

# Analisis Sentimen Ulasan Aplikasi RedBus Menggunakan Metode SVM dan AdaBoost

Shelomita Putrinda Culio<sup>a1</sup>, Luh Gede Astuti<sup>a2</sup>

<sup>a</sup>Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,  
Universitas Udayana  
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia  
<sup>1</sup>shelomitapc@gmail.com  
<sup>2</sup>lg.astuti@unud.ac.id (Corresponding Author)

## Abstract

*Bus transportation is a mode of transportation relied upon by the public due to its accessibility and affordable ticket prices. Sentiment analysis of RedBus app reviews on the Google Play Store can provide insight into user sentiment toward the app. The aims of this study are to analyze sentiment in reviews of the RedBus app using two approaches: the Random Forest model and a combination of Random Forest with AdaBoost. The analysis classifies user opinions as positive or negative. The study uses the TF-IDF method for feature extraction, and the evaluation methods include K-Fold Cross Validation and Confusion Matrix. The findings indicate that the combination of Random Forest with AdaBoost significantly enhances performance compared to the standard Random Forest model. Using a combination of Random Forest with AdaBoost results in an average accuracy of 89.0%, while the standard Random Forest model attains an average accuracy of 85.0%.*

**Keywords:** Sentiment Analysis, RedBus, Random Forest, AdaBoost, TF-IDF

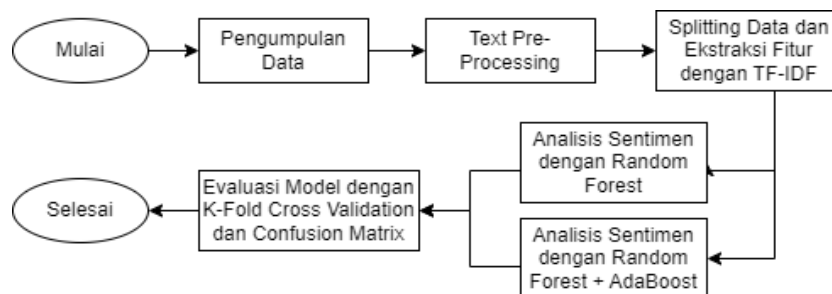
## 1. Pendahuluan

Transportasi publik memiliki peran yang penting dalam kehidupan masyarakat sehari-hari. Sebagai salah satu moda transportasi utama, bus cukup menjadi andalan masyarakat dalam berpergian, baik dalam kota maupun antarkota, khususnya karena alasan aksesibilitas yang luas dan harga tiket yang cukup terjangkau. Perkembangan era digital dan dorongan kebutuhan masyarakat terhadap layanan yang memberikan kemudahan dalam pemesanan tiket bus, menyebabkan mulai munculnya beberapa aplikasi pemesanan tiket bus dan *shuttle online*. Salah satu aplikasi pemesanan tiket bus *online* yang cukup populer di Indonesia adalah RedBus. Aplikasi RedBus menawarkan akses pada informasi lengkap mengenai jadwal, rute, harga tiket, dan pilihan pembayaran yang fleksibel. Pada layanan Google Play Store, aplikasi RedBus memiliki total unduhan yang cukup tinggi dengan lebih dari 50 juta unduhan dan 2 juta ulasan. Pandangan dan tingkat kepuasan pengguna dicurahkan dalam ulasan yang beragam. Ulasan pengguna Aplikasi RedBus pada Google Play Store tersebut dapat dianalisis dengan menggunakan teknik khusus, yaitu analisis sentimen, untuk mengklasifikasikan bagaimana sentimen masyarakat terhadap aplikasi. Analisis sentimen, juga dikenal sebagai *opinion mining*, merupakan studi komputasi yang menganalisis perilaku, pandangan, dan emosi yang terkandung dalam suatu teks [1]. Teknik ini bertujuan untuk mengkategorikan teks berdasarkan opini positif, negatif, atau netral. Pendekatan yang dapat digunakan untuk proses analisis sentimen antara lain *machine learning-based*, *lexicon-based*, dan *hybrid* [2]. Beberapa teknik atau metode *machine learning* yang cukup populer pada *text mining* adalah *Support Vector Machine*, *Random Forest*, *Decision Tree*, dan *Naive Bayes*. *Random Forest* merupakan metode dalam *machine learning* yang menyatukan banyak *decision tree* yang dilatih secara paralel untuk membuat suatu prediksi [3]. Permasalahan yang dapat muncul pada metode *machine learning* adalah memiliki tingkat akurasi yang relatif rendah [4]. Salah satu langkah yang dapat diambil untuk meningkatkan keakuratan hasil pengujian adalah menggunakan metode *ensemble*, khususnya algoritma *boosting*, yaitu metode yang menggabungkan beberapa model pembelajaran yang lebih lemah untuk menciptakan model yang lebih kuat [5]. Contoh dari algoritma *boosting* tersebut adalah

*AdaBoost (Adaptive Boost)*. Penggunaan *AdaBoost* untuk meningkatkan akurasi algoritma analisis sentimen sudah dibuktikan pada penelitian terdahulu, seperti pada penelitian mengenai perbandingan algoritma *Multinomial Naive Bayes* dan *Decision Tree* pada analisis sentimen aplikasi PeduliLindungi [6]. Pengaplikasian *AdaBoost* mampu meningkatkan rata-rata akurasi *Multinomial Naive Bayes* dari 83,7% menjadi 88,8% dan akurasi rata-rata *Decision Tree* dari 82,8% menjadi 84,1%. Selain itu, pada penelitian lain mengenai klasifikasi data *mining* pada dataset diabetes yang *imbalance* menggunakan algoritma *K-Nearest Neighbor* [7], *AdaBoost* mampu meningkatkan akurasi tertinggi dari 92,90% menjadi 95,40%. Oleh karena itu, penelitian ini akan menerapkan algoritma *AdaBoost* bersama dengan metode *machine learning* lainnya. Penelitian berupa analisis sentimen ulasan aplikasi RedBus menggunakan dua pendekatan, yaitu metode *Random Forest* dan kombinasi *Random Forest* dengan *AdaBoost*. Perbandingan antara kedua pendekatan ini akan memberikan gambaran yang jelas mengenai dampak penggunaan *AdaBoost* pada kinerja algoritma *Random Forest*, terutama pada akurasi analisis sentimen ulasan.

## 2. Metode Penelitian

### 2.1. Tahapan Penelitian



Gambar 1. Tahapan Penelitian

Penelitian diawali dengan mengumpulkan data teks berlabel yang berisi ulasan atau komentar pengguna aplikasi RedBus. Data teks yang telah dikumpulkan kemudian diolah melalui *text preprocessing* dalam beberapa tahap, seperti *cleaning text*, *case folding*, *stopword removal*, dan lainnya. Data tersebut kemudian dibagi menjadi data *training* dan data *testing*. Data diekstraksi fitur menggunakan teknik TF-IDF. Kemudian, dilakukan pembangunan dua model analisis sentimen, yaitu model *Random Forest* dan model kombinasi *Random Forest* dengan *AdaBoost*. *Grid Search* digunakan untuk melakukan pencarian parameter terbaik untuk kedua model. Tahap terakhir adalah melakukan evaluasi terhadap kedua model dengan dengan *K-Fold Cross Validation*, *Confusion Matrix*, serta perhitungan *accuracy*, *precision*, *recall*, dan *F-1 score* untuk setiap *fold*. Seluruh tahapan penelitian ini akan diimplementasikan menggunakan bahasa pemrograman *Python* dan platform *Google Colab*. Tahapan keseluruhan penelitian ditunjukkan pada Gambar 1.

### 2.2. Pengumpulan Data

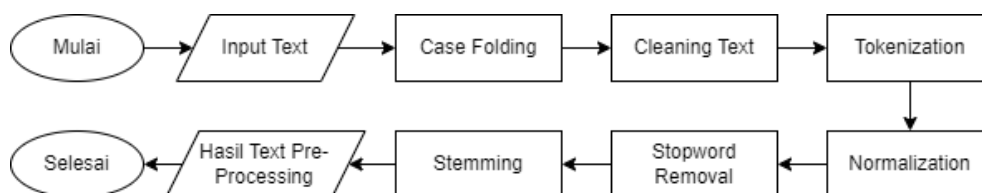
Data ulasan teks pengguna tentang aplikasi RedBus dalam penelitian ini merupakan data sekunder yang diperoleh dari platform Kaggle. Dataset tersebut berasal dari ulasan pengguna pada aplikasi RedBus di Google Play Store. Jumlah ulasan yang digunakan adalah sebanyak 500 ulasan yang terdiri dari 250 ulasan berlabel positif dan 250 ulasan berlabel negatif. Contoh bentuk dataset untuk penelitian ini terdapat pada Tabel 1.

**Tabel 1.** Contoh Bentuk Dataset

| no  | content   | label   |
|-----|---|---------|
| 348 | Bahaya aplikasi ini jauh dari realnya           | Negatif |
| 37  | Ga sesuai pemberangkatan, Gajelas               | Negatif |
| 355 | Sangat membantu dalam mencari tiket Bus Positif |         |

### 2.3. Text Preprocessing

*Text preprocessing* merupakan proses membersihkan dan memformat teks sehingga siap untuk digunakan atau dianalisis pada tahap selanjutnya [8]. Tahapan *text preprocessing* yang diaplikasikan dalam penelitian ini terlihat pada Gambar 2.



**Gambar 2.** Tahapan *Preprocessing*

*Text preprocessing* dimulai dengan *case folding*, yaitu teks diubah menjadi huruf kecil. Setelah itu, dilakukan *cleaning text* untuk menghilangkan karakter khusus, tanda baca, dan simbol-simbol yang tidak relevan. Teks kemudian dibagi menjadi token-token terpisah pada proses *tokenization*. Tahap selanjutnya adalah *normalization* untuk mengubah kata yang tidak sesuai menjadi ke dalam bentuk standar, seperti mengganti kata *slang* menjadi kata formal. *Stopword removal* merupakan proses menghilangkan kata-kata umum yang tidak memberikan informasi penting bagi analisis sentimen. Tahap terakhir adalah *stemming*, yaitu proses mengubah kata ke bentuk dasarnya dengan menghapus imbuhan. Teks yang telah melalui semua proses tersebut akan menjadi data yang lebih bersih dan terstruktur, sehingga siap untuk proses analisis sentimen.

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

TF-IDF merupakan sebuah pendekatan untuk membobotkan kata-kata dalam dokumen berdasarkan seberapa sering kata tersebut muncul dalam korpus teks. Konsep utama TF-IDF terdiri dari dua bagian, yaitu *Term Frequency* (TF) yang menilai frekuensi kemunculan sebuah kata dalam dokumen tertentu, dan *Inverse Document Frequency* (IDF) yang menilai kelangkaan kata tersebut di seluruh kumpulan dokumen (korpus) [9], seperti yang digambarkan pada persamaan (1).

$$TF - IDF(t, d, D) = \frac{\text{Jml muncul kata } t \text{ dalam dokumen } d}{\text{Jml total kata dalam dokumen } d} \times \log \left( \frac{\text{Total dokumen dalam } D}{\text{Jml dokumen yang mengandung } t} \right) \quad (1)$$

Dimana *t* adalah kata yang ingin dihitung frekuensi kemunculannya, *d* adalah dokumen tempat kata *t* muncul, dan *D* adalah jumlah total dokumen. Pada analisis sentimen, kata-kata penting dan unik dalam dokumen akan memiliki bobot yang besar, sedangkan kata-kata umum dan sering muncul akan memiliki bobot yang lebih kecil. Proses ekstraksi fitur menggunakan TF-IDF dapat membantu untuk fokus pada kata-kata yang relevan dan mengabaikan kata yang tidak relevan dalam analisis sentimen.

### 2.5. Random Forest

*Random Forest* merupakan metode *machine learning* yang menyatukan banyak *decision tree* untuk menciptakan prediksi yang lebih akurat. Algoritma ini melibatkan pengambilan sampel acak dengan penggantian (*bootstrap*) dari data *training* untuk membangun *decision tree* secara terpisah. Setiap *tree* dilatih menggunakan subset acak dari fitur yang tersedia pada setiap simpul.

Prediksi akhir dihitung sebagai rata-rata (untuk regresi) atau voting mayoritas (untuk klasifikasi) dari semua *tree* [3]. Metode ini menggunakan indeks Gini untuk menentukan pemisahan terbaik pada setiap simpul pohon keputusan, yaitu dengan mengukur tingkat ketidakseimbangan distribusi kelas pada data seperti pada persamaan (2). Pemisahan yang optimal adalah yang meminimalkan nilai indeks tersebut [10].

$$Gini = 1 - \sum_{i=1}^k (P_i)^2 \quad (2)$$

Keterangan:

$k$  = jumlah kelas,  $P_i$  = proporsi kelas  $i$  dalam data

Dalam penelitian ini, *Random Forest* digunakan untuk mengklasifikasikan teks berdasarkan sentimen positif dan negatif dengan melakukan pembelajaran pada data training berlabel, yaitu ulasan teks, dan membuat prediksi pada data baru. Beberapa parameter dapat disesuaikan untuk mengoptimalkan performa model algoritma ini, seperti  $n\_estimators$  yang merupakan jumlah *decision tree* yang akan dibangun,  $max\_depth$  yang merupakan kedalaman maksimum dari setiap *tree*, dan  $random\_state$  yang merupakan pengontrol proses pengacakan dalam pengambilan sampel.

## 2.6. AdaBoost (Adaptive Boost)

*AdaBoost (Adaptive Boost)* merupakan algoritma *ensemble* yang umum digunakan sebagai algoritma *boosting*. Algoritma ini meningkatkan kinerja model klasifikasi dengan cara menggabungkan beberapa model yang lemah (*weak classifier*) menjadi model yang lebih kuat (*strong classifier*). Konsep kerja algoritma ini melibatkan pembobotan data *training* berdasarkan kesalahan klasifikasi sebelumnya, yaitu data yang salah diklasifikasi akan mendapatkan bobot lebih tinggi sehingga pada pelatihan model baru selanjutnya ditekankan pada data yang sulit diklasifikasikan [11]. Proses ini akan dilakukan hingga jumlah iterasi tertentu, menyesuaikan dengan parameter  $n\_estimators$  yang digunakan. Dalam analisis sentimen, *AdaBoost* dapat bermanfaat untuk mengurangi risiko *overfitting* dan meningkatkan akurasi prediksi dengan mengatasi kesulitan dalam data teks yang kompleks atau tidak seimbang.

## 2.7. K-Fold Cross Validation

*K-Fold Cross Validation* merupakan prosedur pengambilan sampel ulang yang digunakan untuk melakukan evaluasi model *machine learning* menggunakan sampel data yang terbatas. Metode ini bekerja dengan membagi data menjadi  $k$  *fold*, tiap *fold* berfungsi sebagai data *training* dan *testing* secara bergantian. Maka, setiap data akan memiliki kesempatan untuk digunakan dalam set *testing* sekali dan dalam set *training*  $k-1$  kali. *K-Fold Cross Validation* berguna untuk mengurangi risiko bias dan memberikan evaluasi model yang lebih andal [12]. Nilai  $k$  yang sering digunakan adalah  $k=5$  dan  $k=10$ . Dalam penelitian ini, penggunaan *K-Fold Cross Validation* dilakukan untuk memperoleh hasil evaluasi analisis sentimen yang lebih objektif dan stabil.

## 2.8. Confusion Matrix

*Confusion Matrix* merupakan metode evaluasi yang diterapkan untuk menggambarkan performa suatu metode klasifikasi. Matriks ini memperlihatkan hasil prediksi model dibandingkan dengan kondisi aktual dari data yang diamati [13]. *Confusion Matrix* memiliki 4 elemen utama seperti pada Tabel 2.

**Tabel 2.** Confusion Matrix

|        |                     | Prediction         |                     |
|--------|---------------------|--------------------|---------------------|
|        |                     | True Positive (TP) | False Negative (FN) |
| Actual | True Positive (TP)  |                    |                     |
|        | False Positive (FP) |                    | True Negative (TN)  |

Berdasarkan 4 elemen tersebut, beberapa metrik evaluasi dapat dihitung, yaitu *accuracy* yang mengindikasikan sejauh mana data telah diklasifikasikan dengan tepat dari seluruh data yang digunakan seperti pada persamaan (3), *precision* yang mengukur proporsi prediksi positif yang benar seperti pada persamaan (4), *recall* yang mengukur kemampuan model untuk mendeteksi kasus positif seperti pada persamaan (5), dan *F-1 score* yang memperlihatkan gambaran kinerja yang seimbang antara *precision* dan *recall* seperti pada persamaan (6) [13].

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative} \times 100\% \quad (3)$$

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \times 100\% \quad (4)$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \times 100\% \quad (5)$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \times 100\% \quad (6)$$

### 3. Hasil dan Diskusi

#### 3.1. Text Preprocessing

Tahap *text preprocessing* yang diterapkan pada 500 data ulasan (250 label positif dan 250 label negatif) dilakukan menggunakan bahasa pemrograman *Python*. Dataset dinormalisasi pada tahap *normalization* menggunakan file *colloquial-indonesian-lexicon.csv* yang diperoleh dari GitHub untuk menyelaraskan format dan konten data. Proses *stopword removal* dilakukan dengan menggunakan *library NLTK* dan proses *stemming* menggunakan *library Sastrawi*. Selanjutnya dilakukan perhitungan persentase tanda baca (*punct*) dan perhitungan panjang teks (*review\_len*) untuk setiap data ulasan. Hasil dari tahap *text preprocessing* tersaji dalam Tabel 3.

**Tabel 3.** Hasil *Text Preprocessing*

| Tahap Pre-Processing Content |   |
|------------------------------|---|
| Initial data                 | Mantap, banyak potongan harga tiket, banyak pilihan buss                        |
| Case Folding                 | mantap, banyak potongan harga tiket, banyak pilihan buss                        |
| Cleaning Text                | mantap banyak potongan harga tiket banyak pilihan buss                          |
| Tokenization                 | ['mantap', 'banyak', 'potongan', 'harga', 'tiket', 'banyak', 'pilihan', 'buss'] |
| Normalization                | ['mantap', 'banyak', 'potongan', 'harga', 'tiket', 'banyak', 'pilihan', 'bus']  |
| Stopword Removal             | ['mantap', 'potongan', 'harga', 'tiket', 'pilihan', 'bus']                      |
| Stemming                     | ['mantap', 'potong', 'harga', 'tiket', 'pilih', 'bus']                          |

Jumlah data yang dihasilkan setelah melalui *text preprocessing* adalah tetap sejumlah 500 ulasan. Selanjutnya, label data diubah dari bentuk teks menjadi data nominal dengan proses *mapping*, yaitu label negatif diubah menjadi 0 dan label positif diubah menjadi 1. Hasil perubahan bentuk label ulasan terlihat pada Tabel 4.

**Tabel 4.** Tabel Hasil *Label Mapping*

| content   | label |
|---|-------|
| ['aplikasi', 'bantu', 'cari', 'beli', 'tiket', 'bus']         | 1     |
| ['aplikasi', 'sampah', 'jam', 'berangkat', 'tidak', 'sesuai'] | 0     |
| ['bayar', 'loading', 'ulang', 'gagal']                        | 0     |

### 3.2. Ekstraksi Fitur

Setelah melalui tahap *text preprocessing*, data di *split* untuk dialokasikan sebagai data *training* sebanyak 80% dan data *testing* sebanyak 20%. Data ulasan berlabel positif terbagi menjadi 200 data *training* dan 50 data *testing*, begitu pula dengan data ulasan berlabel negatif terbagi menjadi 200 data *training* dan 50 data *testing*.

Setelah itu, dilanjutkan ekstraksi fitur terhadap data ulasan menggunakan TF-IDF. Fitur yang diekstraksi menggunakan TF-IDF akan digunakan sebagai masukan untuk model analisis sentimen. Tahap ekstraksi fitur ini menggunakan bahasa pemrograman *Python* dengan *library* TF-IDF dari *Scikit-Learn* berupa modul *TfidfVectorizer*. Total jumlah fitur yang telah diekstrak adalah sebanyak 400 fitur dari total 200 dokumen. Hasil tahap ekstraksi fitur diperlihatkan pada Gambar 3 dan 4.

```

Hasil kata ekstraksi fitur dengan TF-IDF:
'ada', 'admin', 'adminnya', 'adu', 'agen', 'aju', 'akibat', 'akses', 'aktif'
'akun', 'akurat', 'alam', 'alas', 'alhamdulillah', 'aman', 'an', 'anak', 'apk'
'aplikasi', 'app', 'april', 'arah', 'armada', 'asih', 'asli', 'baca', 'bagus'
'baik', 'balas', 'banding', 'bandung', 'banget', 'bantu', 'banyak', 'batal', 'batas'
'bayar', 'bebas', 'beda', 'bekas', 'belah', 'beli', 'berangkat', 'bersih', 'besok'
'biar', 'bikin', 'bilang', 'bintang', 'biru', 'bis', 'bocor', 'bohong', 'booking'
'bookingsnya', 'bukti', 'buruk', 'bus', 'busuk', 'butuh', 'cacat', 'call', 'cancel'
'cari', 'center', 'cepat', 'cetak', 'cikampek', 'cirebon', 'class', 'coba', 'cocok'
'contoh', 'crew', 'cs', 'cuman', 'customer', 'daftar', 'dah', 'dana', 'dasar'
'deh', 'depan', 'dgan', 'dibalikin', 'dicancel', 'dikabarin', 'diskon', 'dll', 'doang'
'download', 'duduk', 'duit', 'eh', 'email', 'enak', 'enggak', 'eror', 'error'
'evaluasi', 'executive', 'fasilitas', 'fast', 'feedback', 'friendly', 'gagal', 'gampang',
'gara', 'gila', 'gilir', 'good', 'gue', 'guna', 'habis', 'handoyo', 'hangus'
'hapus', 'harap', 'harga', 'hasil', 'hati', 'hilang', 'hubung', 'iklan', 'indomaret'
'indonesia', 'info', 'informasi', 'install', 'jadi', 'jadwal', 'jakarta', 'jalan', 'jalur'
'jam', 'jamin', 'jangkau', 'jaya', 'jelas', 'jelek', 'jemput', 'jual', 'jurus'
'kabar', 'kacau', 'kadaluarsa', 'kadang', 'kait', 'kali', 'kalideres', 'kalo', 'kampung'
'kantor', 'kapok', 'karawang', 'kasih', 'kayak', 'kecewa', 'keluh', 'kemarin', 'kembali'
'kendala', 'keren', 'kerja', 'kesini', 'ketik', 'kirim', 'klik', 'kode', 'kolom'
    
```

**Gambar 3.** Hasil Ekstraksi Fitur dengan TF-IDF

|     |            |          |      |          |          |        |               |      |        |       |      |   |
|-----|------------|----------|------|----------|----------|--------|---------------|------|--------|-------|------|---|
|     | Review_len | punct    | ada  | admin    | adminnya | adu    | agen          | aju  | akibat | \     |      |   |
| 374 | 113        | 2.7      | 0.0  | 0.000000 | 0.0      | 0.0    | 0.0           | 0.0  | 0.0    | 0.0   |      |   |
| 302 | 32         | 0.0      | 0.0  | 0.000000 | 0.0      | 0.0    | 0.0           | 0.0  | 0.0    | 0.0   |      |   |
| 362 | 155        | 0.6      | 0.0  | 0.304804 | 0.0      | 0.0    | 0.0           | 0.0  | 0.0    | 0.0   |      |   |
| 106 | 201        | 1.5      | 0.0  | 0.000000 | 0.0      | 0.0    | 0.0           | 0.0  | 0.0    | 0.0   |      |   |
| 364 | 75         | 1.3      | 0.0  | 0.000000 | 0.0      | 0.0    | 0.0           | 0.0  | 0.0    | 0.0   |      |   |
|     | akses      | aktif    | akun | akurat   | alam     | alas   | alhamdulillah | aman | an     | anak  | \    |   |
| 374 | 0.0        | 0.0      | 0.0  | 0.0      | 0.0      | 0.0    | 0.0           | 0.0  | 0.0    | 0.0   |      |   |
| 302 | 0.0        | 0.0      | 0.0  | 0.0      | 0.0      | 0.0    | 0.0           | 0.0  | 0.0    | 0.0   |      |   |
| 362 | 0.0        | 0.0      | 0.0  | 0.0      | 0.0      | 0.0    | 0.0           | 0.0  | 0.0    | 0.0   |      |   |
| 106 | 0.0        | 0.0      | 0.0  | 0.0      | 0.0      | 0.0    | 0.0           | 0.0  | 0.0    | 0.0   |      |   |
| 364 | 0.0        | 0.0      | 0.0  | 0.0      | 0.0      | 0.0    | 0.0           | 0.0  | 0.0    | 0.0   |      |   |
|     | apk        | aplikasi | app  | april    | arah     | armada | asih          | asli | baca   | bagus | baik | \ |
| 374 | 0.0        | 0.264281 | 0.0  | 0.0      | 0.0      | 0.0    | 0.0           | 0.0  | 0.0    | 0.0   | 0.0  |   |
| 302 | 0.0        | 0.337140 | 0.0  | 0.0      | 0.0      | 0.0    | 0.0           | 0.0  | 0.0    | 0.0   | 0.0  |   |
| 362 | 0.0        | 0.114752 | 0.0  | 0.0      | 0.0      | 0.0    | 0.0           | 0.0  | 0.0    | 0.0   | 0.0  |   |
| 106 | 0.0        | 0.358108 | 0.0  | 0.0      | 0.0      | 0.0    | 0.0           | 0.0  | 0.0    | 0.0   | 0.0  |   |
| 364 | 0.0        | 0.000000 | 0.0  | 0.0      | 0.0      | 0.0    | 0.0           | 0.0  | 0.0    | 0.0   | 0.0  |   |

**Gambar 4.** Contoh Hasil Vektorisasi Fitur Beberapa Data

### 3.3. Pemodelan Analisis Sentimen *Random Forest* dan *AdaBoost*

Pada tahap analisis sentimen, dilakukan pembuatan 2 model, yaitu model *Random Forest* dan model *Random Forest* dengan penggunaan *AdaBoost*. Model *Random Forest* dibangun menggunakan bahasa pemrograman *Python* dengan menggunakan modul *RandomForestClassifier* dari *library* Scikit-learn. Evaluasi *training* model dilakukan dengan teknik *K-Fold Cross Validation* dan menggunakan jumlah *fold* k=5 untuk memastikan kinerja model konsisten. Pada model ini, *Grid search* dengan modul *GridSearchCV* digunakan untuk mendapatkan parameter terbaik yang akan digunakan. Parameter terbaik yang digunakan dalam model *Random Forest* meliputi jumlah *tree* (*n\_estimators*=100), kedalaman maksimum *tree* (*max\_depth*=None), dan *random\_state*=0. Hasil evaluasi 80% data *training* untuk setiap *fold* dari model *Random Forest* ditampilkan pada Tabel 5. Akurasi terbaik yang dihasilkan pada model *Random Forest* adalah sebesar 90,0% dengan hasil akurasi rata-rata sebesar 87,0%, sedangkan nilai rata-rata *precision* sebesar 87,1%, *recall* sebesar 87,5%, dan *F1-score* 87,0%.

**Tabel 5.** Hasil Evaluasi *K-Fold Cross Validation* Model *Random Forest*

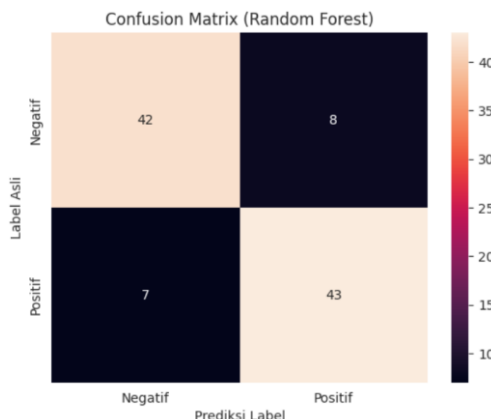
| <b>N-Fold</b>  | <b>Accuracy</b> | <b>Precision</b> | <b>Recall</b> | <b>F1-Score</b> |
|----------------|-----------------|------------------|---------------|-----------------|
| 1              | 0.8875          | 0.897436         | 0.875         | 0.886076        |
| 2              | 0.8750          | 0.857143         | 0.900         | 0.878049        |
| 3              | 0.8500          | 0.850000         | 0.850         | 0.850000        |
| 4              | 0.8375          | 0.846154         | 0.825         | 0.835443        |
| 5              | 0.9000          | 0.880952         | 0.925         | 0.902439        |
| <b>Average</b> | <b>0.870</b>    | <b>0.871</b>     | <b>0.875</b>  | <b>0.870</b>    |

Sementara itu, hasil *testing* model *Random Forest* dengan menggunakan parameter terbaik tersebut pada 20% data *testing* mendapatkan hasil seperti pada tabel 6. Akurasi rata-rata yang didapatkan adalah sebesar 85%, dengan nilai rata-rata *precision* sebesar 85%, *recall* sebesar 85%, dan *F1-score* 85%.

**Tabel 6.** Hasil *Testing* Model *Random Forest*

|                | <b>Accuracy</b> | <b>Precision</b> | <b>Recall</b> | <b>F1-Score</b> |
|----------------|-----------------|------------------|---------------|-----------------|
| Negatif        | 0.85            | 0.86             | 0.84          | 0.85            |
| Positif        | 0.85            | 0.84             | 0.86          | 0.85            |
| <b>Average</b> | <b>0.85</b>     | <b>0.85</b>      | <b>0.85</b>   | <b>0.85</b>     |

Kemudian, pada hasil evaluasi menggunakan *Confusion Matrix* didapatkan hasil bahwa model berhasil memprediksi dengan benar 43 ulasan positif dan salah menebak 7 ulasan positif menjadi negatif. Kemudian model berhasil memprediksi dengan benar 42 ulasan negatif dan salah menebak 8 ulasan negatif menjadi positif. Hasil *Confusion Matrix* ditampilkan pada Gambar 5.



**Gambar 5.** Hasil Evaluasi Confusion Matrix Model Random Forest

Pada model kedua, yaitu kombinasi *Random Forest* dengan algoritma *AdaBoost*, menggunakan modul *AdaBoostClassifier* dan *RandomForestClassifier* dari *library* Scikit-learn. Evaluasi model dilakukan dengan teknik *K-Fold Cross Validation* dengan jumlah *fold* yang digunakan adalah  $k=5$ . Pada model ini, *Grid search* dengan modul *GridSearchCV* digunakan untuk mendapatkan parameter terbaik yang akan digunakan. Model menggunakan *base estimator Random Forest* dengan parameter jumlah estimator ( $n\_estimator=50$ ), kedalaman maksimum *tree* ( $max\_depth=5$ ), dan *random state=0*. Kemudian, model *AdaBoost* dibangun dengan *base estimator Random Forest*, parameter tingkat pembelajaran ( $learning\_rate=0.1$ ), dan jumlah estimator ( $n\_estimator=20$ ). Hasil evaluasi 80% data *training* untuk setiap *fold* dari model *Random Forest* dengan *AdaBoost* ditampilkan pada Tabel 7. Akurasi terbaik yang dihasilkan pada model *Random Forest* dengan *AdaBoost* adalah sebesar 93,75% dengan hasil akurasi rata-rata sebesar 90,75%, sedangkan nilai rata-rata *precision* sebesar 91,22%, *recall* sebesar 90,0%, dan *F1-score* 90,64%.

**Tabel 7.** Hasil Evaluasi *K-Fold Cross Validation* Model *Random Forest* dengan *AdaBoost*

| N-Fold         | Accuracy      | Precision     | Recall      | F1-Score      |
|----------------|---------------|---------------|-------------|---------------|
| 1              | 0.9250        | 0.972222      | 0.875       | 0.921053      |
| 2              | 0.9250        | 0.947368      | 0.900       | 0.923077      |
| 3              | 0.9000        | 0.900000      | 0.900       | 0.900000      |
| 4              | 0.8500        | 0.850000      | 0.850       | 0.850000      |
| 5              | 0.9375        | 0.926829      | 0.950       | 0.938272      |
| <b>Average</b> | <b>0.9075</b> | <b>0.9122</b> | <b>0.90</b> | <b>0.9064</b> |

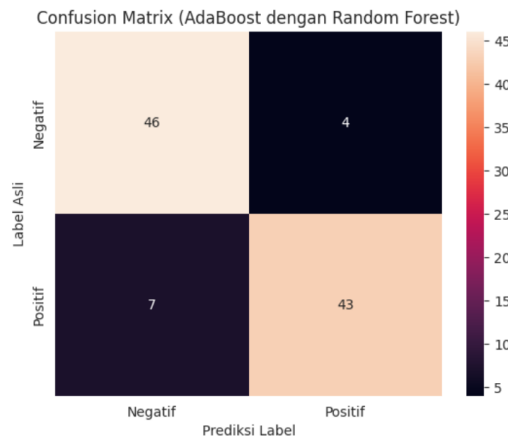
Sementara itu, hasil *testing* model *Random Forest* dengan *AdaBoost* menggunakan parameter terbaik tersebut pada 20% data *testing* mendapatkan hasil seperti pada tabel 8. Akurasi rata-rata yang didapatkan adalah sebesar 89%, dengan nilai rata-rata *precision* sebesar 89%, *recall* sebesar 89%, dan *F1-score* 89%.

**Tabel 8.** Hasil *Testing* Model *Random Forest*

|                | Accuracy    | Precision   | Recall      | F1-Score    |
|----------------|-------------|-------------|-------------|-------------|
| Negatif        | 0.89        | 0.87        | 0.92        | 0.89        |
| Positif        | 0.89        | 0.91        | 0.86        | 0.89        |
| <b>Average</b> | <b>0.89</b> | <b>0.89</b> | <b>0.89</b> | <b>0.89</b> |

Selanjutnya, pada hasil evaluasi menggunakan *Confusion Matrix* didapatkan hasil bahwa model berhasil memprediksi dengan benar 43 ulasan positif dan salah menebak 7 ulasan positif menjadi

negatif. Kemudian model berhasil memprediksi dengan benar 46 ulasan negatif dan salah menebak 4 ulasan negatif menjadi positif. Hasil *Confusion Matrix* ditampilkan pada Gambar 6.



**Gambar 6.** Hasil Evaluasi *Confusion Matrix* Model *Random Forest* dengan *AdaBoost*

Berdasarkan hasil evaluasi yang didapatkan dari kedua model tersebut, didapatkan perbandingan nilai rata-rata *accuracy*, *recall*, *precision*, dan *F1-score* antara kedua model yang ditampilkan pada Tabel 9. Data pada tabel tersebut menunjukkan bahwa model kombinasi *Random Forest* dengan *AdaBoost* menghasilkan nilai rata-rata *accuracy*, *recall*, *precision*, dan *F1-score* yang lebih tinggi dibandingkan model *Random Forest* biasa.

**Tabel 9.** Perbandingan Kinerja Rata-Rata Model *Random Forest* dan *Random Forest + AdaBoost*

|         | Accuracy | Precision | Recall | F1-Score |
|---------|----------|-----------|--------|----------|
| RF      | 0.85     | 0.85      | 0.85   | 0.85     |
| RF + AB | 0.89     | 0.89      | 0.89   | 0.89     |

#### 4. Kesimpulan

Berdasarkan temuan dari penelitian yang telah dilakukan untuk membandingkan kinerja model *Random Forest* dan model kombinasi *Random Forest* dengan *AdaBoost* dalam analisis sentimen ulasan aplikasi RedBus, dapat disimpulkan bahwa pendekatan kombinasi *Random Forest* dengan *AdaBoost* memberikan peningkatan kinerja yang cukup signifikan. Model *Random Forest* dengan *AdaBoost* berhasil mencapai akurasi rata-rata sebesar 89,0%, sedangkan akurasi rata-rata model *Random Forest* biasa adalah 85,0%. Model kombinasi tersebut juga menunjukkan peningkatan kinerja pada nilai *precision* yang mencapai 89,0%, *recall* 89,0%, dan *F1-score* 89,0%. Saran untuk penelitian selanjutnya dapat mengeksplorasi model kombinasi lain, seperti menggunakan algoritma *boosting* yang berbeda atau model *ensemble* lain. Selain itu, juga dapat mempertimbangkan penggunaan metode *deep learning* untuk melakukan perbandingan kinerja dengan model yang diterapkan dalam penelitian ini.

#### Daftar Pustaka

- [1] B. Saberi and S. Saad, "Sentiment Analysis or Opinion Mining: A Review," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 7, no. 5, pp. 1660-1666, 2017.
- [2] S. Thomas, Yuliana and N. P, "Studi Analisis Metode Analisis Sentimen pada YouTube," *JIFOTECH (Journal of Information Technology)*, vol. 1, no. 1, pp. 1-7., 2021.
- [3] S. Rigatti, "Random Forest," *Journal of Insurance Medicine*, vol. 47, no. 1, pp. 31-39, 2017.

- [4] E. K. Adhitya, R. Satria and Subagyo, "Komparasi Metode Machine Learning dan Non-Machine Learning untuk Estimasi Usaha Perangkat Lunak.," *Journal of Software Engineering*, vol. 1, no. 2, pp. 109-113, 2015.
- [5] M. B. Hamzah, "Classification of Movie Review Sentiment Analysis Using Chi-Square and Multinomial Naïve Bayes with Adaptive Boosting," *Journal of Advances in Information Systems and Technology*, vol. 3, no. 1, pp. 67-74, 2021.
- [6] C. M. S. Ramdani, A. N. Rachman and R. Setiawan, "Comparison of the Multinomial Naive Bayes Algorithm and Decision Tree with the Application of AdaBoost in Sentiment Analysis Reviews PeduliLindungi Application," *IJISTECH (International Journal of Information System and Technology)*, vol. 6, no. 4, pp. 419-430, 2022.
- [7] N. Novianti, M. Zarlis and P. Sihombing, "Penerapan Algoritma Adaboost Untuk Peningkatan Kinerja Klasifikasi Data Mining Pada Imbalance Dataset Diabetes," *Jurnal Media Informatika Budidarma*, vol. 6, no. 2, pp. 1200-1206, 2022.
- [8] A. I. Kadhim, "An Evaluation of Preprocessing Techniques for Text Classification," *International Journal of Computer Science and Information Security (IJCSIS)*, vol. 16, no. 6, pp. 22-32, 2018.
- [9] S. Kalra, L. Li and H. R. Tizhoosh, "Automatic Classification of Pathology Reports using TF-IDF Features," *arXiv preprint arXiv:1903.07406.*, pp. 1-9, 5 March 2019.
- [10] B. Warsito and A. Prahutama, "Sentiment Analysis on Tokopedia Product Online Reviews using Random Forest Method," in *the 5th International Conference on Energy, Environmental and Information System (ICENIS 2020)*, Semarang, 2020.
- [11] S. I. Gultom, "Implementasi Data Mining Menentukan Pola Hidup Sehat Bagi Pengguna KB Menggunakan Algoritma Adaboost (Studi Kasus: Dinas Serdang Bedagai)," *Informasi dan Teknologi Ilmiah (INTI)*, vol. 7, no. 3, pp. 298-304, 2020.
- [12] J. Brownlee, "A Gentle Introduction to k-fold Cross-Validation," *Machine Learning Mastery*, 4 Oktober 2023. [Online]. Available: <https://machinelearningmastery.com/k-fold-cross-validation/>. [Accessed 5 May 2024].
- [13] R. Arthana, "Mengenal Accuracy, Precision, Recall dan Specificity serta yang Diprioritaskan dalam Machine Learning," *Medium*, 5 April 2019. [Online]. Available: <https://rey1024.medium.com/mengenal-accuracy-precision-recall-dan-specificity-serta-yang-diprioritaskan-b79ff4d77de8>. [Accessed 5 May 2024].