JNATIA Volume 1, Nomor 1, November 2022
Jurnal Nasional Teknologi Informasi dan Aplikasinya

Analisis Sentimen Ulasan Pengguna Aplikasi Pelayanan Masyarakat Dengan Menggunakan Algoritma *Random Forest*

I Nyoman Arlan Kusuma Ardika[a1], I Gede Arta Wibawa[a2]

[a]Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Udayana
Badung, Bali, Indonesia
[1]landarlan60@gmail.com
[2]gede.arta@unud.ac.id

Abstract

Public services by the government generally have an impact that is quickly responded to by the community. One form of public response is through their opinions through writings written on social media or reviews of applications developed by the government. Machine learning has been widely used for automatic opinion mining to classify sentiment classes. The classification method that can be used to classify public opinion into positive or negative sentiment classes is random forest. Based on the test results of the random forest algorithm in classifying sentiments from user reviews of public service applications by the government, the highest accuracy value was obtained at 84% by performing hyperparameter tuning.

Keywords: *Random Forest, hyperparameter tuning, TF-IDF, analisis sentimen, klasifikasi teks, Natural Language Processing*

1. Pendahuluan

Pelayanan publik merupakan suatu bentuk pelayanan yang diberikan oleh pemerintah kepada masyarakat umum. Umumnya, pelayanan tersebut akan menghasilkan suatu umpan balik kepada pemerintah dalam bentuk opini masyarakat yang dipublikasikan melalui *social media*. Adapun bentuk opini yang disampaikan dapat berupa *tweets* pada Twitter ataupun ulasan aplikasi pada *app store*. Adapun opini tersebut disampaikan agar dapat menjadi bahan evaluasi dalam proses pengambilan keputusan agar dapat meningkatkan pelayanan publik di masa mendatang.

Saat ini, perkembangan teknologi informasi memberikan banyak kontribusi dalam mengolah data untuk memberikan informasi. Salah satu penerapan teknologi informasi tersebut adalah analisis sentimen terhadap data opini. Dengan melakukan analisis sentimen, mesin dapat memberikan hasil berupa sentimen seseorang dalam bentuk data teks yang akan mengelompokan polaritas opini untuk mengetahui kelas sentimen dari opini tersebut termasuk dalam kelas positif atau kelas negatif[1].

Saat ini, ulasan aplikasi digunakan sebagai media untuk pengguna aplikasi agar dapat menyampaikan opini ataupun saran kepada para pengembang mengenai pelayanan yang diberikan. Ulasan-ulasan tersebut dapat digunakan sebagai data bagi para peneliti untuk dapat dianalisa sentimen dari ulasan tersebut. Dengan bantuan skor yang umumnya terdapat dalam *platform* pengunduhan aplikasi Android seperti *Google Play Store*, ulasan-ulasan tersebut dapat dikelompokan dalam kelas sentimen positif ataupun kelas sentimen negatif.

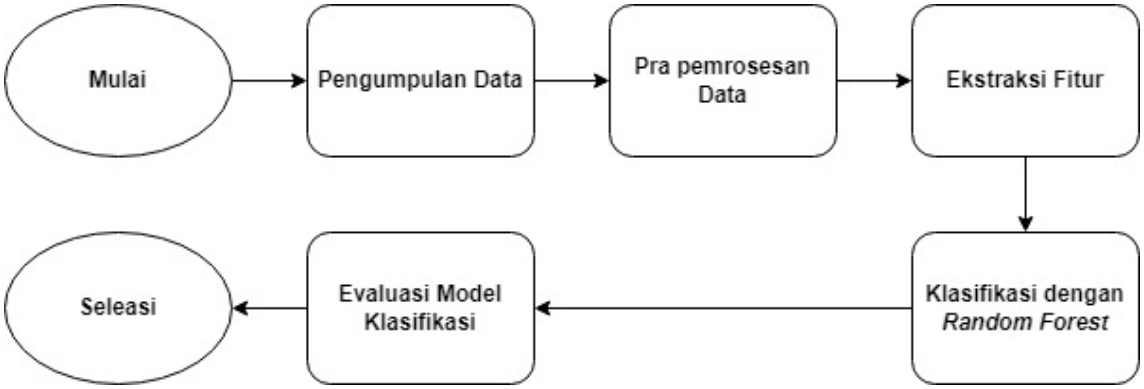
Adapun beberapa penelitian terkait analisis sentimen mengenai pelayanan masyarakat oleh pemerintah antara lain: penelitian yang dilakukan oleh Rosdiana, dkk yang melakukan analisis sentimen terhadap pelayanan pemerintah kota Makassar menggunakan metode *Naïve Bayes* pada tahun 2019 dan mendapatkan akurasi sebesar 91,6%[2]; penelitian yang dilakukan oleh Zamazami, dkk yang melakukan analisis sentimen terhadap ulasan film menggunakan metode

Modified Balanced Random Forest pada tahun 2021 dan mendapatkan nilai akurasi sebesar 79%[3]; serta penelitian yang dilakukan oleh Ailiyya pada tahun 2020 mengenai analisis sentimen berbasis aspek terhadap ulasan pengguna aplikasi Tokopedia dengan metode *Support Vector Machine* dan mendapatkan nilai akurasi sebesar 69,6%[4].

Berdasarkan pemaparan serta penelitian terkait sebelumnya, maka penulis tertarik untuk melakukan penelitian terhadap ulasan pengguna aplikasi pelayanan masyarakat oleh pemerintah dengan menggunakan metode *Random Forest*. Adapun data penelitian ini didapatkan dengan melakukan *scraping* pada situs Google Play Store dan dilabeli sentimennya berdasarkan skor pada ulasan.

2. Metodologi Penelitian

2.1. Alur Penelitian



Gambar 1. Alur Penelitian

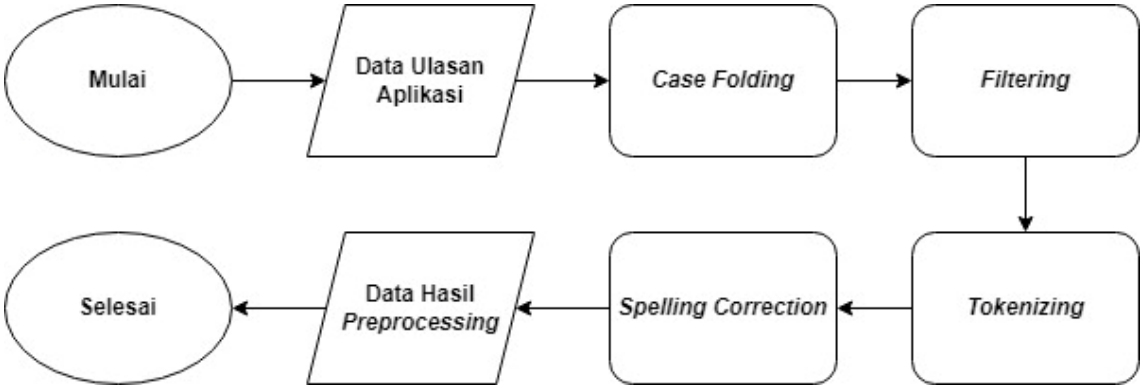
Penelitian diawali dengan mengumpulkan data penelitian yakni data ulasan pengguna aplikasi pelayanan masyarakat yang diberi label atau kelas sentimen positif atau negatif. Kemudian, akan dilakukan pra pemrosesan terhadap data tersebut untuk menormalisasikan data-data ang telah diperoleh. Berikutnya, data yang telah ternormalisasi akan dilakukan ekstraksi fitur dan menggunakan fitur tersebut untuk melakukan pembuatan model klasifiaksi dengan algoritma *Random Forest.* Setelah model diperoleh, model tersebut akan dilakukan evaluasi untuk mengetahui tingkat kekonsistenan hasil yang diberikan oleh model.

2.2. Pengumpulan Data

Data yang digunakan untuk mendukung penelitian ini adalah data sekunder, yakni ulasan pengguna aplikasi pelayanan masyarakat yang didapatkan melalui metode *web crawling* pada situs *Google Play Store*. Adapun aplikasi yang diperoleh data ulasannya meliputi: JAKI – Jakarta Kini, Pikobar, dan Sapawarga. Dataset ulasan diperoleh pada 26 September 2022 pukul 16:12 WITA yang memiliki total 1.356 data yang tersebar dalam 5 kelas skor ulasan. Label sentimen ulasan positif ataupun negatif diperoleh dengan mengelompokan 5 kelas skor menjadi 2 kelas sentimen, yang mana untuk skor 1-3 akan dikelompokan menjadi kelas sentimen negatif, dan skor 4-5 akan dikelompokan menjadi kelas sentimen positif.

2.3. Pra-pemrosesan Data

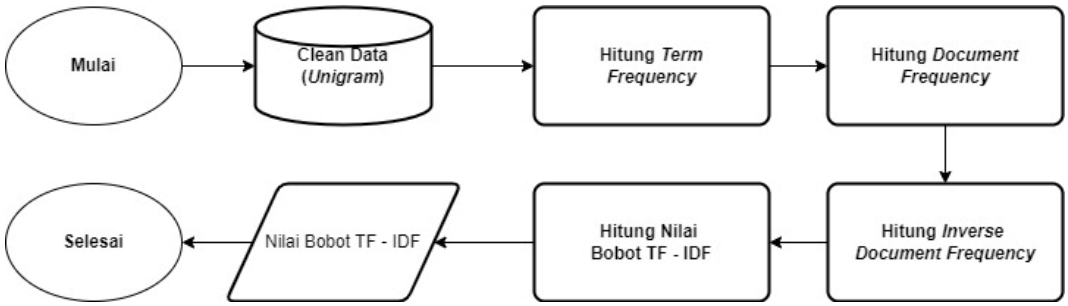
Data yang telah dikumpulkan dan dilabeli sentimennya akan memasuki tahap *preprocessing* untuk membersihkan data sebelum digunakan dalam tahap ekstraksi fitur[5]. Adapun tahapan *text preprocessing* yang dilakukan antara lain: *case folding, filtering, tokenizing,* dan *spelling correction* yang dapat dilihat pada Gambar 2:



Gambar 2. Alur Tahap Pra-pemrosesan Data

Data ulasan pengguna aplikasi akan memasuki tahap pra pemrosesan dimana dilakukan *case folding* terlebih dahulu dengan mengubah setiap karakter dalam ulasan menjadi huruf *lowercase.* Kemudian, dilakukan proses *filtering* yang akan menghapus *special characters* seperti tanda baca dan menghapus *stopwords* bahasa Indonesia. Berikutnya, data akan memasuki proses *tokenizing* yang akan memecah data yang awalnya dalam bentuk kalimat menjadi token-token atau satuan kata. Dan pada tahap terakhir pra pemrosesan, setiap token akan dilakukan pengecekan terhadap ejaannya agar mematuhi kaidah penulisan bahasa Indonesia yang baku. Adapun *tools* ataupun sumber yang digunakan untuk mempermudah proses *preprocessing* data dalam penelitian ini antara lain portal Kaggle milik Gilbert[6], repository GitHub milik Adeputri[7] dan artikel Medium milik Rohman[8].

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



Gambar 3. Alur Proses Perhitungan TF – IDF

Pada tahap ekstraksi fitur, data yang telah dilakukan pra pemrosesan akan dilakukan pembobotan berdasarkan hasil *tokenization* sebelumnya. Pembobotan ini dilakukan dengan melakukan perhitungan nilai TF – IDF berdasarkan persamaan (1), (2), dan (3).

$$TF - IDF = TF \ x \ IDF \tag{1}$$

$$TF = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \tag{2}$$

$$IDF = log \frac{N}{n_t} \tag{3}$$

Keterangan:
$f_{t,d}$ = jumlah kemunculan kata t pada dokumen d
$\sum_{t' \in d} f_{t',d}$ = jumlah kata pada dokumen d
N = jumlah dokumen
n_t = jumlah dokumen dengan kata t
Nilai bobot tersebut akan digunakan untuk mengukur seberapa besar kesesuaian suatu kata terhadap sekelompok dokumen[5]. Dengan demikian, akan diketahui apakah kata tersebut dapat digunakan sebagai fitur pada tahap seleksi fitur. Tidak hanya itu, pada tahap ini juga dilakukan

vektorisasi yang akan mengubah data teks menjadi vektor fitur yang diperlukan dalam algoritma klasifikasi nanti[9].

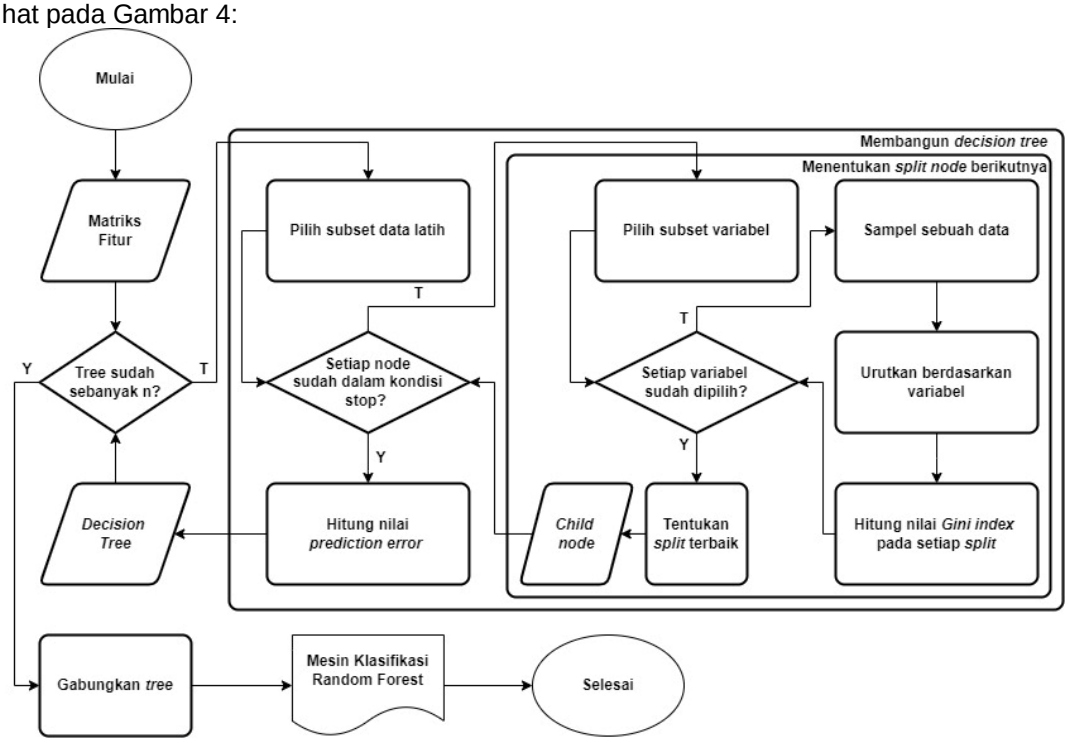
2.5. Pemodelan Klasifikasi Sentimen dengan *Random Forest*

Random Forest merupakan algoritma *supervised learning* yang digunakan dalam proses klasifikasi dan regresi[10]. Dalam penelitian ini, algoritma *random forest* akan digunakan untuk menyelesaikan permasalahan klasifikasi sentimen. Klasifikasi sentimen menggunakan metode *Random Forest* dilakukan dengan membangun model *ensemble tree*[11]. Setiap *decision tree* yang dibangun dalam *Random Forest* dibangun menggunakan *subset* data latih yang berbeda, yang mana untuk setiap *split* dari *node* digunakan perhitungan *Gini Index* dengan nilai terkecil dari setiap kelas untuk mendapatkan nilai *gain* terbesar[12]. Adapun *gini index* dapat dihitung seperti pada persamaan (4)

$$Gini\ (T) = 1 - \sum_{i=1}^{n} (p_i)^2 \tag{4}$$

Keterangan:
n = jumlah kemunculan kata t pada dokumen d
p_i = jumlah kata t pada dokumen d

Dalam membangun *tree*, dataset akan dipecah menjadi dataset latih, dataset validasi, dan dataset uji. Matriks fitur yang telah didapatkan melalui tahap ekstraksi fitur akan memasuki algoritma *Random Forest* untuk menghasilkan mesin klasifikasi yang digunakan untuk menganalisis ulasan untuk didapatkan sentimennya[13]. Adapun gambaran mengenai bagaimana setiap *decision tree* dibangun dan menentukan setiap *node* dalam *decision tree* dapat dilihat pada Gambar 4:



Gambar 4. Alur Pembuatan Model Klasifiaksi *Random Forest*

Kemudian, setelah diperoleh model klasifikasi sentimen, langkah berikutnya adalah melakukan pengujian dengan melakukan klasifikasi sentimen dari ulasan yang terdapat pada data validasi dan data uji.

2.6. Evaluasi Mesin

Evaluasi mesin dilakukan dengan melakukan perhitungan terhadap nilai akurasi dari model *machine learning* dalam melakukan klasifikasi sentimen. Evaluasi mesin klasifikasi *Random*

Forest, dapat dilakukan dengan melakukan pengujian nilai akurasi[14]. Pengujian akurasi dilakukan dengan tujuan dapat mengetahui tingkat ketepatan model dalam memprediksi data baru. Nilai akurasi dapat diperoleh dengan menghitung perbandingan antara jumlah prediksi yang benar dengan jumlah total data yang diprediksi. Adapun nilai akurasi dapat dihitung seperti pada persamaan (5):

$$Accuracy = \frac{jumlah\ prediksi\ benar}{jumlah\ total\ prediksi} \quad (5)$$

3. Hasil dan Diskusi

Data yang digunakan dalam penelitian diperoleh melalui metode *web crawling* dengan memanfaatkan pemanggilan API pada module **google-play-scraper** yang tersedia dalam bahasa pemrograman Python[15]. Dengan menyertakan *application ID* pada pemanggilan fungsi **reviews_all** yang terdapat pada module, akan diperoleh data ulasan aplikasi dalam bentuk **numpy array** yang memiliki 10 kolom dalam atribut **review**. Adapun hasil pada tahap pengumpulan data dapat dilihat pada Gambar 5:

	reviewId	userName	userImage	content	score	thumbsUpCount	reviewCreatedVersion	at	replyContent	repliedAt
0	881c6605-9c9a-4335-9fc2-7be29ee06eb3	Romli	https://play-lh.googleusercontent.com/a-/ACNPE...	Aplikasi yg nyusahin, Mau verifikasi NIK aja s...	1	0	1.2.52	2022-09-20 07:55:12	None	NaT
1	41e3b0d2-c701-49f8-8bac-c032c390a60d	Mugi adi	https://play-lh.googleusercontent.com/a-/ACNPE...	Aplikasi bagus, dan sangat membantu utk update...	5	2	1.2.50	2022-09-06 21:24:05	Hi, Kak Mugi adi. Terima kasih atas ulasannya....	2022-09-07 14:21:53
2	078778ef-62b2-4065-8f27-75822ac63399	Jamaludin Azza	https://play-lh.googleusercontent.com/a-/ACNPE...	Apps nya bagus untuk daftar vaksin booster, un...	5	12	1.2.48	2022-08-09 10:33:07	Hi, Kak Jamaludin. Terima kasih atas ulasannya...	2022-08-10 15:35:59
3	cc47f845-d783-49d7-b2ea-9bdfa2b71d53	Boby Kurnia	https://play-lh.googleusercontent.com/a/ALm5wu...	Sangat tidak bermanfaat dan tidak menguntungka...	1	18	1.2.47	2022-07-22 09:56:39	Hai, Kak Boby Kurnia. Mohon maaf atas kendala ...	2022-07-22 15:04:00
4	22a90320-f604-44c3-be50-06b51925b347	Wayne East java (Wayne)	https://play-lh.googleusercontent.com/a-/ACNPE...	Aplikasi ini masih tahap uji coba , masih prog...	3	8	1.2.47	2022-07-30 19:51:07	None	NaT
...	...	...	...	...	...	...	...	...	...	...
9718	b38478d8-db07-4254-8a65-e3c4f6a82daa	Yudi Odon	https://play-lh.googleusercontent.com/a-/ACNPE...	Mantull	5	0	None	2020-06-17 00:00:39	Sampurasun! Terima kasih atas rating dan ulasa...	2020-06-18 13:56:03
9719	90bc3fe2-df6c-4bcc-b745-83d43f197266	Dadang Sudrajat	https://play-lh.googleusercontent.com/a-/ACNPE...	Ok	4	0	None	2020-09-29 12:18:55	Sampurasun! Terima kasih atas rating dan ulasa...	2020-10-02 14:44:18
9720	63ef3c05-7fe8-4d32-afe3-dc99d62ecede	Bank Sampah sukamaju sejahtera Padalarang	https://play-lh.googleusercontent.com/a-/ACNPE...	Goof	5	0	None	2022-04-19 13:05:40	Sampurasun wargi jabar terimakasih atas masuka...	2022-04-20 10:13:25
9721	1a86e9ce-eddc-495a-b701-61492548aab6	Siti Nurlaila2	https://play-lh.googleusercontent.com/a/ALm5wu...	Good	5	0	None	2020-05-03 02:25:25	Terima kasih banyak atas reviewnya. Tetap meng...	2020-05-27 18:38:58
9722	6187da6f-70e3-4160-942c-1a7087272524	Eart Tv	https://play-lh.googleusercontent.com/a-/ACNPE...	Ari amang sabagai naon install apk Sapawarga? ...	1	3	None	2020-09-14 10:43:03	Sampurasun! Untuk sementara ini Sapawarga hany...	2020-09-16 19:06:37

9723 rows × 10 columns

Gambar 5. Hasil pengumpulan data ulasan melalui metode *web crawling*

Setelah pengumpulan data dilakukan, diperoleh data ulasan sebanyak 9.723 data yang memiliki panjang karakter ulasan yang beragam dan memiliki 9 fitur lainnya. Oleh karena itu, dilakukan penyeragaman data dengan menghilangkan kolom yang tidak diperlukan dan menghilangkan data yang memiliki rentang panjang karakter ulasan pada 60 dan 100 karakter. Adapun hasil penyeragaman data dapat dilihat pada Gambar 6:

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1356 entries, 13 to 9613
Data columns (total 4 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   content               1356 non-null   object
 1   score                 1356 non-null   int64
 2   reviewCreatedVersion  1110 non-null   object
 3   contentLength         1356 non-null   int64
dtypes: int64(2), object(2)
memory usage: 53.0+ KB
```

Gambar 6. Hasil penyeragaman data ulasan

Data ulasan aplikasi pelayanan masyarakat yang memiliki jumlah sebesar 1.356 data ulasan memasuki tahap *preprocessing* terlebih dahulu untuk menyeragamkan isi dari data yang diperoleh. Adapun hasil dari setiap tahap *preprocessing* data dapat dilihat pada Tabel 1:

Tabel 1. Hasil Preprocessing

Process No.	Process Title	Result
1	**Data Awal**	Sangat bagus aplikasi nya utk lapor masalah sangat cepat responnya..
2	***Case Folding (filtered)***	sangat bagus aplikasi nya utk lapor masalah sangat cepat responnya..
3	***Filtering (cleaned)***	sangat bagus aplikasi utk lapor masalah sangat cepat responnya
4	***Tokenizing***	[‘sangat’, ‘bagus’, ‘aplikasi’, ‘utk’, ‘lapor’, ‘masalah’, ‘sangat’, ‘cepat’, ‘responnya’]
5	***Spelling Correction***	[‘sangat’, ‘bagus’, ‘aplikasi’, ‘untuk’, ‘lapor’, ‘masalah’, ‘sangat’, ‘cepat’, ‘responnya’]

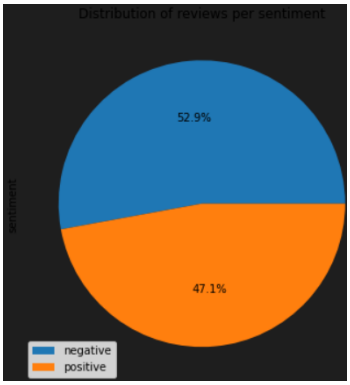
Setelah dilakukan tahap *preprocessing* terhadap dataset, data dilakukan pelabelan ulang menggunakan acuan skor yang terdapat pada data, dimana untuk data dengan skor 4-5 akan dilabei dengan nilai 1 yang mengindikasikan data dengan sentimen positif, dan data dengan skor 1-3 akan dilabeli dengan nilai 0 yang mengindikasikan data dengan sentimen negatif. Adapun hasil dari pelabelan sentimen terhadap data ulasan dapat dilihat pada Gambar 7:

	content	score	spell	sentiment
0	Sangat bagus aplikasi nya utk lapor masalah sa...	5	['sangat', 'bagus', 'aplikasi', 'untuk', 'lapo...	1
1	ga jelas bikin emosi donk. teknologi itu bahay...	1	['tidak', 'jelas', 'emosi', 'teknologi', 'itu'...	0
2	Kenapa applikasi jaki tidak bisa dibuka, lebih...	1	['kenapa', 'applikasi', 'jaki', 'tidak', 'bisa...	0
3	Tidak bisa cek pajak kendaraan, selalu eror da...	1	['tidak', 'bisa', 'cek', 'pajak', 'kendaraan',...	0
4	Jaki merupakan aplikasi yang sangat praktis da...	5	['jaki', 'merupakan', 'aplikasi', 'sangat', 'p...	1
...	...	...	...	...
1351	Rt8 rw4 Dusun cipeuteuy.desa giri laya .kec Pa...	3	['rw', 'dusun', 'cipeuteuy', 'desa', 'giri', '...	0
1352	Aslamualaikum sampurasun nama dudung alamat xe...	1	['aslamualaikum', 'sampurasun', 'nama', 'dudun...	0
1353	Gimana cara daftarnya? Mohon bantuan nya ini u...	1	['gimana', 'cara', 'daftarnya', 'mohon', 'bant...	0
1354	Teuing ah teu valid pemerintah teh nyieun apli...	1	['teuing', 'teu', 'valid', 'pemerintah', 'teh'...	0
1355	sampurasun! upami anggota bpd tiasa nga akses ...	2	['sampurasun', 'upami', 'anggota', 'bpd', 'tia...	0

1356 rows × 4 columns

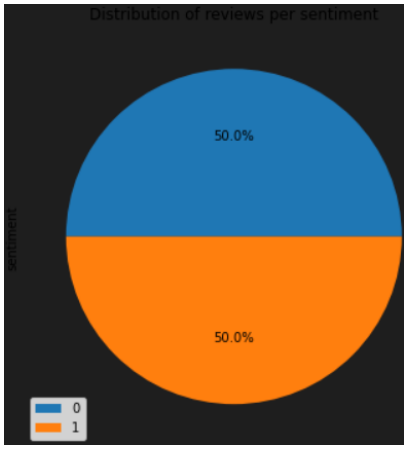
Gambar 7. Hasil pelabelan sentimen data ulasan

Kemudian, data diseimbangkan terlebih dahulu berdasarkan jumlah kelas sentimen yang lebih sedikit. Pada dataset yang digunakan, terdapat 717 data yang memiliki label sentimen negatif, dan 639 data yang memiliki label sentimen positif. Adapun gambaran mengenai bagaimana data setelah pelabelan sentimen terlihat seperti Gambar 8:



Gambar 8. Distribusi data ulasan berdasarkan sentimen setelah pelabelan

Oleh karena itu, data dengan sentimen negatif dikurangi sebanyak 78 data agar dapat mengimbangi data sentimen positif. Sehingga, setelah dilakukan penyeimbangan dataset, terdapat 1.278 data yang dapat digunakan untuk melakukan analisis sentimen. Adapun hasil penyeimbangan data ulasan terlihat seperti pada Gambar 9:



Gambar 9. Distribusi data ulasan berdasarkan sentimen setelah penyeimbangan data

Kemudian, data dari kedua kelas sentimen digunakan untuk melakukan ekstraksi fitur. Sebelum memasuki tahap ekstraksi fitur, data dibagi menjadi dataset latih, dataset validasi, dan dataset uji terlebih dahulu yang terdistribusi menjadi 56% data latih, 24% data validasi dan 20% data uji. Sehingga setelah dibagi, rincian distribusi data yang diperoleh terlihat seperti pada Tabel 2:

Tabel 2. Distribusi Data

	Kelompok Data	Jumlah Data
1	**Data Latih**	715
2	**Data Validasi**	307
3	**Data Uji**	256
	Total	1278

Kemudian, dilakukan ekstraksi fitur, pembobotan dan vektorisasi data menggunakan metode TF – IDF terhadap data latih yang akan digunakan untuk membangun model klasifikasi sentimen dengan menggunakan algoritma *Random Forest*. Adapun hasil dari tahap ekstraksi fitur, pembobotan dan vektorisasi data dapat dilihat pada Gambar 10, Gambar 11 dan Gambar 12:

```
print("Total kata hasil ekstraksi : ",len(hasil_ekstraksi), "\n", hasil_ekstraksi)
✓ 0.5s

Total kata hasil ekstraksi :  1648
 ['a', 'aaamiin', 'aah', 'aamiiin', 'aamin', 'abal', 'abdi', 'adaya', 'adik', 'adil', '
'afdhol', 'aflikasi', 'aga', 'ahli', 'ahmad', 'ahok', 'ajah', 'ajak', 'ajh', 'ajiiigggg
'aktivasinya', 'akun', 'akuntabilitas', 'akurasi', 'akurat', 'ala', 'alamat', 'alamatny
'alokasi', 'aman', 'amanah', 'ambil', 'amblas', 'ambulance', 'amdin', 'amen', 'amiiin',
'anggaran', 'ani', 'anjing', 'ank', 'anonim', 'antrian', 'apapn', 'apapun', 'aparat', '
'aplikasiny', 'aplikasinya', 'aplikasinyaa', 'apliksi', 'aplisaki', 'aplk', 'apload', '
'asfirasi', 'asli', 'asn', 'aspirasi', 'assalammualaikum', 'assessment', 'astaghfirulla
'bahaya', 'bakti', 'balai', 'bale', 'banding', 'bandung', 'bangga', 'bangka', 'bangodua
'bantu', 'bantuan', 'bantuannya', 'banyaj', 'bapuk', 'barat', 'barcode', 'barcodenya',
'bencana', 'beneran', 'beneranlah', 'ber', 'beras', 'berat', 'berbagi', 'berbayar', 'be
```

Gambar 10. Hasil tahap ekstraksi fitur dengan TF-IDF

Pada Gambar 10, terlihat hasil dari ekstraksi fitur yang memuat setiap kata yang terdapat dalam setiap ulasan pada data latih yang diproses menggunakan metode TF – IDF. Adapun total jumlah fitur yang berhasil diekstrak menggunakan metode ini adalah sebanyak 1.648 fitur dari 715 total dokumen.

```
  Output exceeds the size limit. Open the full ou
    (0, 1604)     0.30712464318582094
    (0, 1430)     0.250552342337264
    (0, 1270)     0.23322593083280235
    (0, 972)      0.3543084181016948
    (0, 871)      0.3186152702683444
    (0, 779)      0.3543084181016948
    (0, 384)      0.33342926508648824
    (0, 277)      0.2714314953524705
    (0, 215)      0.3543084181016948
    (0, 114)      0.3543084181016948
    (1, 1605)     0.23783856063035155
```

Gambar 11. Hasil pembobotan fitur menggunakan metode TF-IDF

Pada Gambar 11, terlihat hasil pembobotan TF-IDF terhadap data latih ulasan aplikasi yang memuat nilai bobot setiap kata dalam dokumen. Dimana nilai bobot ini akan digunakan untuk membangun mesin klasifikasi dengan algoritma *Random Forest*. Namun, sebelum memasuki algoritma klasifikasi, nilai-nilai bobot fitur harus dilakukan vektorisasi terlebih dahulu agar dapat diterima sebagai masukan oleh mesin dan dapat melatih mesin.

	a	aaamiin	aah	aamiiin	aamin	abal	abdi	adaya	adik	adil	...	\
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
..	...	...	...	...	...	...	...	...	...	...	...	
710	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
711	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
712	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
713	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
714	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	

	whatsapp	wifi	wilayah	ya	yaa	yaaa	yb	yogyakarta	yra	zaman
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
..	...	...	...	...	...	...	...	...	...	...
710	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
711	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
712	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
713	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
714	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

[715 rows x 1648 columns]

Gambar 12. Hasil vektorisasi fitur pada data latih

Pada Gambar 12, terlihat hasil vektor fitur yang diperoleh berdasarkan nilai bobot setiap fitur terhadap setiap dokumen yang terdapat pada data latih. Vektor fitur tersebut akan digunakan sebagai masukan untuk algoritma *Random Forest* dalam membangun mesin klasifikasi sentimen. Setelah vektor fitur diperoleh melalui metode TF – IDF yang telah dilakukan, maka langkah berikutnya adalah melatih model klasifikasi sentimen.

Dalam membangun model klasifikasi sentimen dengan algoritma *Random Forest*, digunakan module **RandomForestClassifier** dari *library* Scikit-learn yang menggunakan bahasa pemrograman Python. Untuk membangun hutan, vektor fitur dimuat pada objek RandomForestClassifier dengan menggunakan metode **fit()**. Setelah vektor fitur dimual, maka

mesin klasifikasi sentimen ulasan aplikasi akan diperoleh. Adapun pembuatan model klasifikasi sentimen terlihat seperti pada Gambar 13:

```
rf0 = RandomForestClassifier()
rf0.fit(X_train, y_train.values.ravel())

X_val_ = vectorize(X_val, tfidf_vect_fit)
from sklearn.metrics import accuracy_score,precision_score,recall_score

for mdl in [rf0]:
    y_pred = mdl.predict(X_val_)
    accuracy = round(accuracy_score(y_val,y_pred), 3)
    precision = round(precision_score(y_val,y_pred), 3)
    recall = round(recall_score(y_val,y_pred), 3)
    print('MAX DEPTH: {} \nMAX FEATURES : {} \nMIN SAMPLE LEAF : {} \n# OF E
        mdl.max_depth,
        mdl.max_features,
        mdl.min_samples_leaf,
        mdl.n_estimators,
        mdl.n_jobs,
        accuracy,
        precision,
        recall))
✓ 0.7s

MAX DEPTH: None
MAX FEATURES : auto
MIN SAMPLE LEAF : 1
# OF EST: 100
# OF JOBS : None
-- A: 0.805 / P: 0.826 / R: 0.76
```

Gambar 13. Hasil pembuatan model klasifikasi tanpa *hyperparameter tuning*

Setelah pelatihan model klasifikasi dilakukan, diperoleh nilai akurasi sebesar 80%. Kemudian, dilakukan *tuning* terhadap *hyperparameter* dari algorima *Random Forest* yang meliputi: kedalaman maksimum setiap *tree***,** jumlah maksimum fitur**,** jumlah sampel minimum pada node *leaf***,** jumlah *tree* pada hutan dan jumlah pekerjaan yang dilakukan CPU secara bersamaan[16]. Adapun beberapa hasil dari *hyperparameter tuning* algorimta *Random Forest* terhadap data latih dari ulasan aplikasi pelayanan masyarakat adalah seperti pada Tabel 3:

Tabel 3. Nilai Akurasi Model Klasifikasi Setelah *Hyperparameter Tuning*

#	max_depth	max_features	min_samples_leaf	n_estimators	n_jobs	Akurasi
1	10	log2	1	800	1	81.3%
2	20	log2	1	800	1	80.7%
3	None	log2	2	800	2	80.4%
4	10	log2	1	800	2	80.3%
5	None	log2	1	400	4	80.1%

Setelah itu, dilakukan validasi dan uji coba terhadap model klasifikasi yang telah dibangun dengan dilakukan *hyperparameter tuning* dengan menggunakan kelompok data validasi dan data uji yang sebelumnya telah didistribusikan, yang menghasilkan nilai akurasi seperti pada Tabel 4:

Tabel 4. Perbandingan Nilai Akurasi Model Terhadap Data Latih, Validasi dan Uji

#	Akurasi Latih	Akurasi Validasi	Akurasi Uji
1	81,3%	82,1%	84%
2	80,7%	84%	84%
3	80,4%	83,1%	82,4%
4	80,3%	84%	82,4%
5	80,1%	81,4%	84%

Berdasarkan validasi dan uji data terhadap mesin klasifikasi sentimen *Random Forest* pada kelima mesin yang telah dibangun dengan *hyperparameter* berbeda, diperoleh nilai akurasi tertinggi dan stabil terdapat pada mesin kelima, dengan nilai akurasi prediksi data uji tertinggi yaitu pada angka 84%, dengan nilai akurasi latih yang stabil atau tidak berubah secara signifikan dengan nilai akurasi validasi.

4. Kesimpulan

Berdasarkan hasil evaluasi, diperoleh bahwa penggunaan algoritma *Random Forest* beserta *tuning* terhadap *hyperparameter Random Forest* dapat memberikan hasil yang cukup baik dalam melakukan analisis atau klasifikasi sentimen terhadap ulasan pengguna aplikasi pelayanan masyarakat. Jika dibandingkan dengan penelitian – penelitian terkait, penggunaan algoritma *Random Forest* mampu memberikan hasil klasifikasi sentimen yang baik dan mampu bersaing dengan algoritma lainnya seperti *Support Vector Machine*, *Naïve Bayes, dan Modified Balanced Random Forest* [4][2][3]. Dengan melakukan *hyperparameter tuning*, diperoleh peningkatan hasil akurasi model yang sebelumnya bernilai 80% tanpa melakukan *hyperparameter tuning* menjadi 84% dengan melakukan *hyperparameter tuning*.

Daftar Pustaka

[1] W. Paulina, F. A. Bachtiar and N. Rusydi “Analisis Sentimen Berbasis Aspek Ulasan Pelanggan Terhadap Kertanegara Premium Guest House Menggunakan Support Vector Machine”. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 4, no. 4, p. 1141-1149, 2020.

[2] Rosdiana, E. Tungadi, Z. Saharuna and M. N. Y. Utomo, M. N. “Analisis Sentimen pada Twitter terhadap Pelayanan Pemerintah Kota Makassar”. *Jurnal Kategori Teknik Komputer dan Jaringan,* p 87-93, 2019

[3] F. N. Zamzami, K. Adiwijaya and M. Dwifebri. “Analisis Sentimen Terhadap Review Film Menggunakan Metode Modified Balanced Random Forest dan Mutual Information”. *Jurnal Media informatika Budidarma*, p 415-421, 2021

[4] S. Ailiyya, “Analisis Sentimen Berbasis Aspek Pada Ulasan Aplikasi Tokopedia Menggunakan Support Vector Machine”, Universitas Islam Negeri Syarif Hidayatullah, 2020.

[5] A. Kadhim. “An Evaluation of Preprocessing Techniques for Text Classification”. *International Journal of Computer Science and Information Security (LISCIS),* p. 22-32, 2018

[6] O. Gilbert, “Sentiment Analysis with TFIDF and Random Forest”, 5 May 2020. [Online]. Available: https://www.kaggle.com/code/onadegibert/sentiment-analysis-with-tfidf-and-random-forest [Accessed on 22 September 2022]

[7] Adeputri, “text-preprocessing”, 9 July 2021. [Online]. Available: https://github.com/adeputri123/text-preprocesing [Accessed on 24 September 2022]

[8] Y. A. Rohman. “Spell Check Bahasa Indonesia menggunakan Pre-trained Word Vectors Fasttext Model”. 3 March 2020. [Online]. Available: https://medium.com/@yasirabd/spell-check-indonesia-menggunakan-pre-trained-fasttext-model-14e90a3f1ac0 [Accessed on 25 September 2022]

[9] U. Parida, M. Nayak and A. K. Nayak, "News Text Categorization using Random Forest and Naïve Bayes," *2021 1st Odisha International Conference on Electrical Power Engineering, Communication and Computing Technology(ODICON)*, p. 1-4, 2021

[10] K. Kirasich, T. Smith and B. Sadler. “Random Forest vs Logistic Regression: Binary Classification for Heterogeneous Datasets”. *SMU Data Science Review,* vol. 1, no. 3, p. 1-24, 2018

[11] M. Abuella and B. Chowdhury, "Random forest ensemble of support vector regression models for solar power forecasting," *2017 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, p. 1-5, 2017

[12] Nuranisah, "Analisis Menggunakan Random Forest Dengan Gini Index Algoritma Pada Data", 2021 *Seminar of Social Sciences Engineering & Humaniora (SCENARIO),* p. 19-24, 2021

[13] A. Dhar, N.S. Dash and K. Roy, "Application of TF-IDF Feature for Categorizing Documents of Online Bangla Web Text Corpus". *Intelligent Engineering Informatics,* p. 51-59, 2018

[14] M. S. Kumar, V. Soundarya, S. Kavitha, E. S. Keerthika and E. Aswini, "Credit Card Fraud Detection Using Random Forest Algorithm", *2019 3rd International Conference on Computing and Communications Technologies (ICCCT)*, pp. 149-153, 2019

[15] J. Mingyu, "Google-Play-Scraper", 19 August 2022. [Online]. Available: https://pypi.org/project/google-play-scraper/ [Accessed on 25 September 2022]

[16] P. Probst, MN. Wright and A-L. Boulesteix, "Hyperparameters and tuning strategies for random forest". *WIREs Data Mining Knowledge Discovery*, 2019

Halaman ini sengaja dibiarkan kosong