

## Analisis Sentimen Pada Ulasan “Lazada” Berbahasa Indonesia Menggunakan *K-Nearest Neighbor* (K-NN) Dengan Perbaikan Kata Menggunakan *Jaro Winkler Distance*

Yane Marita Febrianti<sup>1</sup>, Indriati<sup>2</sup>, Agus Wahyu Widodo<sup>3</sup>

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya  
Email: <sup>1</sup>febriantiyane28@gmail.com, <sup>2</sup>indriati.tif@ub.ac.id, <sup>3</sup>a\_wahyu\_w@ub.ac.id

### Abstrak

Perkembangan sebuah teknologi informasi saat ini membawa dampak yang cukup besar terhadap pola hidup masyarakat salah satunya pada daya beli. Saat ini daya beli masyarakat lebih cenderung berbelanja secara *online* karena dianggap lebih mudah. Namun, bagaimana konsumen mengetahui jika barang yang akan dibeli bagus atau sebaliknya, oleh karena itu muncul adanya suatu ulasan atau komentar pada setiap barang yang dijual. Ulasan pada suatu barang membawa pengaruh yang cukup besar terhadap daya beli konsumen untuk mengetahui kualitas barang tersebut, tidak heran jika sebuah ulasan menjadi salah satu tujuan utama yang dilihat oleh konsumen setelah harga, tetapi tidak semua ulasan yang diberikan konsumen dapat dimengerti oleh konsumen lain dikarenakan penggunaan kata yang disingkat, penggunaan bahasa modern, salah dalam mengetik huruf, dan tidak bakunya kata yang digunakan dalam penulisan ulasan. Dengan latar belakang diatas, peneliti mengusulkan pembuatan sistem Analisis Sentimen Pada Ulasan “Lazada” Berbahasa Indonesia Menggunakan *K-Nearest Neighbor* (K-NN) Dengan Perbaikan Kata Menggunakan *Jaro Winkler Distance*. Pengujian berdasarkan nilai *precision*, *recall*, dan *accuracy* pada masing-masing analisis sentimen tanpa perbaikan kata, maupun dengan perbaikan kata. Hasil pengujian dengan nilai *accuracy* yang baik terdapat pada analisis sentimen dengan perbaikan kata yakni 76 %, dengan nilai *precision* 0,76, dengan nilai *recall* 1.

**Kata kunci:** *analysis sentimen, perbaikan kata, k-nearest neighbor (K-NN), jaro winkler distance, accuracy, precision, recall*

### Abstract

The development of an information technology currently carries a considerable impact against the pattern of life one on purchasing power. The current purchasing power are more likely to shop online because it's considered easier. But, how does a consumer know if the items to be purchased good or otherwise. Therefore it appears there is a review or comment on any goods sold. Review on items bring considerable influence against the purchasing power of consumers to know the quality of the goods, does not be surprised if a review into one of the main goals being viewed by consumers after the price. However, not all reviews provided the consumers can be understood by other consumers due to use the word is abbreviated, it use modern languages, in typing letters, the researcher proposes the creation of a system Analysis of Sentiment on the Reviews “Lazada” Berbahasa Indonesia Using the *K-Nearest Neighbor* (K-NN) and Repair Word Using *Jaro Winkler Distance*. Testing based on the value of *precision*, *recall*, and *accuracy* at each analysis sentiment without repair word, or with repair word. The test result with good accuracy value is present on the analysis sentiment with repair word is 76 %, with value of *precision* 0,76, and *recall* 1.

**Keywords:** *analysis of sentiment, repair word, k-nearest neighbor (K-NN), jaro winkler distance, accuracy, precision, recall*

## 1. PENDAHULUAN

Sebuah teknologi yang saat ini berkembang sangat pesat yakni internet membawa dampak

yang sangat besar dalam masyarakat, salah satunya adalah dalam kecenderungan berbelanja secara *online* pada sebuah toko yang memang dikhususkan untuk menjual secara *online* seperti

"Lazada". Namun bagaimana konsumen mengetahui jika barang yang akan dibeli tersebut merupakan barang dengan kualitas yang bagus, maka dari permasalahan tersebut muncul adanya sebuah ulasan atau komentar dari konsumen pada masing-masing barang yang dijual yang bertujuan untuk membantu konsumen mengetahui kualitas barang yang akan dibeli. Ulasan dari konsumen di toko *online* seperti "Lazada" memang membawa pengaruh yang cukup besar terhadap daya beli konsumen berikutnya pada barang yang sama, dan juga merupakan suatu ukuran dimana konsumen lain akan melihat baik buruknya barang yang dijual tersebut berdasarkan ulasan dari pembeli sebelumnya. Tidak heran jika sebuah ulasan menjadi salah satu tujuan utama yang dilihat oleh konsumen setelah harga. Namun tidak semua ulasan yang diberikan konsumen dapat dimengerti oleh konsumen lain dikarenakan penggunaan kata yang seharusnya baku menjadi disingkat, kemudian penggunaan berbagai bahasa yang saat ini sedang *trend*, beberapa juga ada yang mungkin salah ketika mengetik beberapa huruf yang jarak antar huruf tersebut berdekatan, atau terdapat kata tidak baku yang digunakan dalam penulisan ulasan. Dari beberapa kesalahan dalam penulisan ulasan terdapat kesalahan yang disengaja maupun tidak sengaja atau yang biasa disebut *typo*. Pada kesalahan yang disengaja terdapat beberapa kesalahan yang memang konsisten dan dilakukan seseorang secara terus-menerus dalam menuliskan suatu ulasan seperti contoh kata "bagus disingkat menjadi bgs", sehingga dibutuhkan sebuah perbaikan kata untuk kesalahan disengaja maupun tidak disengaja untuk mendapatkan nilai analisis sentimen dengan *accuracy* yang baik.

Dari permasalahan diatas, analisis sentimen memiliki peran yang sangat penting yakni untuk mengekstraksi polaritas ulasan yang diberikan antar kelas (positif dan negatif) dari ulasan teks, oleh karena itu dibutuhkan metode klasifikasi yang digunakan untuk mengekstraksi ulasan tersebut (Antinasari, Perdana, & Fauzi, 2017). Pengetahuan dan ilmu-ilmu yang telah dipelajari dalam *Text Mining* harus menjadi dasar dalam menyelesaikan sistem ini. Ada beberapa macam metode yang digunakan untuk klasifikasi teks yakni menggunakan metode yang umum digunakan yakni *Naive Bayes*, *K-Nearest Neighbor* (K-NN), *Minimum Edit Distance*, dan masih banyak lagi metode klasifikasi yang lain. Sedangkan metode yang digunakan dalam

melakukan perbaikan kata diantaranya metode *TF-IDF*, *vektor space*, *Search Matching*, ada juga metode lain yang ada pada *Aproximate String Matching* yakni yang pertama ada *Levenshtein Distance*, kedua *Hamming Distance*, ketiga *Damerau Levenshtein Distance*, dan yang keempat adalah *Jaro Winkler Distance*.

Terdapat penelitian untuk klasifikasi teks menggunakan *K-Nearest Neighbor* (K-NN) dengan alasan menggunakan penelitian K-NN menunjukkan *accuracy* yang cukup baik, dengan memberikan nilai  $k=3$ , K-NN dapat memberikan hasil persentase yang cukup baik yakni 88,29% (Samuel, Delima, & Rachmat, 2014). Selain itu, terdapat penelitian untuk membandingkan algoritme perbaikan kata dengan hasil penelitian yang menyebutkan jika algoritme *Jaro Winkler Distance* mampu memberikan nilai tertinggi dibanding algoritme pada *Aproximate String Matching* yang lainnya. Nilai yang diperoleh *Jaro Winkler Distance* yakni MAP 0,87 yang merupakan nilai terbesar dari ketiga algoritme yang lain, dimana nilai MAP terbagi menjadi empat kesalahan penulisan yaitu yang pertama kesalahan dalam penghapusan huruf dengan total nilai 0,92, kedua kesalahan penambahan huruf dengan total nilai 0,90, ketiga kesalahan penggantian huruf dengan total nilai 0,70, dan yang terakhir kesalahan penukaran huruf dengan total nilai 0,95 (Rocmawati & Kusumaningrum, 2015). Selanjutnya, terdapat penelitian yang juga menggunakan algoritme *Jaro Winkler Distance* yang digunakan untuk membandingkan kesamaan dokumen berbahasa Indonesia dengan mendapatkan kesimpulan bahwa aplikasi akan berjalan dengan baik pada dokumen jika dokumen tersebut memiliki kemiripan dan urutan kata yang sama (Kurniawati, Pupitodjati, & Rahman, 2016).

Berdasarkan permasalahan dan alasan yang telah dipaparkan diatas, dan didukung oleh beberapa penelitian yang sudah dilakukan, maka pada penelitian ini akan dibangun sebuah sistem untuk melakukan analisis sentimen pada ulasan menggunakan metode *K-Nearest Neighbor* (K-NN) sebagai metode klasifikasi teks, kemudian hasil klasifikasi tersebut akan dibandingkan dengan hasil klasifikasi yang sudah melalui perbaikan kata menggunakan algoritme *Jaro Winkler Distance*. Dengan demikian sistem yang akan dibangun ini diharapkan dapat membantu seoptimal mungkin terhadap permasalahan untuk analisis sentimen khususnya pada ulasan dengan kata yang tidak baku dengan melalui

perbaikan kata terlebih dahulu.

## 2. METODOLOGI PENELITIAN

### 2.1 Ulasan/Komentar

Merupakan suatu teks atau kalimat yang berisi penilaian atau komentar terhadap sesuatu hasil karya seseorang. Didalam sebuah ulasan biasanya terdapat berbagai macam pengungkapan seperti pujian, pertanyaan, tafsiran, maupun komentar-komentar yang lain. Pentingnya ulasan dalam produk karena sebagian besar dari konsumen *online* akan melihat terlebih dahulu ulasan yang diberikan konsumen terlebih dahulu sebelum membeli produk tersebut (Nanda, 2015).

### 2.2 Kesalahan Penulisan

Kesalahan penulisan merupakan suatu *human error* (kesalahan manusia) yang cukup sering dilakukan oleh seseorang dalam melakukan sebuah penulisan. Kesalahan penulisan sendiri dibagi menjadi dua yakni kesalahan penulisan yang dilakukan secara sengaja dan tidak sengaja. Untuk kesalahan penulisan yang tidak disengaja atau biasa disebut dengan kesalahan tipografi (*typographical error*), sedangkan untuk kesalahan penulisan yang dilakukan dengan sengaja karena kesalahan tersebut bersifat konsisten yang dilakukan secara berulang-ulang seperti kata “barang disingkat brg”, “bagus disingkat bgs”, serta beberapa kata lain yang sering dituliskan secara tidak baku (Fahma, Cholissodin, & Perdana, 2018).

### 2.3 Text Mining

Populer dengan sebutan Pemrosesan Teks yang saat ini lebih dikenal dengan *Text Mining* merupakan satu dari banyak pengetahuan di bidang *Artificial Intelligence* dimana pada *Text Mining* ini menerapkan sebuah konsep yang disertai dengan teknik yang ada pada data mining yang digunakan dalam pencarian pola teks yang menggunakan proses ekstraksi terlebih dulu untuk mendapatkan informasi dari pola data teks yang tidak terstruktur. Pada penggunaan *Text Mining* teknik yang digunakan yakni melakukan pengambilan data berupa teks yang diambil dari beberapa sumber seperti dokumen, dengan tujuan untuk mencari inti dari dokumen tersebut sehingga dapat dilakukan proses analisis mengenai hubungan dan kemiripan dokumen satu dengan yang lain (Baskoro, Ahmad, &

Furqon, 2015).

### 2.4 Information Retrieval (IR)

*Information Retrieval (IR)* atau biasa disebut Sistem Temu Kembali Informasi adalah pengetahuan atau sistem yang digunakan dalam menemukan kembali informasi yang diminta pengguna secara relevan sesuai dengan permintaan informasi yang dibutuhkan. Aplikasi yang sudah sering digunakan adalah mesin pencarian atau *search engine* dimana sistem harus terhubung dengan internet. Pada sistem temu kembali informasi ini tidak semata-mata hanya untuk menemukan dokumen dengan *query* yang diberikan pengguna, namun sistem juga harus memperhatikan ukuran keefektifan suatu dokumen dengan menghitung *precision* dan *recall*. *Precision* merupakan jumlah dokumen relevan yang diperoleh dengan jumlah seluruh dokumen yang diperoleh mesin pencari. Sedangkan *Recall* merupakan penghitungan yang lebih sempit lagi karena dihitung dari jumlah dokumen relevan setelah dokumen yang relevan tersebut diperoleh (Fahma, Cholissodin, & Perdana, 2018).

### 2.5 Natural Language Processing (NLP)

*Natural language Processing (NLP)* merupakan penerapan ilmu linguistik pada ilmu komputer atau *computational linguistics* yang berkonsentrasi dalam menjembatani interaksi antara komputer dengan bahasa (alami) manusia. NLP berusaha memecahkan masalah dengan berbagai aturan gramatika serta semantik dan merubah bahan menjadi representasi formal yang kemudian diproses komputer. Tantangan-tantangan dalam penerapan NLP untuk memahami bahasa manusia, antara lain (Pustejovsky & Stubbs, 2012):

1. Penandaan kelas kata (*part-of-Speech tagging*). Kesulitan dalam proses penandaan pada kelas (kata benda, kata kerja, kata sifat dll) karena kata pada teks bergantung pada konteks penggunaannya.
2. Segmentasi teks (*text segmentation*) segmentasi sulit direpresentasikan pada bahasa yang tak memiliki pembatas kata spesifik contohnya : Bahasa Mandarin, Jepang, dan Thailand.
3. Disambiguasi makna kata (*word sense disambiguation*). Beberapa kata memiliki makna lebih dari satu makna baik dalam bentuk homonim, polisemi.

4. Ambiguitas Sintaksis (*Syntactic Ambiguity*). Setiap bahasa memiliki banyak kemungkinan struktur kalimat. Untuk membuat struktur kalimat yang tepat perlu gabungan informasi secara semantik maupun kontekstual.
5. Masukan yang tidak sempurna atau tidak teratur. Sering terjadi kesalahan pada ejaan, aksan bahasa, dan gramatikal dalam bahasa tulis menyulitkan pemrosesan bahasa alami.
6. Pertuturan (*speech act*). Gaya bahasa serta konteks menentukan maksud yang diinginkan. Karena kadang kalimat tidak menggambarkan maksud penulis.

## 2.6 Term Frequency-Inverse Frequency (TF-IDF Document)

Merupakan statistik numerik yang mencerminkan betapa pentingnya sebuah kata dalam sebuah dokumen dalam koleksi atau *corpus*. TF merupakan frekuensi *term* pada suatu dokumen, *term* adalah sebuah kata atau frase. Frekuensi merupakan atribut penting *term* untuk membedakan dirinya dari *term* lain (Xia & Chai, 2011). Artinya, nilai TF dari *term*  $i$  adalah:

$$TF_i = tf_{ij} \quad (1)$$

$tf_{ij}$  menunjukkan frekuensi *term*  $i$  dalam dokumen  $j$ . Karena jumlah frekuensi *term* mungkin sangat besar, rumus berikut ini juga sering digunakan untuk menghitung nilai TF (Xia & Chai, 2011).

$$TF_i = \log_2(tf_{ij}) \quad (2)$$

Sedangkan IDF pada dasarnya mengukur jumlah informasi yang diberikan oleh sebuah kata, yaitu apakah *term* itu biasa atau jarang terjadi di semua dokumen. Formula dasar diberikan oleh (Robertson, 2004). Sehingga menghasilkan formula IDF berikut ini:

$$IDF_i = \log_2\left(\frac{N}{n_j}\right) + 1 = \log_2(N) - \log_2(n_j) + 1 \quad (3)$$

Dimana  $N$  adalah jumlah total dokumen dalam koleksi dan  $n_j$  adalah jumlah dokumen yang mengandung setidaknya satu kejadian dari *term*  $i$  (Xia & Chai, 2011). Tujuan penghitungan IDF adalah untuk mencari kata-kata yang benar-benar merepresentasikan suatu dokumen teks pada suatu koleksi. Metode pembobotan kata yang digunakan dalam penelitian ini adalah metode TF-IDF. Metode ini digunakan karena metode ini paling baik dalam perolehan

informasi. Perhitungan bobot yaitu, frekuensi kemunculan sebuah kata di dalam sebuah dokumen tertentu yang disebut *Term Frequency* (TF) dan inverse frekuensi dokumen yang mengandung kata yang disebut *Inverse Document Frequency* (IDF). Frekuensi kemunculan kata di dalam dokumen yang diberikan menunjukkan seberapa penting kata tersebut di dalam dokumen. Sehingga bobot hubungan antara sebuah kata dan sebuah dokumen akan tinggi apabila frekuensi kata tinggi di dalam dokumen dan frekuensi keseluruhan dokumen yang mengandung kata tersebut akan rendah pada kumpulan dokumen. Kemudian TF-IDF dihitung dengan cara:

$$TF - IDF = TF \times IDF \quad (4)$$

Dengan konvensi, nilai TF-IDF meningkat secara proporsional dengan sebuah kata berapa kali muncul dalam sebuah dokumen, namun diimbangi oleh frekuensi kata di dalam korpus, yang membantu mengendalikan fakta bahwa beberapa kata lebih umum daripada yang lain (Christian, Agus, & Suhartono, 2016).

## 2.7 Cosine Similarity

*Cosine Similarity* atau biasa disebut dengan ukuran kesamaan yang umum digunakan dalam mengukur sudut antara vektor dokumen dengan vektor *query*. Setiap vektor merepresentasikan setiap kata dalam setiap dokumen. Perhitungan yang dihasilkan dari *Cosine Similarity* berupa tingkat kesamaan antara dokumen dengan *query* yang diinputkan apakah sesuai atau tidak. Semakin tinggi nilai *cosine similarity* maka tingkat kemiripan dokumen dan *query* semakin baik. Sehingga didapatkan rumus yang digunakan untuk *cosine similarity* adalah sebagai berikut (Fahma, Cholissodin, & Perdana, 2018):

$$\text{CosSim}(d_j, q) = \vec{d_j} \cdot \vec{q} = \sum_{i=1}^t (w_{ij} \cdot w_{iq}) \quad (5)$$

## 2.8 Metode K-Nearest Neighbor (K-NN)

Untuk proses pemecahan masalah klasifikasi metode *K-Nearest Neighbor* (K-NN) inilah yang menjadi metode paling sederhana diantara metode klasifikasi lainnya. Teknik yang digunakan pada K-NN ini yakni dengan mengklasifikasikan data dengan obyek yang memiliki nilai tetangga yang paling dekat. hasil yang didapatkan dari proses K-NN ini akan menjadi lebih tinggi atau yang paling optimal ketika dalam perhitungan setiap *term* digunakan pembobotan *cosine similarity* pada masing-



masing dokumen yang akan diklasifikasikan (Nurjanah, Pradana, & Fauzi, 2017):

1. *Term Frequency* (TF)

Merupakan frekuensi kemunculan kata pada suatu dokumen teks. *Term Frequency* (tft,d) didefinisikan jumlah kemunculan *term* t pada dokumen d.

2. *Document Frequency* (DF)

Merupakan kata-kata yang banyak terdapat pada dokumen, kata tersebut tidak informatif, seperti kata dan, di, atau, bisa, merupakan.

3. *Invers Document Frequency* (IDF)

Merupakan frekuensi kemunculan *term* pada keseluruhan dokumen teks. *Term* yang jarang muncul pada keseluruhan dokumen teks memiliki nilai *Invers Document Frequency* lebih besar dibandingkan dengan *term* yang sering muncul.

4. *Term Frequency - Invers Document Frequency* (TF-IDF)

Nilai tf-idf dari sebuah kata merupakan kombinasi dari nilai tf dan nilai idf dalam perhitungan bobot.

## 2.9 Metode Jaro Winkler Distance

Merupakan algoritme yang memiliki nilai tertinggi dari algoritme lainnya, meskipun sama-sama digunakan untuk mengukur kesamaan pada dua string. Jika nilai *Jaro Winkler Distance* semakin tinggi atau mendekati nilai satu maka string tersebut semakin mirip dengan string acuan (Kurniawati, Pupitodjati, & Rahman, 2016). Algoritme ini memiliki tiga bagian penyelesaian yaitu : (1) Menghitung panjang string, (2) Menemukan jumlah karakter yang sama didalam dua string, (3) Menemukan jumlah transposisi.

1. Menghitung panjang string
2. Menemukan jumlah karakter yang sama di dalam dua string

$$dj = \frac{1}{3} \times \left( \frac{m}{|s1|} + \frac{m}{|s2|} + \frac{m-t}{m} \right) \quad (6)$$

Ket :

m = jumlah karakter yang sama persis

|s1| = panjang string 1

|s2| = panjang string 2

t = jumlah transportasi

Jarak teoritis dua buah karakter :

$$\left( \frac{\max(|s1|, |s2|)}{s} \right) < -1 \quad (7)$$

3. Menemukan jumlah transposisi

$$dw = dj + (lp(1 - dj)) \quad (8)$$

Ket :

dj = Jaro distance untuk string s<sub>1</sub> dan s<sub>2</sub>

l = panjang prefiks umum di awal string nilai maksimalnya 4 karakter.

P = konstanta scaling faktor

## 2.10 Precision dan Recall

Pengukuran suatu efektifitas dari *information retrieval* dapat dilakukan dengan melakukan perhitungan nilai ketepatan (*precision*), dan nilai perolehan (*recall*). *Precision* dapat diartikan sebagai kecocokan antara permintaan informasi dengan query yang diminta, ketika seseorang mencari informasi di sebuah sistem, dan sistem menampilkan beberapa dokumen, maka kecocokan dokumen tersebut disebut relevan. Sedangkan *Recall* adalah proporsi jumlah dokumen yang dapat ditemukan kembali oleh sebuah proses pencarian di sistem *Information Retrieval* (Lestari, 2015).

Rasio ketepatan (*precision*) adalah perbandingan antara dokumen relevan dengan jumlah dokumen yang ditemu balik dalam penelusuran. Ketepatan (*precision*) berkaitan dengan kemampuan sistem untuk tidak memanggil dokumen yang tidak relevan. Untuk menghitung nilai ketepatan (*precision*) digunakan rumus sebagai berikut (Hasugian, 2006) :

$$P = \frac{\text{Jumlah dokumen relevan yang terambil}}{\text{Jumlah dokumen terambil dalam pencarian}} \quad (9)$$

Rasio perolehan (*recall*) adalah perbandingan dokumen ditemukan dengan jumlah total dokumen relevan dalam sistem. Perolehan (*recall*) berhubungan dengan kemampuan sistem untuk memanggil dokumen yang relevan. Untuk menghitung nilai perolehan (*recall*) digunakan rumus sebagai berikut (Hasugian, 2006):

$$R = \frac{\text{Jumlah dokumen relevan yang terambil}}{\text{Jumlah dokumen relevan pada database}} \quad (2.10)$$

### 3. IMPLEMENTASI

Alur penyelesaian masalah ini ada proses klasifikasi data *input*, dan proses perbaikan kata. Pada proses ini menjelaskan gambaran dari proses klasifikasi berdasarkan data uji yang diberikan, yang hasil tersebut dihitung melalui nilai *cosine similarity* yang diambil sejumlah *k*. Proses klasifikasi dilakukan dengan menggunakan metode *K-Nearest Neighbor*, dimana data *input* harus melalui proses *preprocessing* dan pembobotan kata terlebih dahulu sebelum dilakukan perhitungan nilai *cosinenya*. Proses klasifikasi akan diulang kembali, namun pada proses *preprocessing* dilakukan perbaikan kata yang dianggap ambigu atau tidak baku menggunakan algoritme *Jaro Winkler Distance*, pembobotan kata, kemudian dilakukan perhitungan *cosinenya*. Hasil klasifikasi pada kelas yang sudah didefinisikan yakni kelas positif, dan kelas negatif pada masing-masing hasil klasifikasi baik tanpa perbaikan kata maupun yang sudah melalui perbaikan kata, hasil klasifikasi analisis sentimen ulasan tersebut yang nantinya menjadi *output* sistem. Proses dari alur penyelesaian masalah pada Gambar 1.



Gambar 1. Alur Penyelesaian Masalah

Berdasarkan pada Gambar 1 tentang alur klasifikasi data *input* dan perbaikan kata, terdapat beberapa proses kerja sistem. Proses diawali dengan pemberian data *input* berupa data ulasan dari ulasan toko online “Lazada”, dan data kamus yang digunakan sebagai acuan perbaikan kata. Segala bentuk *input* data disimpan dalam bentuk format .txt agar memudahkan pemrosesan data.

#### 4. HASIL PENGUJIAN DAN ANALISIS

##### 4.1 Pengujian Pengaruh Nilai $k$

Pada proses pengujian yang pertama ini adalah pengujian pengaruh nilai  $k$ . Proses pengujian dilakukan untuk menemukan pada nilai  $k$  yang paling baik atau optimal yang digunakan dalam proses klasifikasi K-NN yang nantinya akan mempengaruhi hasil *accuracy* dari sistem. Nilai  $k$  yang digunakan pada pengujian ini bervariasi yang mana setiap nilai  $k$  akan menjadi parameter dari pengujian. Pengujian ini dilakukan dengan dua tahapan yakni tahap 1 untuk proses *preprocessing* tanpa perbaikan kata, dan tahap 2 untuk *preprocessing* dengan perbaikan kata.

##### 4.1.1 Pengujian Pengaruh Nilai $k$ Tanpa Perbaikan Kata

Pada pengujian pertama tanpa perbaikan kata dilakukan untuk mengetahui nilai  $k$  mana yang merupakan solusi paling baik atau optimal yang digunakan dalam proses klasifikasi, pengujian ini menggunakan 450 data latih dengan 50 data uji. Pengujian pengaruh nilai  $k$  dengan proses *preprocessing* tanpa perbaikan kata atau tahap 1 ditunjukkan pada tabel 1.

**Tabel 1 Pengujian Pengaruh Nilai  $k$  Tanpa Perbaikan Kata**

No	Nilai $k$ -	<i>Precision</i>	<i>Recall</i>	<i>Accuracy</i> (%)
1	3	0,72	1	72
2	5	0,72	1	72
3	10	0,72	1	72
4	15	0,72	1	72
5	20	0,74	1	74
6	25	0,72	1	72
7	30	0,72	1	72
8	40	0,7	1	70
9	50	0,68	1	68

Berdasarkan Tabel 1, bisa dilihat jika nilai *recall* untuk setiap nilai  $k$  untuk seluruh data yang diujikan memperoleh nilai yang optimal yaitu 1. Sedangkan untuk perolehan nilai *precision* pada seluruh data mendapatkan hasil nilai yang berbeda, karena dipengaruhi oleh jumlah kesesuaian antara analisis yang dilakukan pakar dan sistem. Pada tabel diatas diketahui jika nilai  $k$  yang optimal adalah pada  $k=20$  dengan *accuracy* sebesar 74%.

##### 4.1.2 Pengujian Pengaruh Nilai $k$ Dengan Perbaikan Kata

Pada pengujian pertama dengan perbaikan kata dilakukan untuk mengetahui nilai  $k$  mana yang merupakan solusi paling baik atau optimal yang digunakan dalam proses klasifikasi, pengujian ini menggunakan 450 data latih dengan 50 data uji. Pengujian pengaruh nilai  $k$  dengan proses *preprocessing* dengan perbaikan kata atau tahap 2 ditunjukkan pada tabel 2.

**Tabel 2 Pengujian Pengaruh Nilai  $k$  Dengan Perbaikan Kata**

No	Nilai $k$ -	<i>Precision</i>	<i>Recall</i>	<i>Accuracy</i> (%)
1	3	0,74	1	74
2	5	0,74	1	74
3	10	0,74	1	74
4	15	0,74	1	74
5	20	0,76	1	76
6	25	0,7	1	70
7	30	0,7	1	70
8	40	0,7	1	70
9	50	0,68	1	68

Berdasarkan Tabel 2, bisa dilihat jika nilai *recall* untuk setiap nilai  $k$  untuk seluruh data yang diujikan memperoleh nilai yang optimal yaitu 1. Sedangkan untuk perolehan nilai *precision* pada seluruh data mendapatkan hasil nilai yang berbeda. Perubahan nilai *precision* dipengaruhi oleh jumlah kesesuaian antara analisis yang dilakukan pakar dan sistem. Pada tabel diatas diketahui jika nilai  $k$  yang optimal adalah pada  $k=20$  dengan nilai *accuracy* sebesar 76%.

##### 4.2 Pengujian Pengaruh Banyaknya Data Latih

Proses pengujian kedua dilakukan berdasarkan nilai  $k$  terbaik yang sudah didapatkan dari proses pengujian yang telah dilakukan sebelumnya. Dari nilai  $k$  tersebut akan dilakukan pengujian dengan memberikan data latih secara bertahap dari 250 data, 300 data, 350 data, 400 data, hingga 450 data dengan setiap data latih yang diberikan diujikan dengan 50 data uji. Pada pengujian ini bertujuan untuk mendapatkan jumlah data latih yang memberikan hasil *accuracy* optimal terhadap sistem. Pengujian ini dilakukan dengan dua

tahapan yakni tahap 1 untuk proses *preprocessing* tanpa perbaikan kata, dan tahap 2 untuk *preprocessing* dengan perbaikan kata.

#### 4.2.1 Pengujian Pengaruh Banyaknya Data Latih Tanpa Perbaikan Kata

Pada pengujian ini dilakukan berdasarkan nilai  $k$  terbaik yang sudah didapatkan dari pengujian sebelumnya, pengujian ini menggunakan data latih secara bertahap dari 250 data, 300 data, 350 data, 400 data, hingga 450 data dengan setiap data latih yang diberikan diujikan dengan 50 data uji. Pengujian pengaruh banyaknya data latih dengan proses *preprocessing* tanpa perbaikan kata atau tahap 1 ditunjukkan pada tabel 3.

**Tabel 3 Pengujian Pengaruh Banyaknya Data Latih Tanpa Perbaikan Kata**

No	Jumlah Data	Precision	Recall	Accuracy (%)
1	250	0,72	1	72
2	300	0,7	1	70
3	350	0,7	1	70
4	400	0,7	1	70
5	450	0,74	1	74

Berdasarkan Tabel 3, bisa dilihat jika nilai *recall* untuk setiap nilai  $k$  untuk seluruh data yang diujikan memperoleh nilai yang optimal yaitu 1. Sedangkan untuk perolehan nilai *precision* pada seluruh data mendapatkan hasil nilai yang berbeda. Perubahan nilai *precision* dipengaruhi oleh jumlah kesesuaian antara analisis yang dilakukan pakar dan sistem. Pada tabel diatas diketahui jika nilai *accuracy* yang optimal adalah pada data ke 450 dengan *accuracy* sebesar 74%.

#### 4.2.2 Pengujian Pengaruh Banyaknya Data Latih Dengan Perbaikan Kata

Pada pengujian ini dilakukan berdasarkan nilai  $k$  terbaik yang sudah didapatkan dari pengujian sebelumnya, pengujian ini menggunakan data latih secara bertahap dari 250 data, 300 data, 350 data, 400 data, hingga 450 data dengan setiap data latih yang diberikan diujikan dengan 50 data uji. Pengujian pengaruh banyaknya data latih dengan proses *preprocessing* dengan perbaikan kata atau tahap 1 ditunjukkan pada tabel 4.

**Tabel 4 Pengujian Pengaruh Banyaknya Data Latih Dengan Perbaikan Kata**

No	Jumlah Data	Precision	Recall	Accuracy (%)
1	250	0,72	1	72
2	300	0,7	1	70
3	350	0,66	1	66
4	400	0,64	1	64
5	450	0,76	1	76

Berdasarkan Tabel 4, bisa dilihat jika nilai *recall* untuk setiap nilai  $k$  untuk seluruh data yang diujikan memperoleh nilai yang optimal yaitu 1. Sedangkan untuk perolehan nilai *precision* pada seluruh data mendapatkan hasil nilai yang berbeda. Perubahan nilai *precision* dipengaruhi oleh jumlah kesesuaian antara analisis yang dilakukan pakar dan sistem. Pada tabel diatas diketahui jika nilai *accuracy* yang optimal adalah pada data ke 450 dengan *accuracy* sebesar 76%.

#### 4.3 Pengujian Pengaruh Perbaikan Kata

Pada pengujian ini dilakukan untuk mengetahui nilai dari *precision*, *recall*, dan *accuracy* pada masing-masing pengujian dengan menggunakan nilai  $k$  dan data latih optimal yang didapatkan. Pengujian ini dilakukan dengan dua tahapan yakni tahap 1 untuk proses *preprocessing* tanpa perbaikan kata, dan tahap 2 untuk *preprocessing* dengan perbaikan kata. Hasil pengujian ditunjukkan pada tabel 5.

**Tabel 5 Pengujian Pengaruh Penggunaan Perbaikan Kata**

Pengujian	Precision	Recall	Accuracy (%)
Tahap 1	0,74	1	74
Tahap 2	0,76	1	76

Berdasarkan Tabel 5, bisa dilihat jika analisis sentimen tanpa perbaikan kata nilai *precision* sebesar 0,74, nilai *recall* sebesar 1, dan nilai *accuracy* sebesar 74 %. Sedangkan untuk analisis sentimen dengan perbaikan kata nilai *precision* sebesar 0,76, nilai *recall* sebesar 1, dan nilai *accuracy* sebesar 76 %. Sehingga analisis sentimen akan lebih baik jika dilakukan perbaikan kata terlebih dahulu.

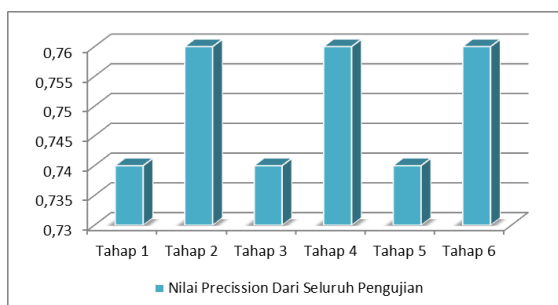


#### 4.4 Hasil Dari Seluruh Pengujian

Pada pengujian ini dilakukan untuk mengetahui hasil dari setiap proses pengujian yang telah dilakukan, hasil dari setiap pengujian akan ditampilkan dalam bentuk grafik yang terdiri dari nilai *precision*, *recall*, dan *accuracy*. Pada grafik hasil seluruh pengujian akan menampilkan 6 tahapan yang ada pada setiap proses pengujian dengan urutan sebagai berikut:

- Tahap 1  
Pengaruh nilai  $k$  tanpa perbaikan kata.
- Tahap 2  
Pengaruh nilai  $k$  dengan perbaikan kata.
- Tahap 3  
Pengaruh banyaknya data latih tanpa perbaikan kata.
- Tahap 4  
Pengaruh banyaknya data latih dengan perbaikan kata.
- Tahap 5  
Pengaruh penggunaan perbaikan kata tanpa perbaikan kata di proses *preprocessing*.
- Tahap 6  
Pengaruh penggunaan perbaikan kata dengan perbaikan kata di proses *preprocessing*.

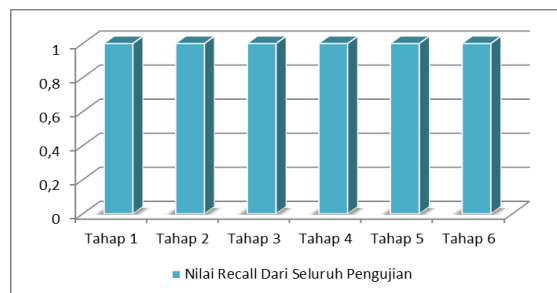
##### 4.4.1 Hasil Dari Seluruh Pengujian *Precision*



Gambar 2 Nilai *Precision* Dari Seluruh Pengujian

Berdasarkan Gambar 2 yang menunjukkan hasil *precision* dari seluruh pengujian maka dapat disimpulkan jika nilai *precision* akan cenderung lebih tinggi ketika terdapat perbaikan kata terlebih dahulu karena ketika dalam *preprocessing* kata yang tidak baku akan diperbaiki menggunakan kamus singkatan. Nilai *precision* ditentukan oleh tingkat ketepatan prediksi sistem sesuai dengan perhitungan prediksi benar dari seluruh total data latih, maupun prediksi salah dari seluruh total data latih yang diberikan.

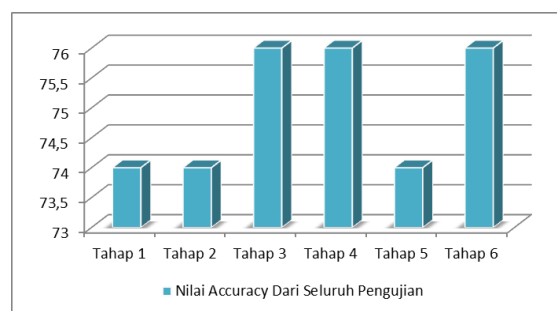
##### 4.4.2 Hasil Dari Seluruh Pengujian *Recall*



Gambar 3 Nilai *Recall* Dari Seluruh Pengujian

Berdasarkan Gambar 3 yang menunjukkan hasil *recall* dari seluruh pengujian maka dapat disimpulkan jika nilai *recall* menunjukkan hasil yang optimal yaitu 1 yang berarti tingkat keberhasilan sistem dalam mengenali suatu kelas sangat baik, tidak hanya pada pengujian dengan perbaikan kata, namun juga dalam pengujian tanpa perbaikan.

##### 4.4.3 Hasil Seluruh Pengujian *Accuracy*



Gambar 4 Nilai *Accuracy* Dari Seluruh Pengujian

Berdasarkan Gambar 4 yang menunjukkan hasil *accuracy* dari seluruh pengujian maka dapat disimpulkan jika nilai *accuracy* menunjukkan hasil yang tinggi ketika pada pengujian dengan perbaikan kata, karena ketika dalam *preprocessing* kata yang tidak baku akan diperbaiki menggunakan kamus singkatan. Namun, dalam pengujian tahap 3 yakni pengujian tentang pengaruh banyaknya data latih tanpa perbaikan kata juga mendapatkan *accuracy* yang tinggi karena pada pengujian tersebut sebelumnya sudah dilakukan pengujian untuk mengetahui pengaruh nilai  $k$ , dan didapatkan nilai  $k=20$  yang paling optimal. Sehingga dari nilai  $k$  tersebut mempengaruhi *accuracy* dari kedua pengujian pengaruh banyaknya data latih, baik tanpa perbaikan kata maupun dengan perbaikan kata keduanya memperoleh hasil *accuracy* yang optimal.

## 5. KESIMPULAN

Setelah seluruh rangkaian proses pengujian dan analisis selesai dilakukan untuk permasalahan analisis sentimen dan perbaikan kata menggunakan *K-Nearest Neighbor* (K-NN) dan *Jaro Winkler Distance*, sehingga didapatkan kesimpulan jika metode *K-Nearest Neighbor* (K-NN) dapat diterapkan dan dilakukan proses implementasi dengan baik untuk menentukan analisis sentimen pada ulasan “Lazada” berbahasa Indonesia, dan perbaikan kata menggunakan metode *Jaro Winkler Distance* dapat menghasilkan perbaikan kata yang baik. Dari ketiga hasil pengujian yang telah dilakukan, didapatkan nilai  $k$  terbaik adalah  $k=20$ , dengan data latih optimal sebanyak 450 data, maka didapatkan nilai *precision* sebesar 0,74, nilai *recall* sebesar 1, dan nilai *accuracy* sebesar 74 % untuk analisis sentimen tanpa perbaikan kata, serta nilai *precision* sebesar 0,76, nilai *recall* sebesar 1, dan nilai *accuracy* sebesar 76 % untuk analisis sentimen dengan perbaikan kata. Sehingga dapat disimpulkan bahwa analisis sentimen menggunakan perbaikan kata lebih baik daripada analisis sentimen tanpa perbaikan kata yang dapat dilihat pada nilai *precision*, dan *accuracy*.

## 6. DAFTAR PUSTAKA

- Antinasari, P., Setya Perdana, R., & Ali Fauzi, M. 2017. Analisis Sentimen Tentang Opini Film Pada Dokumen Twitter Berbahasa Indonesia Menggunakan Naive Bayes Dengan Perbaikan Kata Baku, 1-9.
- Baskoro, S., Ahmad, R., & Furqon, M. 2015. Pencarian Pasal Pada Kitab Undang-Undang Hukum Pidana (KUHP) Berdasarkan Kasus Menggunakan Metode *Cosine Similarity* dan *Latent Semantic Indexing* (LSI), 1-6.
- Christian, H., Agus, M. P., & Suhartono, D. 2016. *Single Document Automatic Text Summarization Using Term Frequency-Inverse Document Frequency* (TF-IDF), 1-10.
- Estu Nurjanah, W., Setya Pradana, R., & Ali Fauzi, M. 2017. Analisis Sentimen Terhadap Tayangan Televisi Berdasarkan Opini Masyarakat pada Media Sosial *Twitter* Menggunakan Metode *K-Nearest Neighbor* dan Pembobotan Jumlah *Retweet*, 1750-1757.
- Hasugian, J. 2006. Penelusuran Informasi Ilmiah Secara *Online*: Perlakuan terhadap Seorang Pencari Informasi sebagai *Real User*, 1-13.
- Kurniawati, A., Pupitodjati, S., & Rahman, S. 2016. Implementasi Algoritme *Jaro Winkler Distance* Untuk Membandingkan Kesamaan Dokumen Berbahasa Indonesia, 1-4.
- Lestari, N. P. 2015. Uji *Recall And Precision* Sistem Temu Kembali Informasi *Opac* Perpustakaan ITS Surabaya, 1-18.
- Pustejovsky, J., & Stubbs, A. 2012. *Natural language Annotation for Machine Learning*. Cambridge: O'Reilly.
- Robertson, S. 2004. *Understanding Inverse Document Frequency: On Theoretical Arguments for IDF*, 1-19.
- Rocmawati, R., & Kusumaningrum, R. 2015. Studi Perbandingan Algoritme Pencarian *String* Dalam Metode *Approximate String Matching* Untuk Identifikasi Kesalahan Pengetikan Teks, 1-11.
- Xia, T., & Chai, Y. 2011. *An Improvement to TF-IDF: Term Distribution based Term Weight Algorithm*, 1-8.