

PENERAPAN METODE *ADAPTIVE BOOSTING (ADABOOST)* PADA *DECISION TREE* UNTUK ANALISIS SENTIMEN PELANGGAN MAXIM

Erni Triana^{1*}, Mustafid², Rukun Santoso³

^{1,2,3} Departemen Statistika, Fakultas Sains dan Matematika, Universitas Diponegoro

*e-mail: ernitriana16@gmail.com

DOI: 10.14710/j.gauss.14.2.608-618

Article Info:

Received: 2024-12-05

Accepted: 2025-12-29

Available Online: 2025-12-30

Keywords:

Online Transportation; Adaptive Boosting; Decision Tree; Grid Search.

Abstract: Information technology is currently growing rapidly, one form of technology beneficiary is using the internet, namely online transportation services based on mobile applications. Maxim is one of the online transportation services in Indonesia that offers relatively cheaper prices compared to other online transportation services. This study aims to apply the Adaptive Boosting (Adaboost) method with Decision Tree to classify Maxim's customer review data so that it can establish customer satisfaction factors. Review data was obtained from June – December 2022 with a total of 1500 reviews. Classification was carried out using the Adaptive Boosting method with a Decision Tree and Tuning Hyperparameter Grid Search. Adaptive Boosting is used to improve the performance of the Decision Tree so it can work better. The Grid Search algorithm is used to determine the best hyperparameter combination in Adaptive Boosting so that the classification process can be more optimal. Classification using the Adaptive Boosting model with Decision Tree yields accuracy, precision and recall values of 83,69%, 86,75% and 85,71% with the best parameter combination based on Grid Search is *n_estimator* (number of trees) 300 and learning rate 0,001. Based on this accuracy value, it can be concluded that the Adaptive Boosting model is quite good at classifying Maxim's customer review data.

1. PENDAHULUAN

Perkembangan teknologi informasi yang semakin pesat menjadi sarana untuk mengumpulkan informasi dengan mudah dan cepat dengan adanya internet. Selain digunakan sebagai untuk mengumpulkan informasi, penggunaan internet juga dapat dimanfaatkan untuk menunjang aktivitas suatu perusahaan salah satunya perusahaan jasa transportasi, yaitu dengan membuat inovasi transportasi ojek *online* berbasis aplikasi *mobile*. Salah satu layanan transportasi tersebut adalah Maxim yang merupakan perusahaan transportasi asal Rusia yang memasuki Indonesia pada tahun 2013. Maxim menawarkan harga yang lebih terjangkau jika dibandingkan dengan layanan ojek *online* lain. Pelanggan atau pengguna aplikasi Maxim dapat memberikan ulasan opini mereka terhadap layanan Maxim pada situs *Google Play Store*. Ulasan tersebut dapat menjadi penilaian terhadap layanan suatu aplikasi agar dapat memperbaiki kualitasnya. Analisis sentimen diperlukan untuk memudahkan pengolahan data yang berbentuk teks dengan mengklasifikasikan beberapa teks dan menentukan suatu pendapat yang diutarakan dalam dokumen tersebut bersifat positif, negatif, atau netral (Liu, 2012). Metode klasifikasi yang digunakan yaitu *Adaptive Boosting (Adaboost)* dengan *Decision Tree* sebagai *base learner* atau algoritma pengklasifikasi dasar dan *Tuning hyperparameter* menggunakan *Grid Search* untuk menentukan kombinasi parameter terbaik sehingga klasifikasi dapat berjalan dengan lebih optimal.

Penelitian ini bertujuan untuk menerapkan metode *Adaptive Boosting (Adaboost)* pada *Decision Tree* dalam mengklasifikasikan data ulasan pelanggan Maxim ke dalam kelas

sentimen positif dan negatif sehingga dapat membentuk faktor-faktor kepuasan pelanggan terhadap layanan Maxim.

2. TINJAUAN PUSTAKA

Text mining merupakan proses mengekstraksi informasi dari sejumlah sumber data tekstual melalui identifikasi pola yang menarik. Sumber data yang digunakan dalam *text mining* adalah kumpulan data tekstual yang tidak terstruktur pada dokumen (Feldman dan Sanger, 2007). Analisis sentimen atau *opinion mining* merupakan sebuah proses untuk menganalisis data tekstual dalam suatu entitas seperti produk, layanan dan topik tertentu untuk mengetahui pendapat, sentimen, penilaian, sikap dan aspek lainnya yang mewakili ruang masalah yang besar (Liu, 2012).

Text pre-processing merupakan tahap awal pengolahan data dengan *text mining* yang dilakukan untuk meningkatkan kualitas data. *Pre-processing* data bertujuan untuk mempersiapkan data tekstual yang akan digunakan agar dapat diproses pada tahap berikutnya. Tahap *pre-processing* pada penelitian ini meliputi:

1. *Case Folding*
Case folding merupakan tahapan untuk mengubah semua huruf menjadi huruf kecil. Tujuan *case folding* untuk menghindari kata-kata dengan arti sama tapi diperlakukan berbeda oleh program.
2. *Remove Number*
Remove number merupakan tahapan untuk menghapus karakter angka pada dokumen. Angka tidak dapat menunjukkan suatu perasaan sehingga perlu dihapus pada proses *text mining*.
3. *Remove Punctuation*
Remove punctuation merupakan tahap penghapusan tanda baca pada dokumen karena tidak menambahkan informasi tambahan.
4. *Remove Emoticon*
Remove emoticon merupakan tahap penghapusan simbol/emoji pada dokumen teks. Simbol/emoji tidak dapat memberikan penilaian sehingga perlu dihapus.
5. Normalisasi Kata
Normalisasi kata merupakan proses pengubahan kata tidak baku menjadi kata baku sesuai dengan KBBI dengan bantuan kamus.

Feature selection merupakan tahap untuk mengurangi kata yang tidak bermakna dengan tujuan agar proses klasifikasi menjadi lebih efektif dan akurat (Feldman dan Sanger, 2007). Tahap *feature selection* pada penelitian ini meliputi:

1. *Stopwords Removal*
Stopwords removal merupakan tahap penghapusan kata-kata paling umum yang sering muncul pada dokumen dan tidak bermakna.
2. *Stemming*
Stemming merupakan tahap penghapusan imbuhan pada kata atau mengubah kata menjadi bentuk kata dasarnya tanpa imbuhan.
3. *Tokenizing*
Tokenizing merupakan proses memecah dokumen menjadi token atau bagian kecil untuk memudahkan pada proses selanjutnya.

Proses pelabelan data pada penelitian ini menggunakan *sentiment scoring* untuk menentukan kelas data teks menjadi positif atau negatif. *Sentiment scoring* menghitung skor akhir untuk setiap kata sentimen dalam sebuah dokumen dengan bantuan kamus sentimen. Data dengan skor akhir > 0 dikategorikan sebagai positif, sedangkan jika skor akhir < 0 dikategorikan sebagai negatif (Wahid dan Azhari, 2016).

Metode TF-IDF merupakan salah satu metode untuk memberikan skor atau nilai untuk sebuah kata (*term*) dalam dokumen. Bobot kata tersebut menjadi tolak ukur seberapa besar tingkat kontribusi sebuah kata untuk pengkategorian suatu kelas dalam suatu dokumen (Deolika *et al.*, 2019). Nilai TF-IDF dapat dihitung dengan persamaan (1) (Sallton dan Buckley, 1998):

$$W_{j,i} = \frac{n_{j,i}}{\sum_{i=1}^k n_{k,i}} \cdot \log_2 \frac{D}{d_j} \quad (1)$$

dengan $W_{j,i}$ merupakan pembobotan TF-IDF untuk *term* ke- j pada dokumen ke- i , $n_{j,i}$ merupakan jumlah kemunculan *term* ke- j pada dokumen ke- i , $\sum_{i=1}^k n_{k,i}$ merupakan jumlah kemunculan seluruh *term* pada dokumen ke- i , D merupakan banyaknya dokumen yang dibangkitkan dan d_j merupakan banyaknya dokumen yang mengandung *term* ke- j .

Klasifikasi merupakan teknik pada data mining untuk memetakan data ke dalam kelompok atau kelas yang telah ditentukan. Pada proses klasifikasi data akan dibagi menjadi dua yaitu, data *training* dan data *testing*. Data *training* digunakan untuk membangun model yang kemudian model tersebut akan diuji ke data *testing*.

Decision Tree atau pohon keputusan memiliki konsep dasar mengubah data menjadi model pohon keputusan, kemudian mengubah model pohon menjadi *rule* (aturan) dan menyederhanakan *rule* (Han *et al.*, 2012). *Decision tree* membagi kelas berikutnya menjadi akar pohon yang lebih spesifik dengan memilih data yang memiliki nilai *information gain* tertinggi sebagai akar (*node root*). Langkah-langkah klasifikasi dengan *decision tree* adalah sebagai berikut (Brownlee, 2020):

1. Menentukan nilai *entropy* dengan persamaan (2).

$$Entropy(S) = - \sum_{i=1}^n p_i \times \log_2 p_i \quad (2)$$

S merupakan himpunan kasus, n merupakan proporsi sampel pada partisi ke- i dalam S , dan p_i merupakan proporsi dari S_i terhadap S .

2. Menentukan nilai *information gain* dengan persamaan (3).

$$gain(S, A) = Entropy(S) - \sum_{i=1}^n \frac{|S_i|}{|S|} \times Entropy(S_i) \quad (3)$$

A merupakan atribut, $|S_i|$ merupakan jumlah kasus pada partisi ke- I , dan $|S|$ merupakan jumlah kasus dalam S .

3. Memilih atribut sebagai akar dengan nilai *information gain* tertinggi.

Metode *Adaptive Boosting* (*Adaboost*) merupakan metode *ensemble* yang digunakan untuk meningkatkan performa algoritma klasifikasi dasar yang pada penelitian ini menggunakan *Decision Tree*. *Adaboost* memberikan bobot yang lebih besar pada amatan yang diklasifikasikan tidak tepat sehingga pada iterasi selanjutnya amatan yang sulit diklasifikasi menerima pengaruh yang lebih besar. Langkah-langkah klasifikasi dengan *Adaboost* adalah sebagai berikut (Hastie *et al.*, 2008):

1. Menentukan bobot awal w_i dengan $i = 1, 2, 3, \dots, N$

$$w_i = \frac{1}{N} \quad (4)$$

2. Untuk setiap iterasi h (*weak learner*), dengan $h = 1, 2, 3, \dots, H$ lakukan hal berikut:
 - a. Sampel dataset menggunakan bobot w_i^h untuk mendapatkan sampel *training* x_i .

- b. Menentukan penduga klasifikasi menggunakan *Decision Tree* dengan memilih variabel yang memiliki nilai *gain* tertinggi dengan persamaan (3).
- c. Menghitung nilai *error* atau kesalahan klasifikasi dengan persamaan (5).

$$err_h = \frac{\sum_{i=1}^N w_i^h I[G_h(x_i) \neq y_i]}{\sum_{i=1}^N w_i^h} \quad (5)$$

w_i^h merupakan bobot awal amatan pada iterasi ke- h , $G_h(x_i)$ merupakan nilai prediksi pada data latih dan $I[G_h(x_i) \neq y_i]$ merupakan fungsi indikator yang akan bernilai 1 apabila data salah diklasifikasikan dan bernilai 0 apabila data diklasifikasikan dengan benar.

- d. Menghitung koefisien α_h atau tingkat kepercayaan atas prediksi *weak learner* pada iterasi h yang dinyatakan dengan persamaan (6).

$$\alpha_h = \ln\left(\frac{1 - err_h}{err_h}\right) \quad (6)$$

- e. Memperbarui bobot data yang diklasifikasikan tidak tepat dengan persamaan (7).

$$w_i^{h+1} = w_i \exp(\alpha_h I[G_h(x_i) \neq y_i]), i = 1, 2, 3, \dots, N \quad (7)$$

3. Sehingga diperoleh output akhir prediksi pada persamaan (8).

$$G(x) = \text{sign} \left[\sum_{h=1}^H \alpha_h G_h(x) \right] \quad (8)$$

Tanda $\text{sign}([\])$ merupakan fungsi untuk mengembalikan lambang angka. $G(x)$ akan bernilai -1 jika $[\sum_{h=1}^H \alpha_h G_h(x)] < 0$ dan bernilai 1 jika $[\sum_{h=1}^H \alpha_h G_h(x)] > 0$.

Tuning hyperparameter atau penyetelan *hyperparameter* merupakan metode untuk menentukan nilai parameter yang diatur agar mendapatkan model yang optimal. Parameter pada metode *Adaptive Boosting* yaitu *n_estimators* (jumlah pohon) dan *learning rate*. *Tuning hyperparameter* pada penelitian ini menggunakan metode *Grid Search*. *Grid Search* merupakan metode pencarian menyeluruh berdasarkan subset ruang *hyperparameter* berdasarkan sejumlah angka yang telah ditentukan menggunakan nilai minimal (*lower bound*) dan nilai maksimal (*upper bound*) (Syarif *et al.*, 2016).

Kinerja sistem klasifikasi perlu diuji untuk memberikan gambaran seberapa baik suatu metode dalam mengklasifikasikan data. Salah satu pengukuran kinerja klasifikasi umumnya dilakukan dengan nilai representasi *confusion matrix*.

Tabel 1. Confusion Matrix

Prediksi	Aktual	
	Positif	Negatif
Positif	<i>True Positive (TP)</i>	<i>False Positive (FP)</i>
Negatif	<i>False Negative (FN)</i>	<i>True Negative (TN)</i>

Ukuran atau parameter yang digunakan untuk menilai dan mengevaluasi model klasifikasi antara lain:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (9)$$

$$Precision = \frac{TP}{TP + FP} \quad (10)$$

$$Recall = \frac{TP}{TP + FN} \quad (11)$$

Word cloud merupakan metode visualisasi dokumen teks yang merepresentasikan grafis dari sebuah dokumen dengan *plotting* kata-kata yang sering digunakan pada sebuah dokumen. Frekuensi kemunculan kata ditunjukkan melalui ukuran huruf dari kata tersebut (Castellà dan Sutton, 2014).

3. METODE PENELITIAN

Jenis data yang digunakan pada penelitian ini adalah data sekunder berupa data hasil *web scraping* ulasan aplikasi Maxim pada situs *Google Play Store* untuk periode bulan Juni – Desember 2022 sebanyak 1500 data ulasan.

Tahapan analisis data yang dilakukan pada penelitian ini adalah sebagai berikut:

1. *Scraping* data ulasan aplikasi Maxim
2. *Pre-processing* data
3. *Feature selection*
4. Pelabelan data dengan *sentiment scoring*
5. Pembobotan kata TF-IDF
6. Membagi data menjadi data *training* dan data *testing*
7. *Tuning hyperparameter* dengan *Grid Search*
8. Membangun model klasifikasi *Adaptive Boosting* dengan *Decision Tree*
9. Evaluasi model dengan *Confusion Matrix*
10. Visualisasi data dengan *Word Cloud*

4. HASIL DAN PEMBAHASAN

Data ulasan aplikasi Maxim diperoleh dengan teknik *web scraping* pada situs *Google Play Store* menggunakan bahasa pemrograman *Python* pada *Platform Web Google Collaboratory*. Jumlah data yang diambil sebanyak 1500 data ulasan paling relevan pada periode bulan Juni - Desember 2022.

Data hasil *scraping* merupakan data teks yang tidak terstruktur sehingga belum sepenuhnya dapat diolah untuk proses klasifikasi. Data tersebut akan diproses untuk menghilangkan karakter yang tidak diperlukan berupa angka, tanda baca, simbol/emoji dan memperbaiki kata dengan salah ejaan pada tahap *pre-processing*. Kemudian akan dilakukan pengurangan dimensi data dengan menghapus kata-kata yang tidak bermakna pada tahap *feature selection*.

Pelabelan data ulasan dilakukan dengan menghitung skor kata sentimen pada setiap dokumen menggunakan bantuan kamus. Berdasarkan hasil pelabelan dengan *seniment scoring* diperoleh jumlah sentimen positif sebanyak 872 ulasan, sentimen negatif sebanyak 536 ulasan dan sentimen netral sebanyak 92 ulasan. Karena ulasan netral tidak digunakan pada penelitian ini, maka dilakukan penghapusan data dengan sentimen netral sehingga jumlah data yang akan dianalisis menjadi 1408 data.

Data ulasan yang berbentuk teks harus diubah ke dalam bentuk numerik dalam proses klasifikasi. Metode TF-IDF digunakan untuk mengubah data teks menjadi numerik dengan memberikan bobot nilai pada setiap kata dalam dokumen. Hasil pembobotan kata dengan metode TF-IDF ditunjukkan pada Tabel 2.

Tabel 2. Hasil Pembobotan TF-IDF

Ulasan	akurat	alamat	batal	opsi	suka	...	sanksi
12	0,1095	0	0	0	0,1053	...	0
219	0,3149	0,2702	0	0	0	...	0
345	0,2422	0	0,1907	0	0	...	0
852	0	0	0	0	0	...	0,3639
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1408	0	0	0,2059	0,3868	0	...	0

Data yang telah memiliki bobot akan dibagi menjadi data *training* dan data *testing* dengan proporsi pembagian data *training* 90% dan data *testing* 10%. Pembagian data dilakukan secara acak menggunakan bahasa pemrograman *Python*.

Metode *Adaptive Boosting* dengan *Decision Tree* menggunakan perhitungan nilai *entropy* dan *information gain* untuk menentukan atribut akarnya (*node root*) pada pohon. Data yang memiliki nilai *information gain* tertinggi yang akan digunakan sebagai atribut akar pada pohon keputusan. Contoh *dataset* sederhana yang digunakan untuk perhitungan *entropy* dan *information gain* ditunjukkan pada Tabel 3.

Tabel 3. Contoh *Dataset*

Ulasan	akurat	aplikasi	...	layan	Sentimen
1	0,543163	0,543163	...	0	positif
2	0,489046	0	...	0	negatif
3	0	0	...	0	negatif
4	0	0	...	0,462753	positif
5	0	0	...	0,50519	positif
6	0	0	...	0	negatif
7	0	0	...	0	positif
8	0	0	...	0	negatif
9	0	0	...	0	negatif
10	0	0	...	0	negatif

Tabel 3 merupakan hasil pembobotan TF-IDF untuk 10 dokumen. Setelah menentukan bobot awal dengan persamaan (4), pada perhitungan manual dengan *dataset* yang kecil akan ditentukan nilai *separator* atau pemisah terlebih dahulu untuk mengubah data menjadi diskrit. Pada contoh ini digunakan *separator* sebesar 0,4 karena dari 10 dokumen yang digunakan terdapat 4 dokumen dengan sentimen positif.

Tabel 4. Perhitungan *Entropy* dan *Information Gain*

	Jumlah	Positif	Negatif	Entropy	Gain
Total	10	4	6	0,970951	

akurat	Kurang	8	3	5	0,954434	0,007403
	Lebih	2	1	1	1	
aplikasi	Kurang	9	3	6	0,918296	0,144484
	Lebih	1	1	0	0	
layan	Kurang	8	2	6	0,811278	0,321928
	Lebih	2	2	0	0	

Berikut contoh perhitungan *entropy* dan *information gain* secara manual berdasarkan Tabel 4 menggunakan persamaan (2) dan persamaan (3):

$$Entropy \text{ total} = -\left(\frac{4}{10}\right) \times \log_2 \left(\frac{4}{10}\right) - \left(\frac{6}{10}\right) \times \log_2 \left(\frac{6}{10}\right) = 0,970951$$

$$Entropy (\text{akurat} < 0,4) = -\left(\frac{3}{8}\right) \times \log_2 \left(\frac{3}{8}\right) - \left(\frac{5}{8}\right) \times \log_2 \left(\frac{5}{8}\right) = 0,954434$$

$$Entropy (\text{akurat} > 0,4) = -\left(\frac{1}{2}\right) \times \log_2 \left(\frac{1}{2}\right) - \left(\frac{1}{2}\right) \times \log_2 \left(\frac{1}{2}\right) = 1$$

$$Gain = 0,970951 - \left(\left(\frac{8}{10}\right) \times 0,954434\right) - \left(\left(\frac{2}{10}\right) \times 1\right) = 0,007403$$

$$Entropy (\text{aplikasi} < 0,4) = -\left(\frac{3}{9}\right) \times \log_2 \left(\frac{3}{9}\right) - \left(\frac{6}{9}\right) \times \log_2 \left(\frac{6}{9}\right) = 0,918296$$

$$Entropy (\text{aplikasi} > 0,4) = 0$$

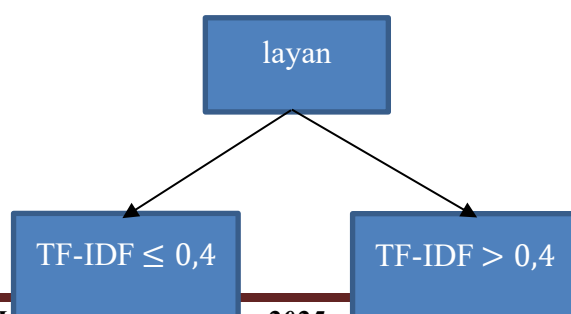
$$Gain = 0,970951 - \left(\left(\frac{9}{10}\right) \times 0,918296\right) - \left(\left(\frac{1}{10}\right) \times 0\right) = 0,144484$$

$$Entropy (\text{layan} < 0,4) = -\left(\frac{2}{8}\right) \times \log_2 \left(\frac{2}{8}\right) - \left(\frac{6}{8}\right) \times \log_2 \left(\frac{6}{8}\right) = 0,811278$$

$$Entropy (\text{layan} > 0,4) = 0$$

$$Gain = 0,970951 - \left(\left(\frac{8}{10}\right) \times 0,811278\right) - \left(\left(\frac{2}{10}\right) \times 0\right) = 0,321928$$

Berdasarkan perhitungan *entropy* dan *information gain* yang telah dilakukan, akan dipilih kata (*term*) dengan nilai *information gain* tertinggi yaitu kata “layan”. Sehingga kata “layan” akan digunakan sebagai *node root* atau akar dari pohon keputusan yang dibentuk.



Negatif

Positif

Gambar 1. Contoh *Decision Tree* yang Terbentuk

Setelah diperoleh pohon keputusan kemudian dilakukan prediksi pada data berdasarkan aturan pada pohon, diperoleh prediksi pada iterasi 1 ditunjukkan pada Tabel 5.

Tabel 5. Hasil Prediksi dari Decision Tree

Ulasan	Klasifikasi	Prediksi	Indikator
1	positif	negatif	1
2	negatif	negatif	0
3	negatif	negatif	0
4	positif	negatif	1
5	positif	positif	0
6	negatif	negatif	0
7	positif	positif	0
8	negatif	negatif	0
9	negatif	negatif	0
10	negatif	negatif	0

Prediksi tersebut akan digunakan untuk menghitung nilai *error* dan α menggunakan persamaan (5) dan (6) sebagai berikut:

$$err_1 = \frac{(0,1(1) + 0,1(0) + 0,1(0) + (0,1(1) + 0,1(0) + \dots + 0,1(0) + 0,1(0))}{(0,1 + 0,1 + 0,1 + 0,1 + 0,1 + 0,1 + 0,1 + 0,1 + 0,1 + 0,1)} = 0,2$$

$$\alpha_1 = \ln\left(\frac{1 - 0,2}{0,2}\right) = 1,386294$$

Tahap terakhir yaitu memperbarui bobot data yang salah diklasifikasikan dan melakukan standarisasi menggunakan persamaan (7) yang ditunjukkan pada Tabel 6.

Tabel 6. Bobot Baru dan Standarisasi data

Ulasan	Bobot Baru	Standarisasi
1	$w_1^2 = 0,1 \exp(1,386294 \times 1) = 0,4$	0,25
2	$w_2^2 = 0,1 \exp(1,386294 \times 0) = 0,1$	0,0625
3	$w_3^2 = 0,1 \exp(1,386294 \times 0) = 0,1$	0,0625
4	$w_4^2 = 0,1 \exp(1,386294 \times 1) = 0,4$	0,25
5	$w_5^2 = 0,1 \exp(1,386294 \times 0) = 0,1$	0,0625
6	$w_6^2 = 0,1 \exp(1,386294 \times 0) = 0,1$	0,0625
7	$w_7^2 = 0,1 \exp(1,386294 \times 0) = 0,1$	0,0625
8	$w_8^2 = 0,1 \exp(1,386294 \times 0) = 0,1$	0,0625
9	$w_9^2 = 0,1 \exp(1,386294 \times 0) = 0,1$	0,0625

10	$w_{10}^2 = 0,1 \exp(1,386294 \times 0) = 0,1$	0,0625
----	--	--------

Bobot yang diperoleh akan digunakan pada perhitungan di iterasi selanjutnya. Proses ini dilakukan berulang hingga mencapai iterasi maksimum yang ditentukan, kemudian dilakukan prediksi akhir dengan menggabungkan *weak learners* pada tiap iterasi menggunakan persamaan (8).

Model klasifikasi pada penelitian ini menggunakan *adaptive boosting* dengan *decision tree* menggunakan parameter *n_estimators* (jumlah pohon) sebanyak 300 dan *learning rate* sebesar 0,001. Parameter tersebut merupakan kombinasi parameter terbaik berdasarkan *Grid Search*. Hasil evaluasi kinerja model *adaptive boosting* ditunjukkan pada Tabel 5.

Tabel 7. Hasil *Confusion Matrix* Model *Adaboost* dengan *Decision Tree*

Prediksi	Aktual	
	Positif	Negatif
Positif	72	12
Negatif	11	46

Hasil *confusion matrix* kemudian digunakan untuk menentukan nilai akurasi, *precision* dan *recall*:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{72 + 46}{72 + 46 + 12 + 11} = 0,8369$$

$$Precision = \frac{TP}{TP + FP} = \frac{72}{72 + 12} = 0,8675$$

$$Recall = \frac{TP}{TP + FN} = \frac{72}{72 + 11} = 0,8571$$

Data ulasan Maxim divisualisasikan dengan representasi grafis dari *word cloud* untuk mengetahui topik apa yang sering dibicarakan oleh pelanggan sehingga dapat membentuk faktor-faktor kepuasan pelanggan Maxim. Hasil visualiasi ulasan positif dan negatif ditunjukkan pada Gambar 2.

DAFTAR PUSTAKA

- Brownlee, J. 2020. *Probability for Machine Learning: Discover How To Harness Uncertainty With Python*. San Fransisco: Machine Learning Mastery.
- Castellà, Q., Sutton, C. 2014. Word storms: Multiples of word clouds for visual comparison of documents. *Proceedings of the 23th International Conference of World*.
- Feldman, R., Sanger, J. 2007. *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. New York:Cambridge.
- Freund, Y., Schapire, R. E. 1996. AiDecision-Theoretic Generalization of ON-Line Learning andian Application to Boosting. *Journal of Computer and System Sciences* Vol.55, Hal: i119-139.
- Han, J., Kamber, M., Pei, J. 2012. *Data Mining: Concepts And Techniques* Third Edition. San Francisco: Morgan Kaufmann.
- Hastie, T. J., Tibshirani, R. J., Friedman, J.H. 2008. *The Elements of Statistical Learning: Data-mining, Inference and Prediction* Second Edition. New York: Springer-Verlag.
- Liu, B. 2012. *Sentiment Analysis and Opinion Mining*, Morgan & Claypool Publishers.
- Quinlan, J. R. 1993. *C4.5: Programs For Machine Learning*. California: Morgan Kaufmann.
- Salton, C., Buckley, C. 1987. The Weighting Approaches in Automatic Text Retrieval. *Information Processing & Management*, Hal: 513-523.
- Syarif, I., Prugel-Bennett, A., Wills, G. 2016. SVM Parameter Optimization Using Grid Search and Genetic Algorithm to Improve Classification Performance. *Telkomnika (Telecommunication Comput. Electron. Control)* Vol 14, No. 4, Hal: 1502-1509.
- Wahid, D. H., Azhari, S. N. 2016. Peringkasan Sentimen Ekstraktif di Twitter Menggunakan Hybrid TF-IDF dan Cosine Similarity. *IJCCS (Indonesian Journal of Computing and Cybernetics System)* Vol. 10, No.2, Hal: 207-218.