mengawasi kegiatan masyarakat, dan lainnya. Tugas utama kepolisian dalam menjaga keamanan dan ketertiban masyarakat memerlukan tanggung jawab besar untuk menyajikan layanan yang cepat, efektif, dan profesional bagi masyarakat. Dalam upaya meningkatkan kualitas dan efektivitas pelayanan kepolisian, maka teknologi informasi dan komunikasi mulai dimanfaatkan. Salah satu inovasi terbaru dalam hal ini adalah meluncurkan sistem pelayanan baru yaitu sebuah aplikasi bernama *Polri Super App* (Jayamuna, 2023).

Aplikasi *Polri Super App* ini diluncurkan pada 28 April 2022. Aplikasi *Polri Super App* merupakan aplikasi mobile terintegrasi yang dapat melakukan banyak hal seperti registrasi kendaraan, perpanjangan STNK, perpanjangan SIM Nasional, penerbitan SIM Internasional, pengaduan masyarakat, pengecekan e-tilang, konfirmasi ETLE, pemantauan SP2HP (Surat Pemberitahuan Perkembangan Hasil Penyidikan), serta akses kepada informasi terkait Kamtibmas, Humas, Peta, dan lainnya. Dilansir dari Kompas.com, Direktur Intelijen dan Keamanan (Dirintelkam) Polda Jawa Barat Kombes Asep Nalaludin menerangkan, pada dasarnya

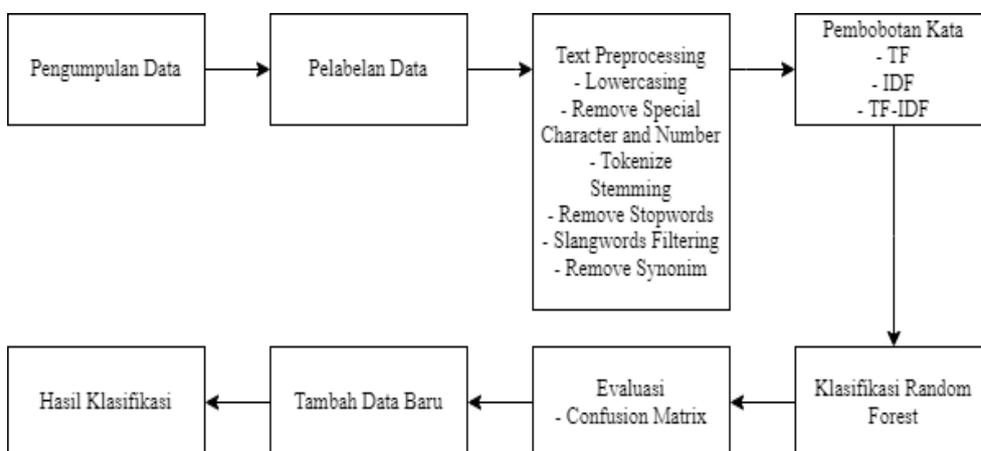
permohonan SKCK oleh masyarakat wajib secara online melalui aplikasi *Super Apps* Presisi Polri.

Seiring dengan penggunaan dan meningkatnya penggunaan aplikasi Polri *Super App*, penting untuk mengetahui tingkat kepuasan masyarakat terhadap layanan ini. Untuk dapat memahami tingkat kepuasan dan ekspektasi dari pengguna aplikasi, diperlukan adanya analisis sentimen berdasarkan ulasan dari pengguna aplikasi. Analisis sentimen adalah metode untuk menggali dan memahami sentimen atau opini yang terkandung dalam data teks. Teknik ini bertujuan untuk mengidentifikasi apakah sentimen tersebut bersifat positif, netral, atau negatif (Sari & Wibowo, 2019). Dengan menganalisis ulasan dari pengguna aplikasi, maka didapatkan sentimen masyarakat yang diharapkan dapat berguna bagi penyelenggara layanan Polri *Super App* agar dapat terus meningkatkan kualitas dan layanan aplikasi.

Algoritma *Random Forest* akan digunakan dalam penelitian ini. Berdasarkan penelitian terdahulu, Ahmad Miftahusalam, Hasih Pratiwi, dan Isnandar Slamet membandingkan metode *Random Forest* dan *Naïve Bayes* dalam menganalisis sentimen terhadap aplikasi BCA Mobile. Hasilnya menunjukkan bahwa metode *Random Forest* memiliki tingkat akurasi yang lebih tinggi dibandingkan dengan metode *Naïve Bayes*, dengan tingkat akurasi sebesar 93,93% untuk *Random Forest* dan 92,31% untuk *Naïve Bayes*(Miftahusalam dkk., 2023). Evita Fitri, Yuri Yuliani, Susy Rosyida, Windu Gata melakukan analisis sentimen terhadap aplikasi Ruangguru menggunakan algoritma *Naïve Bayes*, *Random Forest* dan *Support Vector Machine*. Hasil penelitian menunjukkan bahwa *Random Forest* memiliki tingkat akurasi tertinggi, yaitu 97,16%, diikuti oleh *Support Vector Machine* dengan akurasi 96,01%, dan *Naïve Bayes* memiliki akurasi terendah, yaitu 94,16%(Fitri dkk., 2020). Muhammad Rusdi Rahman, Ahmad Febri Diansyah, dan Hanafi melakukan analisis sentimen terhadap aplikasi Shopee menggunakan metode klasifikasi *Random Forest*. Penelitian tersebut menghasilkan kesimpulan bahwa metode *Random Forest* dapat menghasilkan tingkat akurasi sebesar 94% dalam menganalisis sentimen aplikasi Shopee(Rusdi Rahman & Febri Diansyah, 2024).

METODE PENELITIAN

Penelitian ini akan melakukan analisis sentimen terhadap ulasan pengguna dari aplikasi Polri *Super App* di *Google Play Store* menggunakan algoritma *Random Forest*. Tahapan-tahapan yang akan dilalui dapat dilihat pada Gambar 1.



Gambar 1. Rangkaian Penelitian

Pengumpulan Data

Tahap awal dari penelitian ini adalah mengumpulkan data berupa ulasan pengguna aplikasi Polri Super App dari Google Play Store.

Pelabelan Data

Tahap selanjutnya setelah mengumpulkan data ulasan adalah melakukan pelabelan terhadap setiap ulasan. Pelabelan data akan dilakukan oleh peneliti secara manual dengan ketentuan ulasan dengan sentimen negatif akan diberi label 0 dan sentimen positif akan diberi label 1.

Text Preprocessing

Text Preprocessing merupakan langkah penting dalam algoritma text mining yang bertujuan untuk mengolah teks menjadi bentuk yang lebih terstruktur sehingga dapat diproses lebih lanjut dalam proses text mining. Tahap preprocessing dalam klasifikasi teks berperan penting dalam meningkatkan akurasi klasifikasi data (Ridwansyah, 2022). Adapun tahapan yang dilakukan dalam tahap ini adalah sebagai berikut(Alfonso & Bhisetya Rarasati, 2023):

Lowercasing

Lowercasing merupakan tahapan dari text preprocessing yang memiliki tujuan untuk mengubah huruf kapital pada kalimat-kalimat dalam sebuah dokumen menjadi huruf kecil sehingga semua data yang diperoleh menjadi huruf kecil (Komarudin dkk., 2024).

Tabel 1. Proses Lowercasing

Sebelum	Sesudah
Aplikasinya keren, Sangat membantu masyarakat sekali	aplikasinya keren, sangat membantu masyarakat sekali
Aplikasi ini sangat keren dan sangat membantu sekali dalam mengurus berkas di kepolisian	aplikasi ini sangat keren dan sangat membantu sekali dalam mengurus berkas di kepolisian
Terlalu ribet dan juga sangat lemot apk nya	terlalu ribet dan juga sangat lemot apk nya
Gabisa login verifikasi kode email nya lambat selalu gabisa masuk	gabisa login verifikasi kode email nya lambat selalu gabisa masuk
Verifikasi email nya ga masuk2. Udh di coba beberapa kali.. Belum apa2 udh eror gini	verifikasi email nya ga masuk2. udh di coba beberapa kali.. belum apa2 udh eror gini

Remove Special Character and Number

Remove Special Character and Number merupakan tahapan dari text preprocessing yang memiliki tujuan untuk menghasilkan kalimat pada sebuah dokumen tanpa tanda baca, emotikon, dan angka(Alfonso & Bhisetya Rarasati, 2023).

Tabel 2. Proses Remove Special Character and Number

Sebelum	Sesudah
aplikasinya keren, sangat membantu masyarakat sekali	aplikasinya keren sangat membantu masyarakat sekali
aplikasi ini sangat keren dan sangat membantu sekali dalam mengurus berkas di kepolisian	aplikasi ini sangat keren dan sangat membantu sekali dalam mengurus berkas di kepolisian

terlalu ribet dan juga sangat lemot apk nya	terlalu ribet dan juga sangat lemot apk nya
gabisa login verifikasi kode email nya lambat selalu gabisa masuk	gabisa login verifikasi kode email nya lambat selalu gabisa masuk
verifikasi email nya ga masuk2. udh di coba beberapa kali.. belum apa2 udh eror gini	verifikasi email nya ga masuk udh di coba beberapa kali belum apa udh eror gini

Tokenize

Tokenize merupakan proses pemecahan kalimat menjadi kata-kata tersendiri (Bhisetya Rarasati, 2020). Proses ini dilakukan dengan tujuan agar proses selanjutnya dapat diterapkan pada tingkat kata (Alfonso & Bhisetya Rarasati, t.t.).

Tabel 3. Proses *Tokenize*

Sebelum	Sesudah
aplikasinya keren sangat membantu masyarakat sekali	['aplikasinya', 'keren', 'sangat', 'membantu', 'masyarakat', 'sekali']
aplikasi ini sangat keren dan sangat membantu sekali dalam mengurus berkas berkas di kepolisian	['aplikasi', 'ini', 'sangat', 'keren', 'dan', 'sangat', 'membantu', 'sekali', 'dalam', 'mengurus', 'berkas', 'berkas', 'di', 'kepolisian']
terlalu ribet dan juga sangat lemot apk nya	['terlalu', 'ribet', 'dan', 'juga', 'sangat', 'lemot', 'apk', 'nya']
gabisa login verifikasi kode email nya lambat selalu gabisa masuk	['gabisa', 'login', 'verifikasi', 'kode', 'email', 'nya', 'lambat', 'selalu', 'gabisa', 'masuk']
verifikasi email nya ga masuk udh di coba beberapa kali belum apa udh eror gini	['verifikasi', 'email', 'nya', 'ga', 'masuk', 'udh', 'di', 'coba', 'beberapa', 'kali', 'belum', 'apa', 'udh', 'eror', 'gini']

Stemming

Stemming merupakan proses pemisahan imbuhan yang berupa awalan, sisipan, akhiran, dan konfiks kata turunan menjadi kata-kata penting (Prakoso dkk., 2022). Dalam teks berbahasa Indonesia, proses stemming sangat diperlukan karena Bahasa Indonesia memiliki awalan, sisipan, akhiran, dan konfiks yang membuat suatu kata dasar dapat diubah ke dalam banyak bentuk dan mengakibatkan pencarian data menjadi lebih sulit (Bhisetya Rarasati, 2021).

Tabel 4. Proses *Stemming*

Sebelum	Sesudah
['aplikasinya', 'keren', 'sangat', 'membantu', 'masyarakat', 'sekali']	['aplikasi', 'keren', 'sangat', 'bantu', 'masyarakat', 'sekali']
['aplikasi', 'ini', 'sangat', 'keren', 'dan', 'sangat', 'membantu', 'sekali', 'dalam', 'mengurus', 'berkas', 'berkas', 'di', 'kepolisian']	['aplikasi', 'ini', 'sangat', 'keren', 'dan', 'sangat', 'bantu', 'sekali', 'dalam', 'urus', 'berkas', 'berkas', 'di', 'polisi']

['terlalu', 'ribet', 'dan', 'juga', 'sangat', 'lemot', 'apk', 'nya']	['terlalu', 'ribet', 'dan', 'juga', 'sangat', 'lemot', 'apk', 'nya']
['gabisa', 'login', 'verifikasi', 'kode', 'email', 'nya', 'lambat', 'selalu', 'gabisa', 'masuk']	['gabisa', 'login', 'verifikasi', 'kode', 'email', 'nya', 'lambat', 'selalu', 'gabisa', 'masuk']
['verifikasi', 'email', 'nya', 'ga', 'masuk', 'udh', 'di', 'coba', 'beberapa', 'kali', 'belum', 'apa', 'udh', 'eror', 'gini']	['verifikasi', 'email', 'nya', 'ga', 'masuk', 'udh', 'di', 'coba', 'beberapa', 'kali', 'belum', 'apa', 'udh', 'eror', 'gini']

a. Remove Stopwords

Remove Stopwords merupakan rangkaian dari analisis teks yang berfungsi untuk mengeliminasi kata-kata yang dianggap kurang memberikan kontribusi signifikan terhadap makna dari isi ulasan dalam dokumen. Kata yang dieliminasi disebut kata penghubung atau stopword. Hasil dari filter stopwords adalah data yang semakin ringkas karena penghilangan semua kata penghubung (Komarudin dkk., 2024).

Tabel 5. Proses *Remove Stopwords*

Sebelum	Sesudah
['aplikasinya', 'keren', 'sangat', 'membantu', 'masyarakat', 'sekali']	['aplikasi', 'keren', 'bantu', 'masyarakat']
['aplikasi', 'ini', 'sangat', 'keren', 'dan', 'sangat', 'bantu', 'sekali', 'dalam', 'urus', 'berkas', 'berkas', 'di', 'polisi']	['aplikasi', 'keren', 'bantu', 'urus', 'berkas', 'berkas']
['terlalu', 'ribet', 'dan', 'juga', 'sangat', 'lemot', 'apk', 'nya']	['ribet', 'lemot', 'apk']
['gabisa', 'login', 'verifikasi', 'kode', 'email', 'nya', 'lambat', 'selalu', 'gabisa', 'masuk']	['gabisa', 'login', 'verifikasi', 'kode', 'email', 'lambat', 'gabisa', 'masuk']
['verifikasi', 'email', 'nya', 'ga', 'masuk', 'udh', 'di', 'coba', 'beberapa', 'kali', 'belum', 'apa', 'udh', 'eror', 'gini']	['verifikasi', 'email', 'ga', 'masuk', 'udh', 'coba', 'kali', 'udh', 'eror', 'gini']

b. Slangwords Filtering

Slangwords Filtering merupakan proses menghilangkan frasa yang tidak formal dari teks dan merubahnya menjadi kata-kata baku (Alfonso & Bhisetya Rarasati, t.t.).

Tabel 6. Proses *Slangwords Filtering*

Sebelum	Sesudah
['aplikasi', 'keren', 'bantu', 'masyarakat']	['aplikasi', 'keren', 'bantu', 'masyarakat']

['aplikasi', 'keren', 'bantu', 'urus', 'berkas', 'berkas']	['aplikasi', 'keren', 'bantu', 'urus', 'berkas', 'berkas']
['ribet', 'lemot', 'apk']	['sulit', 'lambat', 'aplikasi']
['gabisa', 'login', 'verifikasi', 'kode', 'email', 'lambat', 'gabisa', 'masuk']	['tidak', 'bisa', 'masuk', 'verifikasi', 'kode', 'email', 'lambat', 'tidak', 'bisa', 'masuk']
['verifikasi', 'email', 'ga', 'masuk', 'udh', 'coba', 'kali', 'udh', 'eror', 'gini']	['verifikasi', 'email', 'tidak', 'masuk', 'sudah', 'coba', 'kali', 'sudah', 'gangguan', 'gini']

Remove Synonim

Dalam proses *text preprocessing*, *remove synonim* memungkinkan penggabungan dua kata yang berbeda tetapi memiliki arti yang sama menjadi satu kata tanpa mengubah frekuensi kata (Bhisetya Rarasati, 2020).

Tabel 7. Proses *Remove Synonim*

Sebelum	Sesudah
['aplikasi', 'keren', 'bantu', 'masyarakat']	aplikasi keren bantu masyarakat
['aplikasi', 'keren', 'bantu', 'urus', 'berkas', 'berkas']	aplikasi keren bantu urus berkas berkas
['sulit', 'lambat', 'aplikasi']	sulit lambat aplikasi
['tidak', 'bisa', 'masuk', 'verifikasi', 'kode', 'email', 'lambat', 'tidak', 'bisa', 'masuk']	tidak bisa masuk verifikasi kode email lambat tidak bisa masuk
['verifikasi', 'email', 'tidak', 'masuk', 'sudah', 'coba', 'kali', 'sudah', 'gangguan', 'gini']	verifikasi email tidak masuk sudah coba kali sudah gangguan gini

Pembobotan Kata

Term Frequency – Inverse Document Frequency (TF-IDF) adalah metode untuk memberikan bobot pada hubungan antara suatu kata dan dokumen dalam analisis teks. Metode TF-IDF menggunakan gabungan dari dua konsep utama, yaitu frekuensi kata yang muncul dalam suatu dokumen (*Term Frequency*) dan kebalikan dari frekuensi dokumen yang mengandung kata-kata tersebut (*Inverse Document Frequency*). Bobot hubungan antara sebuah kata dan sebuah dokumen akan tinggi jika frekuensi kata tersebut tinggi dalam dokumen tersebut tetapi rendah dalam keseluruhan dokumen dalam koleksi yang lebih besar (Nurjannah dkk., 2013). Berikut adalah rumus untuk menentukan TF, IDF, dan TF-IDF (Alfonso & Bhisetya Rarasati, 2020):

$$TF_t = (t, d)$$

$$IDF_t = \log \frac{N}{df(t)} + 1$$

$$W_t = TF_t * IDF_t$$

Klasifikasi Random Forest

Random Forest adalah metode pembelajaran mesin yang menggabungkan gabungan *Decision Tree* untuk dilakukan klasifikasi dimana *majority voting* digunakan untuk membuat keputusan akhir (Larasati dkk., 2022). *Random Forest* memanfaatkan metode *bootstrap aggregating (bagging)* dan seleksi fitur acak (*random feature selection*) dalam penerapannya.

Semakin banyak penggunaan tree yang ada maka akan semakin akurat hasilnya. *Random Forest Classifier* memiliki kemampuan untuk memprediksi kategori terkait dengan estimasi melalui analisis berbagai ulasan, dan dalam beberapa kasus, model ini dapat dikalibrasi untuk menghasilkan probabilitas output. Berikut adalah cara kerja dari algoritma *random forest* (Putra, 2019):

1. Tahap pertama dalam penggunaan metode *bootstrap* adalah melakukan penarikan sampel secara acak berukuran n dari data yang tersedia.
2. Dengan menggunakan data hasil *bootstrap*, dibangunlah *decision tree* menggunakan subset fitur yang diambil secara acak (*random feature selection*) untuk melakukan *training*.
3. Mengulangi langkah 1 dan 2 sebanyak k kali, di mana k adalah jumlah pohon keputusan yang ingin dibuat.
4. Hasil prediksi dari seluruh *decision tree* akan dikumpulkan dan dilakukan *majority voting* untuk mendapatkan hasil akhir.

Pada algoritma *Decision Tree*, pengambilan keputusan untuk melakukan split simpul biasanya menggunakan perhitungan terhadap *entropy* dan *information gain*. *Entropy* mengukur ketidakpastian dalam sebuah himpunan data, sedangkan *information gain* mengukur penurunan *entropy* setelah pembagian data. Tujuan dari pemilihan fitur untuk split adalah untuk memaksimalkan *information gain*, yang mengindikasikan seberapa banyak informasi yang diperoleh dari membagi simpul berdasarkan fitur tersebut. Sementara itu, pada algoritma *Random Forest*, pengambilan keputusan untuk melakukan split simpul biasanya menggunakan *Gini Index*. *Gini Index* mengukur kemurnian (*purity*) himpunan data. Semakin rendah nilai *Gini Index*, semakin baik kemurnian dari sebuah simpul. Ketika memilih fitur untuk melakukan *split*, *Random Forest* akan memilih fitur yang menghasilkan simpul dengan nilai *Gini Index* yang paling rendah, sehingga meningkatkan kemurnian dari simpul-simpul tersebut.

Evaluasi

Tahap berikutnya adalah melakukan evaluasi dari data uji. Evaluasi akan dilakukan berdasarkan *confusion matrix* untuk mendapatkan nilai akurasi dari hasil pengujian terhadap data uji terhadap model yang dibentuk. *Confusion Matrix* adalah sebuah matriks yang berisi nilai prediksi benar dan prediksi salah dari suatu model klasifikasi. Dalam pengujian keakuratan hasil pencarian, evaluasi dilakukan menggunakan nilai *recall*, *precision*, dan *accuracy*. Dimana *precision* mengukur kemampuan sistem untuk menemukan peringkat yang paling relevan dari dokumen yang diambil. Ini didefinisikan sebagai sebagian dari dokumen yang diambil yang benar-benar relevan terhadap *query*. *Recall* mengukur kemampuan sistem untuk menemukan semua item yang relevan dari koleksi dokumen. Ini didefinisikan sebagai sebagian dari dokumen yang relevan terhadap *query* yang berhasil diambil. *Accuracy* merupakan perbandingan antara kasus yang diidentifikasi benar dengan jumlah seluruh kasus dan *error rate* merupakan kasus yang diidentifikasi salah dengan jumlah seluruh kasus (Locarso, 2022). Berikut persamaan yang digunakan untuk mendapatkan nilai akurasi (Bhisetya Rarasati, 2021):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Dimana TP merupakan *True Positive*, FP merupakan *False Positive*, TN merupakan *True Negative*, dan FN merupakan *False Negative* yang didapatkan dari *confusion matrix* yang dapat dilihat pada tabel dibawah.

Tabel 8. *Confusion Matrix*

	<i>Prediction Positive</i>	<i>Prediction Negative</i>
<i>Actual Positive</i>	TP	FN
<i>Actual Negative</i>	FP	TN

Tambah Data Baru

Setelah model *random forest* berhasil dibentuk dan diuji terhadap data uji, selanjutnya adalah mencoba untuk menambahkan data baru yang tidak termasuk ke dalam data latih dan data uji. Pengujian dilakukan untuk mengetahui apakah model yang dibentuk dapat memberikan hasil prediksi sesuai yang diharapkan.

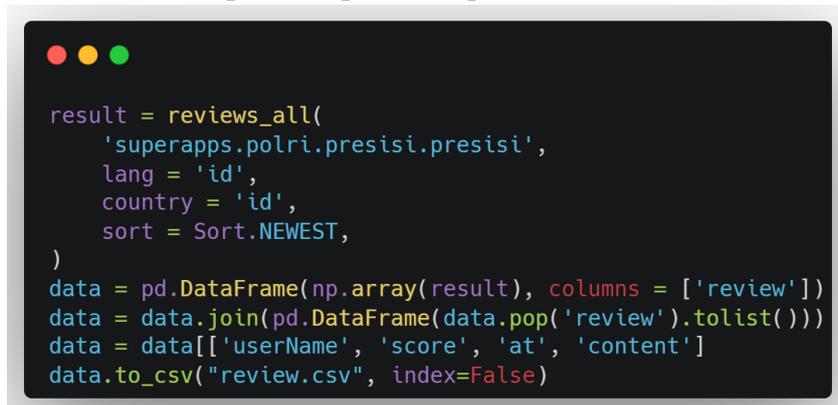
Hasil Klasifikasi

Setelah melakukan penambahan data baru, maka model akan melakukan prediksi terhadap ulasan yang baru ditambahkan. Model akan menghasilkan prediksi berupa 0 atau 1, dimana jika model menghasilkan prediksi 0 maka ulasan diprediksi negatif, sedangkan jika model menghasilkan prediksi 1 maka ulasan diprediksi positif.

HASIL DAN PEMBAHASAN

Pengumpulan Data

Pengumpulan data ulasan dilakukan menggunakan bahasa pemrograman *python* dan menggunakan *library google-play-scraper*. Data ulasan yang digunakan dalam penelitian ini merupakan ulasan yang diunggah pada tanggal 1 Januari 2024 hingga 29 Februari 2024. Hasil pengumpulan data ulasan mendapatkan total 1100 data ulasan. Proses ini akan menghasilkan kumpulan data yang berisi nama pengguna, rating, tanggal ulasan, dan ulasan yang disimpan ke dalam file berekstensi csv. Skrip kode dapat dilihat pada Gambar 2.

A screenshot of a code editor showing Python code for data collection. The code uses the 'reviews_all' function from 'superapps.polri.presisi.presisi' to fetch reviews, sorts them by 'NEWEST', and then uses 'pd.DataFrame' to create a DataFrame. The DataFrame is then joined with another DataFrame and saved to a CSV file named 'review.csv' using 'data.to_csv'.

```
result = reviews_all(
    'superapps.polri.presisi.presisi',
    lang = 'id',
    country = 'id',
    sort = Sort.NEWEST,
)
data = pd.DataFrame(np.array(result), columns = ['review'])
data = data.join(pd.DataFrame(data.pop('review').tolist()))
data = data[['userName', 'score', 'at', 'content']]
data.to_csv("review.csv", index=False)
```

Gambar 2. Pengumpulan Data

Setelah kode pada gambar 2 dijalankan, maka akan tersimpan file berekstensi csv dan berikut adalah beberapa ulasan yang berhasil diambil.

Unnamed: 0	id	userName	score	at	content
0	0	Garda Yodi	1	2024-02-29 05:58:52	Datanya tidak terintergrasi, percuma daftar sk...
1	1	Kha Damfi	1	2024-02-29 05:55:56	aplikasi keren gaada guna nyaaa
2	2	Noverly Suvandi,S.h	5	2024-02-29 05:44:28	Ok dan sangat membantu maju berjaya terus Polri
3	3	Wiwit wijayanto	2	2024-02-29 04:57:47	Saya sudah tf untuk pembuatan sim trus dari ap...
4	4	Tia Wati	1	2024-02-29 04:29:52	Udah dateng ke polres buat bikin skck malah di...
5	5	Ahmad Rofiki	1	2024-02-29 04:25:47	cacatttt
6	6	Dani Ramdani	1	2024-02-29 03:14:37	Iya nih, aplikasi belum sempurna
7	7	Tono aje	2	2024-02-29 02:21:31	Gak semu polres bisa udah terlanjur bayar haru...
8	8	Praise Damai	3	2024-02-28 17:45:03	Kelemahan aplikasi ini ketika saya mengurus SK...
9	9	Muhammad Hartawan	1	2024-02-28 16:43:55	team IT pecat aja ga bener bikin aplikasi..

Gambar 3. Data Ulasan

Pelabelan Data

Ulasan akan dilakukan pelabelan secara manual dan mendapatkan sebanyak 786 komentar dengan sentimen negatif, 309 komentar dengan sentimen positif, dan sebanyak 71 komentar dengan sentimen netral. Komentar dengan sentiment netral akan dihilangkan, sehingga hanya terdapat 2 label, yaitu positif dan negatif.

Tabel 9. Hasil Pelabelan

Komentar	Label
Aplikasinya keren, Sangat membantu masyarakat sekali	Positif
Aplikasi ini sangat keren dan sangat membantu sekali dalam mengurus berkas berkas di kepolisian	Positif
Terlalu ribet dan juga sangat lemot apk nya	Negatif
Gabisa login verifikasi kode email nya lambat selalu gabisa masuk	Negatif
Verifikasi email nya ga masuk2. Udh di coba beberapa kali.. Belum apa2 udh eror gini	Negatif

Text Preprocessing

Setelah melakukan pelabelan terhadap ulasan, *text* akan memasuki tahap *text preprocessing* untuk untuk mengubah bentuk dokumen menjadi lebih terstruktur. Data ulasan akan melalui beberapa tahapan *text preprocessing* seperti *lowercasing* untuk mengubah menjadi seluruh teks menjadi huruf kecil, *remove special character and number* untuk menghapus karakter spesial dan angka, *tokenize* untuk memecah kalimat menjadi kata, *stemming* untuk mengubah kata menjadi bentuk dasarnya, *remove stopwords* untuk menghapus kata hubung, *slangwords filtering* untuk mengubah kata-kata gaul menjadi kata baku, dan *remove synonym* untuk menggabungkan kata yang memiliki arti sama.

Pembobotan Kata

Setelah proses *text preprocessing* selesai, langkah selanjutnya adalah melakukan pembobotan kata. Pembobotan kata akan menggunakan metode *Term Frequency – Inverse Document Frequency* (TF-IDF). Skrip kode untuk melaksanakan pembobotan kata dapat ditemukan pada Gambar 4.

```
tfidf_vectorizer = TfidfVectorizer()  
x_train_tfidf = tfidf_vectorizer.fit_transform(x_train)  
x_test_tfidf = tfidf_vectorizer.transform(x_test)  
joblib.dump(tfidf_vectorizer, 'tfidf_vectorizer.pkl')
```

Gambar 4. TF-IDF

Data ulasan yang telah dibobotkan kata akan selanjutnya dibagi menjadi data latih dan data uji dengan perbandingan 80:20, di mana terdapat 868 data latih dan 218 data uji. Pembagian data ini terlihat dalam Gambar 5.

```
x = data['final']  
y = data['sentiment_label']  
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.20)
```

Gambar 5. Pembagian Data

Klasifikasi *Random Forest*

Setelah melakukan pembagian data menjadi data latih dan data uji, tahap berikutnya adalah melakukan pemodelan menggunakan algoritma *random forest*. Skrip kode dapat dilihat pada Gambar 6.

```
random_forest_classifier = RandomForestClassifier()  
random_forest_classifier.fit(x_train_tfidf, y_train)  
y_pred_rf = random_forest_classifier.predict(x_test_tfidf)  
joblib.dump(random_forest_classifier, 'random_forest_model.pkl')
```

Gambar 6. *Random Forest*

Evaluasi

Hasil penelitian ini diukur dengan nilai akurasi yang didapatkan berdasarkan *confusion matrix* yang dapat dilihat pada Tabel 10.

Tabel 10. Confusion Matrix Hasil Klasifikasi

	Prediction Positive	Prediction Negative
Actual Positive	54	3
Actual Negative	6	155

Tabel 10 menunjukkan hasil pengujian model yang telah dibentuk terhadap *dataset* uji. Terdapat 54 data ulasan yang termasuk ke dalam kategori *True Positive* dimana hasil prediksi dan aktualnya adalah positif, 155 data ulasan yang termasuk ke dalam kategori *True Negatif* dimana hasil prediksi sama dengan aktualnya yaitu negatif. Terdapat pula kesalahan dalam klasifikasi sebanyak 3 data ulasan yang termasuk dalam kategori *False Negative* dikarenakan hasil

Analisis Sentimen Aplikasi Polri *Super App* Menggunakan Algoritma *Random Forest*

prediksinya adalah negatif sedangkan aktualnya adalah positif, dan sebanyak 6 data ulasan yang termasuk dalam kategori *False Positive* dikarenakan hasil prediksinya adalah positif sedangkan aktualnya adalah negatif. Berdasarkan *confusion matrix* yang telah dibuat, maka dapat dilakukan perhitungan untuk mendapatkan nilai akurasi. Berikut adalah perhitungan untuk mendapatkan nilai akurasi menggunakan rumus pada persamaan (4) yang sudah ditulis sebelumnya:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Accuracy = \frac{54 + 155}{54 + 155 + 6 + 3}$$

$$Accuracy = \frac{209}{218}$$

$$Accuracy = 0.9587 = 95.87\%$$

Hasil dari pengujian model terhadap data uji menggunakan algoritma *random forest* mendapatkan nilai akurasi sebesar 0.9587 atau 95.87%. Hasil tersebut menunjukkan bahwa ada 209 data ulasan yang diprediksi benar dari 218 data yang diuji.

Tambah Data Baru

Untuk menambahkan data baru berupa ulasan yang ingin diprediksi sentimennya, pada penelitian ini membuat sebuah website untuk dapat menginputkan data berupa *teks* ataupun *file* berekstensi *.txt* yang berisi kumpulan ulasan.

Hasil Klasifikasi

Setelah data ulasan baru di tambahkan, ulasan tersebut akan menjalankan beberapa tahapan seperti *text preprocessing*, pembobotan kata, dan dilakukan prediksi menggunakan model yang telah dibentuk. Skrip kode dapat dilihat pada gambar 7.

```
def predict_sentiment(input_text):
    tfidf_vectorizer = joblib.load('tfidf_vectorizer.pkl')
    loaded_model = joblib.load('random_forest_model.pkl')

    input_text_lower = input_text.lower()
    cleaned_text = clean_text(input_text_lower)
    tokenized_text = word_tokenize(cleaned_text)
    nltk.download('stopwords')
    stop = []
    stop = stopwords.words('indonesian')
    additional_stopwords = read_additional_stopwords()
    stop.extend(additional_stopwords)
    stemmed_text = stemming(tokenized_text)
    removed_stopwords = [word for word in stemmed_text if word not in stop]
    replaced_slang = replace_word_elongation(" ".join(removed_stopwords)).split()
    replaced_slang = replace_slang(" ".join(replaced_slang)).split()
    synonyms_dict = read_additional_synonyms()
    final_text = replace_synonyms(replaced_slang, synonyms_dict)

    input_tfidf = tfidf_vectorizer.transform([final_text])

    predicted_sentiment = loaded_model.predict(input_tfidf)

    return predicted_sentiment[0], input_text_lower, cleaned_text, tokenized_text, stemmed_text, removed_stopwords,
    replaced_slang, final_text
```

Gambar 7. Prediksi Data Baru

KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, maka dapat ditarik kesimpulan bahwa algoritma *Random Forest* dapat melakukan klasifikasi dengan baik untuk memberikan label pada ulasan dengan tingkat akurasi 95.87%. Model yang dibentuk dapat memprediksi sebanyak 209 data ulasan dengan benar dari 218 data yang diuji. Selain itu, dapat dilihat bahwa banyak pengguna yang memberikan ulasan negatif, yaitu mencapai 786 komentar yang dapat menunjukkan ketidakpuasan pengguna terhadap aplikasi Polri *Super App* dan terdapat pula komentar yang menunjukkan kepuasan pengguna terhadap aplikasi ini sebanyak 314 komentar.

DAFTAR PUSTAKA

- Alfonso, M., & Bhisetya Rarasati, D. (t.t.). *JISA (Jurnal Informatika dan Sains) Sentiment Analysis of 2024 Presidential Candidates Election Using SVM Algorithm*.
- Bhisetya Rarasati, D. (2020). A Grouping of Song-Lyric Themes Using K-Means Clustering. Dalam *Jurnal Informatika dan Sains) e-ISSN* (Vol. 03, Nomor 02).
- Bhisetya Rarasati, D. (2021). Recommendation for Classification of News Categories Using Support Vector Machine Algorithm with SVD. *Ultimatics : Jurnal Teknik Informatika*, 13(2), 72. www.waspada.com,
- Fitri, E., Yuliani, Y., Rosyida, S., & Gata, W. (2020). Analisis Sentimen Terhadap Aplikasi Ruangguru Menggunakan Algoritma Naive Bayes, Random Forest Dan Support Vector Machine. *TRANSFORMTIKA*, 18(1), 71–80. www.nusamandiri.ac.id,
- Jayamuna, I. M. (2023). SISTEM PELAYANAN KEPOLISIAN MELALUI APLIKASI POLRI SUPER APP DI POLDA LAMPUNG SEBAGAI INOVASI PELAYANAN PUBLIK. *Jurnal Socia Logica*, 3(1), 33–43. <https://doi.org/10.572349/SOCIALOGICA.V3I1.338>
- Komarudin, A., Meutia Hilda, A., & Author, C. (2024). Analisis Sentimen Ulasan Aplikasi Identitas Kependudukan Digital Pada Play Store Menggunakan Metode Naive Bayes. Dalam *Computer Science (CO-SCIENCE)* (Vol. 4, Nomor 1). <http://jurnal.bsi.ac.id/index.php/co-science>
- Larasati, F. A., Ratnawati, D. E., & Hanggara, B. T. (2022). *Analisis Sentimen Ulasan Aplikasi Dana dengan Metode Random Forest* (Vol. 6, Nomor 9). <http://j-ptiik.ub.ac.id>
- Locarso, G. K. (2022). ANALISIS SENTIMEN REVIEW APLIKASI PEDULILINDUNGI PADA GOOGLE PLAY STORE MENGGUNAKAN NBC. *Jurnal Teknik Informatika Kaputama (JTIK)*, 6(2).
- Miftahusalam, A., Pratiwi, H., & Slamet, I. (2023). Perbandingan Metode Random Forest dan Naive Bayes pada Analisis Sentimen Review Aplikasi BCA Mobile. *SIPTEK : Seminar Nasional Inovasi dan Pengembangan Teknologi Pendidikan*, 1(1). <https://proceeding.unesa.ac.id/index.php/siptek/article/view/184>
- Nurjannah, M., Fitri Astuti, I., & Program Studi, D. (2013). PENERAPAN ALGORITMA TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY (TF-IDF) UNTUK TEXT MINING Mahasiswa S1 Program Studi Ilmu Komputer FMIPA Universitas Mulawarman 2,3). Dalam *Jurnal Informatika Mulawarman* (Vol. 8, Nomor 3).
- Prakoso, I. B., Richasdy, D., & Purbolaksono, M. D. (2022). Sentiment Analysis of Telkom University as the Best BPU in Indonesia Using the Random Forest

- Method. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, 6(4), 2050. <https://doi.org/10.30865/mib.v6i4.4567>
- Putra, M. I. (2019). Sistem Rekomendasi Kelayakan Kredit Menggunakan Metode Random Forest pada BRI Kantor Cabang Pelaihari. *J. Tek. Inform. dan Sist. Inf*, 13(1), 61.
- Ridwansyah, T. (2022). KLIK: Kajian Ilmiah Informatika dan Komputer Implementasi Text Mining Terhadap Analisis Sentimen Masyarakat Dunia Di Twitter Terhadap Kota Medan Menggunakan K-Fold Cross Validation Dan Naïve Bayes Classifier. *Media Online*, 2(5), 178–185. <https://djournals.com/klik>
- Rusdi Rahman, M., & Febri Diansyah, A. (2024). Sentiment Analysis on the Shopee Application on Playstore Using the Random Forest Classification Method. *Inform : Jurnal Ilmiah Bidang Teknologi Informasi dan Komunikasi*, 9(1). <https://doi.org/10.25139/inform.v9i1.5465>
- Sari, F. V., & Wibowo, A. (2019). ANALISIS SENTIMEN PELANGGAN TOKO ONLINE JD.ID MENGGUNAKAN METODE NAÏVE BAYES CLASSIFIER BERBASIS KONVERSI IKON EMOSI. *Jurnal SIMETRIS*, 10(2).