

IMPLEMENTASI ALGORITMA *K-NEAREST NEIGBORS* UNTUK ANALISIS SENTIMEN APLIKASI *JOBSTREET*

Muhammad Fadhil Rizki^{1*}, Wahyu Pramusinto², Mardi Hardjianto³, Subandi⁴

^{1,2,3,4} Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur, DKI Jakarta, Indonesia

Email: ^{1*}fadhilmfriz@gmail.com, ²wahyu.pramusinto@budiluhur.ac.id, ³mardi.hardjianto@budiluhur.ac.id@budiluhur.ac.id, ⁴subandi@budiluhur.ac.id

Abstrak - Jobstreet merupakan perusahaan pihak ketiga yang menjembatani antara pelamar dengan perusahaan yang dilamar. Para pengguna aplikasi JobStreet dapat memberikan review, review tersebut dapat berupa komentar maupun opini. Karena banyaknya *review* yang diberikan pengguna maka membutuhkan beberapa waktu untuk membaca ulasan lengkapnya. Namun jika hanya sedikit komentarnya maka sentimen tersebut akan bias, maka dibutuhkan aplikasi dimana dapat secara otomatis dapat dengan mudah mengidentifikasi sentimen positif maupun negatif. Dari penjelasan permasalahan di atas maka perlu adanya analisis sentimen positif dan negatif untuk melihat sentimen masyarakat pada aplikasi JobStreet. Pada penelitian ini peneliti menggunakan metode *text mining*, analisis sentimen, dan *K-Nearest Neighbors* sebagai algoritmanya, tingkat akurasi pada analisis *k-nearest neighbors* sangatlah efektif jika digunakan pada data latih yang jumlahnya besar dan terdapat informasi kurang atau bahkan tidak berarti. selain itu data yang dianalisis pada penelitian kali ini berjumlah 1000 data dengan pembagian 70:30, 70% data latih dan sisanya untuk data tes. kemudian data tersebut diolah menggunakan *tf-idf* dan *cosine* untuk selanjutnya dilanjutkan ke tahap klasifikasi menggunakan *KNN*. hasil akurasi terbesar adalah 80,54% dengan nilai $k=17$. sedangkan nilai presisi tertinggi sebesar 82,91% saat $k=19$ dan recall terbesar 80,62% ketika $k=17$. untuk meningkatkan nilai akurasi yang lebih baik, maka diperlukan data yang lebih banyak.

Kata Kunci: Analisis Sentimen, *K-Nearest Neighbor*, *Cosine Similarity*, *TF-IDF*, Penambangan Teks.

IMPLEMENTATION OF *K-NEAREST NEIGHBORS* ALGORITHM FOR SENTIMENT ANALYSIS OF APPLICATION *JOBSTREET*

Abstract - *JobStreet* is a third-party company that acts as a bridge between applicants and the company they are applying for. *JobStreet* application users can provide reviews, these reviews can be in the form of comments or opinions. Due to the large number of reviews provided by users, it will take some time to read the full review. However, if there are only a few comments, the sentiment will be biased, so an application is needed which can automatically identify positive and negative sentiments easily. From the explanation of the problems above, it is necessary to analyze positive and negative sentiments to see public sentiment on the *JobStreet* application. In this study the researchers used the Text Mining method, sentiment analysis, and *k-Nearest Neighbors* as the algorithm, the level of accuracy in the *k-Nearest Neighbors* analysis is very effective when used on large amounts of training data and there is less or even meaningless information. In addition, the data analyzed in this study amounted to 1000 data with a division of 70:30, 70% for training data and the rest for test data. Then the data is processed using *TF-IDF* and *Cosine* to then proceed to the classification stage using *K-NN*. The highest accuracy result is 80.54% with a value of $k = 17$. While the highest precision value is 82.91% when $k = 19$ and the greatest recall is 80.62% when $k = 17$. To increase the value of a better accuracy, more data is needed.

Keywords: Sentiment Analysis, *K-Nearest Neighbor*, *Cosine Similarity*, *TF-IDF*, Text Mining.

1. PENDAHULUAN

Salah satu hak yang dimiliki oleh setiap warga negara Indonesia adalah mendapatkan pekerjaan yang layak. Walau demikian kenyataan tidak berbanding lurus dengan ekspektasi pemerintah maupun masyarakat. Banyak masyarakat Indonesia yang masih kesulitan mendapatkan pekerjaan mulai dari jenjang lulusan SD, SMP, SMA sederajat, Diploma, dan bahkan Sarjana masih sulit untuk mendapatkan pekerjaan. Menurut penelitian yang dilakukan oleh [1], pada tahun 2017 angka pengangguran mengalami penurunan dari tahun 2015 yang awalnya pada Februari tahun 2015 jumlah pengangguran sebanyak 5.81 persen dan menurun pada Februari tahun 2017 menjadi 5.33 persen. Walaupun jika melihat data angka pengangguran yang menurun akan tetapi jumlah atau proporsi dari orang yang memiliki jenjang pendidikan yang tinggi seperti lulusan sarjana universitas terus meningkat.

Jika dilihat pada Februari 2014 jumlah angka pengangguran pada masyarakat yang berpendidikan sarjana berada di angka 6,80 persen dan angka ini terus meningkat pada february 2016 naik menjadi 9,90 persen, kemudian angka ini menurun menjadi 8,10 persen pada agustus 2016 dan terakhir pada february 2017 naik lagi menjadi 8,70 persen [2]. Aplikasi Jobstreet merupakan aplikasi dan penyedia lowongan pekerjaan secara *online* yang banyak diunduh di Indonesia. Selain Jobstreet banyak aplikasi penyedia lowongan pekerjaan secara *online* namun

Jobstreet menjadi pilihan utama masyarakat di Indonesia, pasalnya Pada tahun 2010 tepatnya bulan Juli JobStreet sudah digunakan oleh 60.000 perusahaan dan 7.000.000 masyarakat pencari kerja, selain itu JobStreet menempati peringkat 1 di Indonesia dengan presentase 51,4% menggunakannya sebagai aplikasi yang digunakan untuk mencari pekerjaan. Sehingga banyak pengguna telah mengulas aplikasi tersebut di platform Google Play Store.

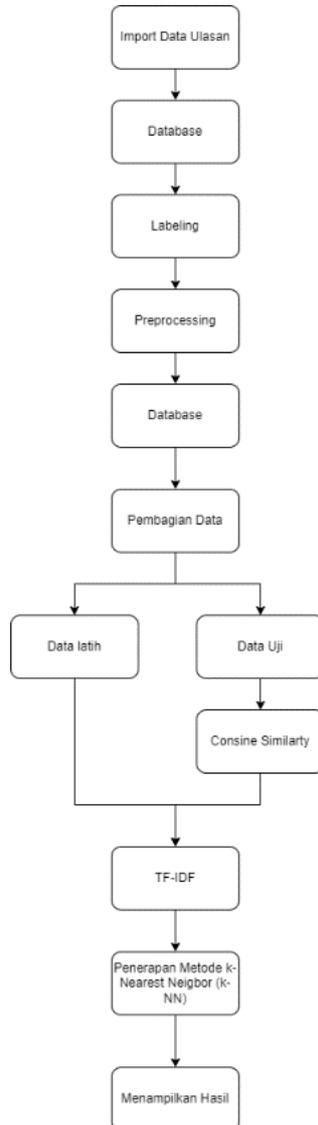
Karena banyaknya sentimen para pengguna dari tahun ke tahun maka perlu waktu yang banyak untuk membaca sentimen lengkapnya. Namun, jika hanya sedikit saja sentimen yang kita lihat maka akan bias. Oleh karena itu kita membutuhkan sebuah aplikasi dimana dapat secara otomatis mengidentifikasi pendapat baik positif atau negatif. Menurut [3] Analisis sentimen merupakan perpaduan antara pengolahan text mining dan data mining, yang fungsinya dapat digunakan untuk mengolah berbagai ulasan atau *sentiment* baik dari konsumen maupun seorang ahli terhadap suatu layanan, organisasi, maupun suatu kegiatan tertentu. Hal ini diperkuat oleh [4] yang mengatakan bahwa analisis sentimen adalah analisis untuk penelitian yang perkembangannya dipengaruhi oleh media sosial, jadi semakin cepat media sosial berkembang maka analisis ini semakin relevan untuk digunakan untuk penelitian.

Menurut [5] Langkah-langkah yang biasanya ada di analisis *sentiment* klasifikasi teks mempunyai empat langkah yakni import data, labeling data, preprocessing, dan menganalisis nya menggunakan suatu metode dan pada penelitian ini menggunakan K-Nearest Neighbor. Pada penelitian [6], hasilnya dibandingkan menggunakan model split-data dengan perbandingan 60:40 Dalam dataset ulasan pengguna aplikasi Bibit dan Bareksa, diperoleh skor presisi, akurasi, dan recall untuk Bibit masing-masing adalah 85.14%, 91.91%, dan 76.44%, sedangkan skor Bareksa adalah 81.70%, 87.15%, 75.73%. Selain itu, penelitian lain oleh [7] meneliti tiga metode, yaitu Naive Bayes, SVM dan KNN, menemukan bahwa di antara ketiga algoritma tersebut, K-Nearest Neighbors (KNN) memiliki kinerja terbaik dan paling akurat. dalam klasifikasi emosional. dalam bahasa Inggris di Halodoc karena memberikan nilai akurasi tertinggi dan nilai AUC tertinggi

Untuk pengolahan data pada penelitian ini, peneliti akan menganalisis bagaimana untuk mengklasifikasikan ulasan yang diberikan oleh pengguna terhadap aplikasi JobStreet, baik itu ulasan positif maupun negatif serta perubahan kata nonformal menjadi formal. Hasil dari pengklasifisian ini adalah berupa text mining atau data berupa text yang kemudian diperlukan metode yang sangat akurat. Dalam penelitian ini Metode yang digunakan untuk mengklasifikasikan adalah K-NN atau K-Nearest Neighbor dan T-IDF atau pembobotan setiap kata. Kemudian untuk pengolahn Text mining sendiri dapat diartikan secara luas sebagai Proses penggalian informasi yang berinteraksi dengan pengguna dalam kumpulan dokumen menggunakan alat analisis yang merupakan bagian dari penambahan data bahwa dia adalah rahasia.

Penambahan teks dapat memberikan solusi untuk masalah seperti pemrosesan, pengurutan/pengelompokan, dan analisis teks tidak terstruktur dalam jumlah besar. Dalam memberikan solusi, Text Mining telah mengadopsi dan mengembangkan banyak teknik dari bidang lain, seperti: penambahan data, pengambilan informasi, statistik dan matematika, pembelajaran mesin, linguistik, pemrosesan dan visualisasi bahasa alami. Kegiatan penelitian text mining meliputi ekstraksi dan penyimpanan teks, pra-pemrosesan konten teks, pengumpulan data statistik, pengindeksan dan analisis konten. Pengklasifikasi yang digunakan dalam penelitian ini adalah K-Nearest Neighbor.

2. METODE PENELITIAN



Gambar 1. Tahapan Alir Metode

2.1 Import Data

Pada tahapan ini data ulasan akan disimpan ke dalam database sebagai raw dataset yang akan digunakan dalam penelitian ini sebagai implemetasi algoritma k-NN untuk analisis sentimen. Data ulasan diambil melalui data ulsan aplikasi JobStreet dalam Google Play Store dengan teknik WEB scraping. Pada tabel 1 adalah sample data yang diambil dari data ulasan aplikasi JobSteert pada Google Play Store:

Tabel 1. Data Ulasan Aplikasi JobSteet Pada Google Play

No	Username	Konten	Score
1	Aldi F.A.	Sangat mudah digunakan dan membantu untuk pencari kerja mencari posisi yang diinginkan.	5
2	Santy Pangaribuan	Mempermudah pencarian kerja	5
3	Rian Dwi S	Mudah dan membantu pencarian kerja	5
4	MR BAE GAMING	Setelah di update aplikasinya malah eror ga jelas,loading terus,bukan mempermudah nyari kerja malah mempersulit	3
5	Msfadhil Fadhil	Ok Kadang error aplikasi harus instal ulang	5

2.2 Labeling Data

Pada tahapan ini data ulasan akan ditandai dengan cara melakukan labelisasi dengan nilai positif atau negatif sebagai acuan perhitungan prediksi dalam tahapan klasifikasi. Atribut labeling didapatkan dari konversi nilai atribut *score* dan *labeling* secara manual. Nilai sentimen yang diambil berdasarkan labeling manual dilakukan dengan cara melihat secara seksama apakah sentimen tersebut masuk ke dalam nilai sentiment positif atau negatif. Sedangkan nilai sentimen yang diambil berdasarkan atribut *score* dilihat berdasarkan nilai *score*-nya. Nilai *score* ≥ 3 merupakan sentiment positif dan *score* < 3 merupakan sentiment negatif. *Labeling* manual bertujuan untuk memperbaiki hasil labelisasi otomatis yang tidak sesuai, dengan melihat secara seksama apakah ulsan tersebut masuk ke dalam sentimen positif atau negatif.

Tabel 2. Proses Labeling Data Ulasan.

No	Konten	Score	Labeling
1	Sangat mudah digunakan dan membantu untuk pencari kerja mencari posisi yang diinginkan.	5	Positif (otomatis)
2	Mempermudah pencarian kerja	5	Positif (otomatis)
3	Mudah dan membantu pencarian kerja	5	Positif (otomatis)
4	Setelah di update aplikasinya malah <i>error</i> ga jelas, <i>loading</i> terus, bukan mempermudah nyari kerja malah mempersulit	3	Negatif (manual)
5	Ok Kadang <i>error</i> aplikasi harus instal ulang	5	Negatif (manual)

2.3 Preprocessing

Menurut [8] *Preprocessing* merupakan tahapan pemrosesan pada analisis sentimen untuk menyiapkan data agar bisa diolah dengan algoritma k-Nearest Neighbors. Hal ini diperkuat oleh [9] yang menyatakan Langkah-langkah yang dilakukan pada tahapan ini adalah *Case Folding*, *Cleaning*, *Lemmatizing*, *Stopwords Removing*, *Tokenization*. Berikut adalah tahapan dalam *Preprocessing*.

a. Case Folding

Case Folding adalah mengubah semua huruf kapital pada teks menjadi huruf non kapital. Tahapan ini dilakukan untuk membuat teks memiliki susunan kata dengan struktur huruf yang sama.

Tabel 3. Proses Case Folding

Kalimat Asli	Hasil Case Folding
Setelah di-update aplikasinya malah <i>error</i> ga jelas, <i>loading</i> terus, bukan mempermudah nyari kerja malah mempersulit	Setelah di-update aplikasinya malah <i>error</i> ga jelas, <i>loading</i> terus, bukan mempermudah nyari kerja malah mempersulit

b. Symbol Removal

Symbol Removal adalah menghapus simbol-simbol pada teks, berfungsi untuk menghapus karakter khusus seperti tanda baca (koma (,), titik (.), tanda tanya (?), tanda seru (!) dan sebagainya), angka numerik (0-9) dan karakter lainnya (\$, %, * dan sebagainya).

Tabel 4. Proses Symbol Removal

Kalimat Asli	Hasil Symbol Removal
Setelah di-update aplikasinya malah eror ga jelas, <i>loading</i> terus, bukan mempermudah nyari kerja malah mempersulit	Setelah di-update aplikasinya malah <i>error</i> ga jelas <i>loading</i> terus bukan mempermudah nyari kerja malah mempersulit

c. Konversi Slangword

Konversi *Slangword* adalah mengubah kata pada teks yang tidak baku kedalam bentuk kata baku. Tahapan ini dilakukan menggunakan bantuan kamus *Slangword*, tahapan ini akan memeriksa setiap kata pada teks apakah terdapat di dalam kamus *slangword* atau tidak. Bila kata tidak baku terdapat dalam kamus *slangword* maka kata tidak baku akan diubah kedalam kata baku yang terdapat di dalam kamus *slangword*.

Tabel 5. Proses Konversi *Slangword*.

Kalimat Asli	Hasil Konversi <i>Slangword</i>
Setelah di-update aplikasinya malah <i>error</i> ga jelas <i>loading</i> terus bukan mempermudah nyari kerja malah mempersulit	setelah di diperbarui aplikasinya bahkan <i>error</i> tidak jelas memuat terus bukan mempermudah mencari kerja bahkan mempersulit

 d. *Stopword Removal*

Stopword Removal adalah tahapan mengambil kata-kata penting dan membuang kata-kata yang kurang penting, pada tahapan ini teks sebelum diklasifikasikan akan dihilangkan dulu teks yang tidak berhubungan dengan analisis sentimen sehingga dimensi teks akan berkurang tanpa mengurangi isi sentimen teks.

Tabel 6. Proses *Stopword Removal*.

Kalimat Asli	Hasil <i>Stopword Removal</i>
Setelah diperbarui aplikasinya bahkan <i>error</i> tidak jelas memuat terus bukan mempermudah mencari kerja bahkan mempersulit	Diperbarui aplikasinya <i>error</i> memuat mempermudah mempersulit

 e. *Stemming*

Stemming adalah suatu proses pengolahan kata-kata untuk mendapatkan kata dasar dari sebuah kata yang telah mengalami imbuhan dengan aturan-aturan tertentu. Proses *Stemming* ini dibantu kamus yang dibuat oleh sastrawi.

Tabel 7. Proses *Stemming*

Kalimat Asli	Hasil <i>Stemming</i>
Diperbarui Aplikasinya <i>Error</i> Memuat Baru aplikasi <i>error</i> muat mudah sulit Mempermudah Mempersulit	Baru aplikasi <i>error</i> muat mudah sulit

f. Tokenisasi

Tokenisasi adalah proses memecah kalimat menjadi satuan kata, proses ini dilakukan dengan melihat setiap spasi pada kalimat maka berdasarkan spasi tersebut kalimat dapat dipecah.

Tabel 8. Proses Tokenisasi.

Kalimat Asli	Hasil Tokenisasi
baru aplikasi <i>error</i> muat mudah sulit	baru,aplikasi, <i>error</i> ,muat,mudah,sulit

2.4 K-Nearest Neighbor

Dari penelitian yang dilakukan oleh [10], metode K-NN adalah model *Supervised Learning* di mana *database* populasi yang digunakan ditugaskan ke kelas tertentu terlebih dahulu. Oleh karena itu, dapat dikatakan bahwa K-Nearest Neighbor merupakan metode yang digunakan dalam proses analisis sentimen untuk menentukan kategori dokumen yang akan diuji dalam proses klasifikasi.

a) TF-IDF

TF-IDF (Term Frequency-Inverse Document Frequency) adalah suatu proses pembobotan tiap kata yang muncul di masing-masing dokumen [11] yang nantinya akan digunakan untuk proses selanjutnya perhitungan cosine similarity. Hal ini diperkuat oleh [12] Pembobotan kata (term) bertujuan untuk memberikan bobot pada setiap kata (term) yang terdapat pada dokumen teks yang akan diproses. Yang menyatakan bahwa Perhitungan dilakukan menggunakan rumus (1), (2), dan (3).

$$W_{tft,d} = \begin{cases} 1 + \log 10t_{ft,d} & \text{if } t_{ft,d} > 0 \\ 0 & \text{if } t_{ft,d} = 0 \end{cases} \quad (1)$$

$$idf_t = \log_{10} \frac{N}{DF} \quad (2)$$

$$W_{t,d} = W_{tft,d} \cdot idf_t \quad (3)$$

Keterangan:

- $W_{ft,d}$ = Bobot kata di setiap dokumen
 $t_{ft,d}$ = Jumlah kemunculan term dalam dokumen
 N = Jumlah keseluruhan dokumen
 D_f = Jumlah dokumen yang mengandung term
 idf = Bobot inverse dalam nilai df
 $W_{t,d}$ = Pembobotan TF-IDF

b) *Cosine Similarity*

Cosine similarity sebagai perhitungan proses jarak dalam metode k-NN. Pada dasarnya metode k-NN membutuhkan perhitungan jarak untuk menentukan kedekatan data uji dengan data tes. Menurut [8], ada beberapa tahapan pada *Cosine similarity* adalah sebagai berikut :

1. Bobot dari tiap-tiap term pada D_i dikalikan dengan setiap term pada semua dokumen data yang ada.
2. Hasil dari perkalian tersebut kemudian dijumlahkan.
3. Kemudian langkah selanjutnya adalah mencari hasil kuadrat dari setiap term pada masing-masing dokumen (termasuk D_i) Lalu setelah itu dijumlahkan kemudian diakarkan.
4. Lakukan pembagian antara hasil dari langkah nomor 2 dengan langkah nomor 3. Maka, didapatkan nilai *Cosine Similarity*.

3. HASIL DAN PEMBAHASAN

Pada bagian ini berisi analisis, hasil implementasi ataupun pengujian serta pembahasan dari topik penelitian, yang bisa dibuat terlebih dahulu metodologi penelitian. Bagian ini juga merepresentasikan penjelasan yang berupa penjelasan, gambar, tabel dan lainnya.

Tabel 9. Data Ulasan Aplikasi JobSteet Pada Google Play

No	Konten	Sentimen
1	Sangat mudah digunakan dan membantu untuk pencari kerja mencari posisi yang diinginkan.	?? (akan dicari)
2	Mempermudah pencarian kerja	Positif
3	Mudah dan membantu pencarian kerja	Positif
4	Setelah di update aplikasinya malah eror ga jelas,loading terus,bukan mempermudah nyari kerja malah mempersulit	Negatif
5	Ok Kadang error aplikasi harus instal ulang	Negatif

Tabel 9 merupakan data penelitian diambil atribut-atribut yang penting yaitu konten dan score. Kemudian atribut sentimen didapatkan dari konversi nilai atribut score dan labeling secara manual. Nilai sentimen yang diambil berdasarkan labeling manual dilakukan dengan cara melihat secara seksama apakah sentimen tersebut masuk ke dalam nilai sentiment positif atau negatif. Sedangkan nilai *sentiment* yang diambil berdasarkan atribut score dilihat berdasarkan nilai scorenya. Nilai *score* ≥ 3 merupakan sentiment positif dan *score* < 3 merupakan sentiment negatif.

Tabel 10. Hasil Tahapan Preprocessing

No	Konten	Hasil
1	Sangat mudah digunakan dan membantu untuk pencari kerja mencari posisi yang diinginkan.	mudah,cari
2	Mempermudah pencarian kerja	mudah,cari
3	Mudah dan membantu pencarian kerja	mudah,cari
4	Setelah di update aplikasinya malah eror ga jelas,loading terus,bukan mempermudah nyari kerja malah mempersulit	baru,aplikasi,eror,muat,mudah,sulit
5	Ok Kadang error aplikasi harus instal ulang	oke,kadang,eror,aplikasi,instal,ulang

Tabel 10 merupakan hasil dari tahapan *preprocessing*, preprocess bertujuan untuk membersihkan dataset agar dataset siap digunakan dalam klasifikasi.

3.1 TFIDF

TF-IDF (*Term Frequency-Inverse Document Frequency*) merupakan langkah dalam menghitung bobot kata (term) dari suatu data. Frekuensi istilah menghitung jumlah kemunculan suatu istilah dalam dokumen. di sisi lain, document frequency adalah jumlah dokumen di mana istilah tersebut muncul. Cara mendapatkan TF-IDF dapat dilihat pada rumus (1), (2), dan (3). Langkah-langkah dalam perhitungan TF-IDF dapat dilihat pada poin berikut ini.

a. TF (*Term Frequency*) dan IDF (*Document Frequency*)

Pada rumus (1) menyatakan nilai term frequency dan rumus (2) menyatakan nilai IDF. Sebagai contoh, ada 5 data ulasan. Akan dicari bobot kata “mudah” pada dokumen kedua. Kata lowongan muncul sebanyak satu kali pada dokumen kedua. Dari 5 dokumen tersebut, kata mudah terulang pada empat dokumen. Maka hasil perhitungannya adalah :

$$\begin{aligned}
 W_{tft,d} &= 1 \\
 Df &= 4 \\
 N &= 5 \\
 idf &= W_{tft,d} * \left(\log_{10} \frac{N}{DF} \right) = 1 * \left(\log_{10} \frac{5}{4} \right) \\
 &= 0.097
 \end{aligned}$$

Hasil seluruh perhitungan TF dan IDF dapat dilihat pada Tabel 3.

Tabel 11. Hasil Perhitungan TF dan IDF

Term	TF					DF	IDF
	D1	D2	D3	D4	D5		
mudah	1	1	1	1		4	0.097
cari	1	1	1			3	0.222
baru				1		1	0.699
aplikasi				1	1	2	0.398
eror				1	1	2	0.398
muat				1		1	0.699
sulit				1		1	0.699
oke					1	1	0.699
kadang					1	1	0.699
instal					1	1	0.699
ulang					1	1	0.699

b. Pembobotan TF-IDF

Pembobotan TF-IDF merupakan perkalian antara nilai TF dan IDF yang sudah dicari pada poin sebelumnya. Cara perhitungan pembobotan TF-IDF dapat dilihat pada rumus (3). Hasil perhitungan pembobotan TF-IDF dapat dilihat pada Tabel 4.

Tabel 12. Hasil Perhitungan Pembobotan TF-IDF

Term	$W_{t,d} = W_{tft,d} * idf_t$				
	D1	D2	D3	D4	D5
mudah	0.097	0.097	0.097	0.097	0
cari	0.222	0.222	0.222	0	0
baru	0	0	0	0.699	0
aplikasi	0	0	0	0.398	0.398
eror	0	0	0	0.398	0.398
muat	0	0	0	0.699	0
sulit	0	0	0	0.699	0
oke	0	0	0	0	0.699
kadang	0	0	0	0	0.699
instal	0	0	0	0	0.699
ulang	0	0	0	0	0.699

c. Hasil *Cosine Similarity*

Untuk menentukan hasil akhir dari *cosine similarity* yaitu dengan cara melakukan pembagian antara hasil dari perhitungan pembobotan kemiripan dokumen dengan langkah perhitungan panjang vektor. Maka, akan didapatkan nilai *Cosine Similarity* pada Tabel 7.

Tabel 15. Hasil *Cosine Similarity*

$\text{Cos}(D_1, D_2)$	$\text{Cos}(D_1, D_3)$	$\text{Cos}(D_1, D_4)$	$\text{Cos}(D_1, D_5)$
0,999	0,999	0,028	0

Tabel 16. Tabel Confusion Matrix

Nilai K	Confusion Matrix		
	Akurasi	Presisi	Recall
3	80.20%	82.19%	80.20%
5	79.19%	81.00%	79.19%
7	79.19%	81.00%	79.19%
9	79.19%	81.23%	79.19%
11	79.53%	81.92%	79.62%
13	80.20%	82.16%	80.28%
15	79.87%	82.19%	79.87%
17	80.54%	82.63%	80.62%
19	80.20%	82.91%	80.30%
21	78.86%	82.01%	78.96%
23	79.19%	81.95%	79.29%
25	78.86%	81.72%	78.96%
27	79.19%	82.24%	79.30%
29	78.52%	81.50%	81.50%
31	78.19%	80.99%	78.29%
33	78.19%	81.57%	81.57%
35	78.52%	78.52%	78.63%
37	78.19%	81.57%	78.30%
39	77.85%	81.35%	77.96%
41	77.85%	81.68%	77.97%

Dari hasil tersebut dapat diurutkan nilai similarity D1 dari yang tertinggi ke yang terendah yaitu :

1. D2 (Positif)
2. D3 (Positif)
3. D4 (Negatif)
4. D5 (Negatif)

3.3 KNN

Untuk menentukan prediksi sentimen menggunakan metode k-NN (k-Nearest Neighbor) dapat melihat hasil dari perhitungan cosine similarity yang sudah diurutkan pada poin sebelumnya. Jika nilai k yang ditetapkan untuk k-NN adalah 3 maka akan dipilih 3 nilai tertinggi dari hasil *cosine similarity*. Dari 3 nilai tertinggi tersebut, kelas sentimen yang paling banyak muncul adalah positif sehingga D1 akan diklasifikasikan ke dalam sentimen positif.

4. KESIMPULAN

Berdasarkan analisis dan pengujian yang sudah dilakukan maka kesimpulan penelitian ini adalah algoritma k-NN mampu melakukan klasifikasi ulasan secara otomatis, hasil pengujian yang didapatkan dengan menggunakan total kurang lebih 1000 data sampel dengan pembagian 70% *dataset* dan 30% data *test* mendapatkan rata-rata nilai akurasi sebesar 79.08%, nilai rata-rata presisi sebesar 81,60%, nilai rata-rata presisi sebesar 79,40%. Dengan hasil pengujian tersebut bahwa algoritma k-NN cukup baik jika diimplementasikan dalam analisis sentimen. Proses klasifikasi semakin akurat jika *dataset* yang digunakan dalam penelitian berjumlah banyak. Nilai akurasi, presisi, dan *recall* bisa lebih ditingkatkan jika data sampel yang digunakan lebih banyak.

DAFTAR PUSTAKA

- [1] A. M. Nurendra and W. Purnamasari, "Hubungan Antara Kualitas Kehidupan Kerja Danketerikatan Kerjapada Pekerja Wanita," *Indigenous : Jurnal Ilmiah Psikologi*, vol. II, pp. 148-154, 2017.
- [2] G. P. Yuhana, Penggunaan Media Online Pencari Kerja Jobstreet Dan Pemenuhan Kebutuhan Informasi Lowongan Kerja, Skripsi ed., Surakarta: Universitas Sebelas Maret, 2019.
- [3] N. T. Romadloni, I. Santoso and S. Budilaksono, "Perbandingan Metode Naive Bayes, Knn Dan Decision Tree Terhadap Analisis Sentimen Transportasi KRL Commuter Line," *Jurnal IKRA-ITH Informatika*, vol. III, pp. 1-9, 2019.
- [4] N. D. Mentari, M. A. Fauzi and L. Muflikhah, "Analisis Sentimen Kurikulum 2013 Pada Sosial Media Twitter Menggunakan Metode K-Nearest Neighbor dan Feature Selection Query Expansion Ranking," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. II, pp. 2739-2743, 2018.
- [5] S. Ernawati and R. Wati, "Penerapan Algoritma K-Nearest Neighbors Pada Analisis Sentimen Review Agen Travel," *LPPM Universitas Bina Sarana Informatika*, vol. VI, pp. 64-69, 2018.
- [6] A. D. A. Putra and S. Juanita, "Analisis Sentimen Pada Ulasan Pengguna Aplikasi Bibit Dan Bareksa Dengan Algoritma KNN," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. VIII, pp. 636-646, 2021.
- [7] E. Indrayuni, A. Nurhadi and D. A. Kristiyanti, "Implementasi Algoritma Naive Bayes, Support Vector Machine, dan K-Nearest Neighbors Untuk Analisa Sentimen Aplikasi Halodoc," *Faktor Exacta*, vol. XIV, pp. 64-71, 2021.
- [8] A. Deviyanto and M. D. R. Wahyudi, "Penerapan Analisis Sentimen Pada Pengguna Twitter Menggunakan Metode K-Nearest Neighbor," *JISKa (Jurnal Informatika Sunan Kalijaga)*, vol. III, pp. 1-13, 2018.
- [9] M. R. A. Nasution and M. Hayaty, "Perbandingan Akurasi dan Waktu Proses Algoritma K-NN dan SVM dalam Analisis Sentimen Twitter," *Jurnal Informatika BSI*, vol. VI, pp. 212-218, 2019.
- [10] R. Sari, "Analisis Sentimen Pada Review Objek Wisata Dunia Fantasi menggunakan Algoritma K-Nearest Neighbor (K-NN)," *Evolusi: Jurnal Sains dan Manajemen*, vol. VIII, pp. 10-17, 2020.
- [11] M. R. Fahlevvi, "Analisis Sentimen Terhadap Ulasan Aplikasi Pejabat Pengelola Informasi dan Dokumentasi Kementerian Dalam Negeri Republik Indonesia di Google Playstore Menggunakan Metode Support Vector Machine," *Institut Pemerintahan Dalam Negeri*, vol. IV, pp. 1-13, 2022.
- [12] W. E. Nurjanah, R. S. Perdana and M. A. Fauzi, "Analisis Sentimen Terhadap Tayangan Televisi Berdasarkan Opini Masyarakat pada Media Sosial Twitter menggunakan Metode K-Nearest Neighbor dan Pembobotan Jumlah Retweet," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. I, pp. 1750-1757, 2017.