Analisis Sentimen pada Ulasan Produk Kecantikan Menggunakan *K-Nearest Neighbor* dan *Information Gain*

Ekky Yulianti Prastika S.[1], Said Al Faraby[2], Mahendra Dwifebri P.[3]

[1,2,3] Universitas Telkom, Bandung
ekkyyps@students.telkomuniversity.ac.id[1], saidalfaraby@telkomuniversity.ac.id[2], mahendradp@telkomuniversity.ac.id[3]

Abstrak

Adanya ulasan pada situs online membuat masyarakat lebih tertarik dalam membeli suatu produk. Untuk membeli suatu produk, khususnya produk kecantikan membutuhkan banyak pertimbangan. Selain itu, masyarakat harus teliti dalam membaca ulasan untuk menentukan apakah suatu produk akan dibeli atau tidak. Namun, ulasan tersebut jumlahnya sangat banyak dan beragam. Hal ini menyebabkan ulasan menjadi bias. Oleh karena itu, dibutuhkan analisis sentimen berbasis aspek dengan pendekatan machine learning. Analisis sentimen berbasis aspek membantu untuk mencari sentimen pada tiap kalimat di suatu ulasan produk. Penelitian ini menggunakan metode machine learning yaitu K-Nearest Neighbor (KNN) dengan dataset female daily. KNN merupakan salah metode sederhana dan efisien tetapi menghasilkan bias. Masalah tersebut dapat diatasi dengan menerapkan metode seleksi fitur. Metode seleksi fitur yang digunakan adalah Information Gain (IG). Pengujian dilakukan pada tahapan preprocessing, seleksi fitur, dan klasifikasi. Hasil pengujian menunjukkan data yang menggunakan stemming, normalisasi, threshold IG 0,5, dan nilai k = 23 pada KNN menghasilkan akurasi terbaik sebesar 74,21%.

Kata kunci : analisis sentiment, berbasis aspek, *knn, information gain,* ulasan produk, *female daily*

Abstract

Reviews on online sites make people interested to buy a product. To buy a product, especially a beauty product need a lot of consideration. In addition, people must be careful in reading reviews to determine purchased a product or not. However, these reviews are numerous and varied. It causes the reviews to be biased. Therefore, it needs an aspect-based sentiment analysis with a machine learning approach to fix this problem. Aspect-based sentiment analysis helps to find the sentiment in each sentence in product reviews. This research, machine learning that used is KNN with a female daily dataset. KNN is one of the simple methods and efficient but biased. It can be solved by applying the feature selection method. The feature selection method is Information Gain (IG). In this research, scenario tests are preprocessing steps, feature selection, and classification steps. The test results show that data using stemming, normalization, threshold IG 0,5, and the value of k = 23 on KNN produces the best accuracy is 74,21%

Keywords: sentiment analysis, aspect-based, knn, information gain, product reviews, female daily

1. Pendahuluan

Latar Belakang

Membaca suatu ulasan produk dapat membuat masyarakat lebih tertarik dengan produk yang memberikan ulasan baik berdasarkan standar deviasi sebesar 5,06 daripada informasi yang berasal dari tim marketing hanya sebesar 4,36 [9]. Produk kecantikan membutuhkan banyak pertimbangan saat akan membeli, karena setiap kulit memiliki kondisi yang berbeda. Sehingga, masyarakat harus teliti dalam membaca ulasannya. Namun, ulasan yang ditulis pada situs online jumlahnya sangat banyak dan beragam. Hal ini menyebabkan ulasan menjadi bias. Berdasarkan kondisi tersebut, dibutuhkan analisis sentimen berbasis aspek karena tiap kalimat dapat memuat beberapa sentimen [5]. Analisis sentimen tingkat aspek dapat memberikan ringkasan ulasan yang dilakukan secara keseluruhan dokumen [13]. Hal ini dapat diselesaikan dengan pendekatan machine learning karena terbukti terselesaikan lebih akurat sebesar 70-80% daripada manusia hanya 69% [16]. Pada penelitian [19] membahas teks klasifikasi menggunakan metode K-Nearest Neighbor (KNN) dan Information Gain (IG) dengan dataset WebKB, Reuters-21578, Ling-Spam, WWW Prisoner, WWW Beethoven, dan Usenet menunjukkan bahwa metode KNN bekerja secara maksimal, menghasilkan akurasi sebesar 95,5%. Sedangkan pada penelitian [3] menggunakan metode KNN dengan dataset the Polarity v2.0 from Cornell movie review menghasilkan akurasi sebesar 60%, tetapi dengan menambahkan metode IG akurasi meningkat menjadi 96,8% karena fitur yang tidak relevan dihapus. Namun, pada kedua penelitian tersebut hanya membahas tentang klasifikasi teks dan analisis sentimen tingkat dokumen yang hanya mengelompokkan opini ke dalam kelas sentimen positif atau negatif. Pada proposal ini akan membahas tentang analisis sentimen tingkat aspek dengan metode machine

learning, yaitu K-Nearest Neighbor (KNN) dengan pemilihan fitur Information Gain (IG). KNN merupakan metode yang menghasilkan bias namun sederhana dan efisien [19]. Permasalahan tersebut dapat diatasi dengan pemilihan fitur. Menurut penelitian [3, 17, 19], metode terbaik untuk pemilihan fitur yang berhubungan adalah IG. Sehingga, KNN mendapatkan kinerja yang lebih baik [3].

Topik dan Batasannya

Topik pada penelitian ini adalah membuat sistem pada analisis sentimen level aspek untuk mengetahui secara detail hasil sentimen berdasarkan aspek yang telah ditentukan. Batasan masalah pada penelitian ini adalah dataset yang berasal dari *website Female Daily* dengan aspek harga, produk, packaging, dan aroma, yang berisikan sentimen positif, netral, dan negatif. Dalam penelitian ini disebutkan bahwa aspek disebut sebagai kelas dan sentimen dianggap sebagai label.

Tujuan

Tujuan dari penelitian ini adalah untuk menganalisis sentimen berbasis aspek pada ulasan produk kecantikan dengan penggunaan stemming dan normalisasi. Serta menganalisis dan mencari threshold terbaik pada IG serta nilai k yang optimal pada metode KNN. Penelitian ini diterapkan pada dataset yang diambil dari *website female daily*.

Organisasi Tulisan

Bagian selanjutnya yang akan dibahas adalah pada bab 2 membahas studi literatur yang terkait dengan penelitian ini. Pada bab 3 membahas teori dan rancangan sistem penelitian. Hasil dan analisis dibahas pada bab 4. Kemudian, kesimpulan dari hasil penelitian yang sudah dilakukan dibahas pada bab 5.

2. Studi Terkait

Saat ini sudah banyak penelitian yang telah menerapkan machine learning untuk analisis sentimen. Beberapa metode *machine learning* yang telah diterapkan adalah metode *Naive Bayes*, SVM, KNN, dan Random Forest. Pada penelitian yang dilakukan oleh Fabrizio [17] membuktikan bahwa machine learning lebih efektif dan efisien. *Machine learning* saat ini juga banyak diterapkan untuk kasus analisis sentimen baik tingkat dokumen maupun tingkat aspek. Analisis sentimen atau *opinion mining* merupakan proses untuk menganalisis dan mendeteksi polaritas dalam suatu teks atau dapat dikatakan untuk mengetahui perasaan dan emosi seseorang yang dituangkan dalam sebuah teks [19, 5]. Tujuan dari analisis sentimen adalah menentukan opini sekelompok orang mengenai beberapa topik tertentu [14].

Beberapa riset telah mengkombinasikan pemilihan fitur dengan metode klasifikasi untuk meningkatkan kinerja sistemnya. Pada penelitian sentimen berbasis aspek [12], mendapatkan hasil akurasi sebesar 78,12% dengan menggunakan metode *Naive Bayes* untuk klasifikasi, *PostTagging*, dan *Chi Square* untuk pemilihan fitur. Selain itu, penelitian tersebut juga menghasilkan nilai F1-measure untuk klasifikasi sentimen sebesar 62,5%, dan *F1-measure* untuk klasifikasi aspek sebesar 85,41%. Akan tetapi, pemilihan metode *Chi Square* sebagai pemilihan fitur pada penelitian ini menyebabkan turunnya kinerja dari sistem yang dibangun.

Penelitian [18] membuktikan bahwa *K-Nearest Neighbor* (KNN) memiliki akurasi yang lebih unggul daripada NB meskipun fiturnya banyak yang direduksi. Akurasi dari KNN sebesar 95,5% sedangkan NB hanya sebesar 92,7%. Hal ini menunjukkan bahwa dengan jumlah fitur berapapun maupun jumlah fitur yang digunakan sangat sedikit, KNN masih mendapatkan akurasi yang tinggi. Pada penelitian [3] yang menggunakan metode KNN terbukti memiliki akurasi yang tinggi pada analisis sentimen ulasan film yaitu sebesar 96,8% daripada NB, SVM, dan *Random Forest* (RF). Penelitian ini juga mengkombinasikan KNN dengan IG sebagai pemilihan fitur. Penggunaan IG terbukti meningkatkan kinerja KNN karena akurasi awal dari KNN hanya sebesar 60% menjadi 96,8%. Hal ini dapat terjadi karena menerapkan IG sebagai metode pemilihan fitur.

[16] menggunakan metode IG menghasilkan akurasi sebesar 96% dengan *IG Classifier* sebagai metode klasifikasi. Penentuan threshold IG dibantu dengan menggunakan Document Frequency (DF) menghasilkan *threshold* tertinggi dengan nilai 0,5 pada *dataset Polarity v.2.0 from Cornell review*. Untuk menangani fitur yang berjumlah besar dan memberikan klasifikasi sentimen yang lebih baik, maka diperlukan pemilihan fitur yang dikombinasikan dengan metode klasifikasi. Dengan melakukan hal tersebut, kinerja yang dihasilkan dari sistem menjadi lebih baik.

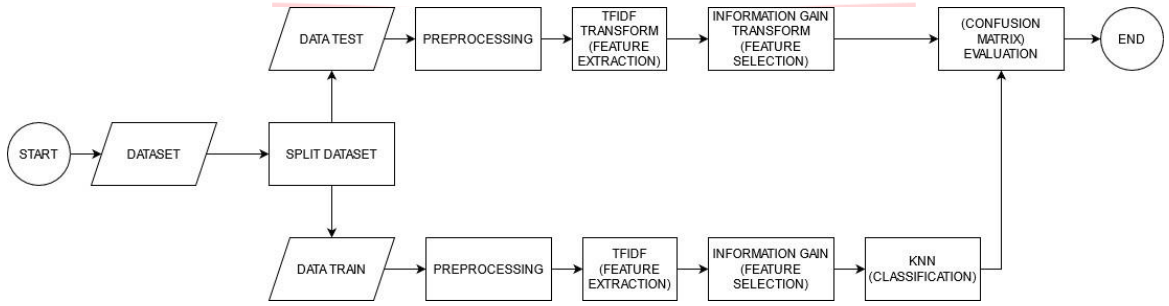
Sedangkan pada penelitian [10], menggunakan ekstraksi fitur dengan metode Word2Vec dan metode klasifikasi dengan Support Vector Machine (SVM) menghasilkan akurasi sebesar 70%. Penelitian ini diimplementasikan pada dataset berjumlah 772 ulasan yang didapatkan dari situs ulasan produk kecantikan Female Daily. Sedangkan dengan menggunakan *Term Frequency and Inverse Document Frequency* (TF-IDF) akurasi yang diperoleh sebesar 85%. Hal ini membuktikkan bahwa penggunaan TF-IDF memberikan pengaruh terhadap kinerja dari sistem. Kekurangan pada penelitian ini adalah jumlah dataset yang digunakan sedikit sehingga akurasi yang dihasilkan kurang maksimal. Penelitian yang telah dilakukan oleh [2, 4] yang

menggunakan *dataset female daily*, menunjukkan bahwa adanya pemroresan data mempengaruhi hasil dari kinerja sistem yang dibangun. Dengan tidak menggunakan *stopword removal*, penelitian yang dilakukan oleh [2] menghasilkan *F-1 Score* sebesar 62,81%. Sedangkan pada penelitian [4], penggunaan stemming dan tanpa penggunaan stopword removal menghasilkan akurasi sebesar 88,35%.

Berdasarkan penelitian yang sudah dijelaskan, maka pada penelitian ini menggunakan dataset sejumlah 3960 yang diambil dari *website Female Daily*. Sistem dibangun dengan menerapkan metode yang telah menghasilkan kinerja sistem yang bagus. Metode tersebut adalah TF-IDF sebagai ekstraksi fitur, *Information Gain* sebagai pemilihan fitur, serta metode KNN untuk melakukan proses klasifikasi sentimen. Klasifikasi sentimen yang akan dilakukan pada penelitian ini adalah klasifikasi sentimen tingkat aspek.

3. Sistem yang Dibangun

Pada bagian ini dijelaskan mengenai desain sistem yang dibangun. Berikut desain sistem yang ada akan menggambarkan alur kerja dari pembangunan sistem:



Penjelasan dari sistem yang dibangun akan dijelaskan pada sub-bab selanjutnya, yaitu penjelasan terkait dataset yang akan digunakan, tahapan praproses data, ekstraksi fitur dengan TF-IDF, pemilihan fitur dengan *Information Gain*, dan penggunaan metode KNN untuk klasifikasi sentimen, serta evaluasi atau pengukuran kinerja sistem.

3.1 *Dataset*

Dataset diperoleh dari *website female daily* dari 5 kategori produk yaitu *toner*, *serum*, *scrub exfoliator*, *lipstick*, dan *sunscreen*. Pelabelan *dataset* dilakukan secara manual oleh peneliti sebelumnya[2]. Tahap pertama dalam pelabelan adalah mengidentifikasi setiap aspek atau kelas yang digunakan. Dalam *dataset* ini, menggunakan 4 aspek, yaitu harga, *packaging*, produk, dan aroma. Kemudian, setiap aspek diklasifikasikan menjadi 3 label atau sentimen, yaitu sentimen positif atau 1, negatif dengan nilai -1, dan netral bernilai 0.

Tabel 1. Contoh Ulasan

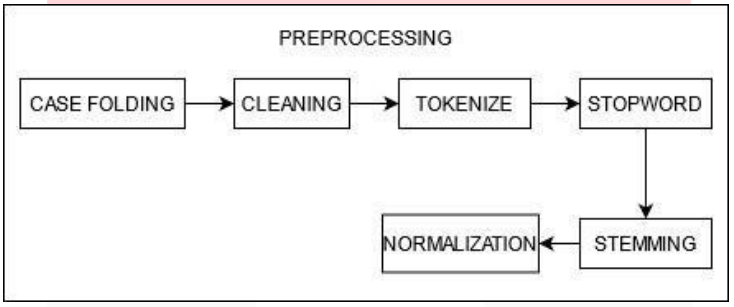
id	Teks ulasan	harga	*packaging*	produk	aroma
708	sunscreen termahal yang pernah gue beli ini kayanya. but it's worth it sih and will definitely buy again. sukanya sama suncreen ini: - high spf - nggak meninggalkan white cast. perfectly blends into the skin nggak membuat muka berminyak - very light - doesn't clog pores produk ini berhasil membuat gue jadi mau pakai sunscreen	-1	0	1	0

Setelah diketahui label dari tiap aspek, *dataset* akan dibagi atau di*split* menjadi 80% data *train* yaitu sejumlah 3168 data dan sejumlah 792 data atau sebesar 20% data *test*. Berikut dapat dilihat perbandingan antara persebaran data yang ada di tiap aspek pada data *train* maupun data *test*:

Gambar 2. Persebaran Data *Train* dan Data *Test*

3.2 *Preprocessing Data*

Tahap *preprocessing* dilakukan agar *dataset* lebih mudah digunakan untuk proses klasifikasi. Selain itu, tahapan *preprocessing* membuat *dataset* lebih seragam dan menghilangkan *noise* pada *dataset*. Berikut tahapan dalam *preprocessing* data:



Gambar 3. Alur praproses data

Berikut penjelasan dari tiap tahapan *preprocessing* data:

1. *Case Folding* merupakan tahapan praproses dimana setiap kata pada data diseragamkan menjadi huruf kecil semua. Berikut contoh dari *case folding*:

Tabel 2. Contoh *Case Folding*

Kalimat Masukan	Hasil *Case Folding*
Sunscreen termahal yang pernah gue beli ini kayanya. but it's worth it sih and will definitely buy again. sukanya sama suncreen ini: - high spf - nggak meninggalkan white cast. perfectly blends into the skin nggak membuat muka berminyak - very light - doesn't clog pores produk ini berhasil membuat gue jadi mau pakai sunscreen :)	sunscreen termahal yang pernah gue beli ini kayanya. but it's worth it sih and will definitely buy again. sukanya sama suncreen ini: - high spf - nggak meninggalkan white cast. perfectly blends into the skin nggak membuat muka berminyak - very light - doesn't clog pores produk ini berhasil membuat gue jadi mau pakai sunscreen :)

2. *Cleaning* adalah tahapan pada praproses data yang bertujuan untuk menghapus *punctuation* atau tanda baca, angka, dan/atau menghapus simbol-simbol pada data. Proses ini termasuk *negation handling* digunakan untuk membantu menghindari ambigu pada kata karena adanya apostrof, misalnya "*doesn't*" menjadi "*does not*". Berikut contoh dari tahapan *cleaning*:

Tabel 3. Contoh *Cleaning*

Kalimat Masukan	Hasil *Cleaning*
sunscreen termahal yang pernah gue beli ini kayanya. but it's worth it sih and will definitely buy again. sukanya sama suncreen ini: - high spf - nggak meninggalkan white cast. perfectly blends into the skin nggak membuat muka berminyak - very light - doesn't clog pores produk ini berhasil membuat gue jadi mau pakai sunscreen :)	sunscreen termahal yang pernah gue beli ini kayanya but it worth it sih and will definitely buy again sukanya sama suncreen ini high spf nggak meninggalkan white cast perfectly blends into the skin nggak membuat muka berminyak very light does not clog pores produk ini berhasil membuat gue jadi mau pakai sunscreen

3. *Tokenization* atau tokenisasi adalah tahap memisahkan atau memotong kalimat menjadi sebuah kata atau token pada suatu data. Berikut contoh penerapan tokenisasi:

Tabel 4. Contoh *Tokenization*

Kalimat Masukan	Hasil *Tokenization*
sunscreen termahal yang pernah gue beli ini kayanya but it worth it sih and will definitely buy again sukanya sama suncreen ini high spf nggak meninggalkan white cast perfectly blends into the skin nggak membuat muka berminyak very light does not clog pores produk ini berhasil membuat gue jadi mau pakai sunscreen	['sunscreen', 'termahal', 'gue', 'beli', 'kayanya', 'worth', 'sih', 'definitely', 'buy', 'sukanya', 'suncreen', 'high', 'spf', 'nggak', 'meninggalkan', 'white', 'cast', 'perfectly', 'blends', 'skin', 'nggak', 'muka', 'berminyak', 'light', 'not', 'clog', 'pores', 'produk', 'berhasil', 'gue', 'pakai', 'sunscreen']

4. *Stopwords Removal* merupakan tahapan menghapus kata-kata yang dianggap kurang berpengaruh atau kurang memiliki informasi pada suatu kalimat. Pada Tugas Akhir ini akan menggunakan 2 jenis *stopword* dari nltk, yaitu *stopword* bahasa Indonesia dan *stopword* bahasa Inggris. Penggunaan dua *stopword*, dikarenakan *dataset* yang digunakan adalah *dataset* multibahasa. Contoh *stopword* dalam bahasa Indonesia adalah "*di*", "*yang*", "*dan*", dan lain sebagainya. Sedangkan *stopword* bahasa Inggris misalnya "*and*", "*but*", "*it*", dan lain-lain. Pada penelitian ini, kata "baik", "guna", "kurang", "tidak", "enggak", *"not", "no"* dihapus dalam kamus agar tidak mempengaruhi hasil sentimen. Berikut contoh dari *stopword removal*:

Tabel 5. Contoh *Stopword Removal*

Kalimat Masukan	Hasil *Stopword Removal*
sunscreen termahal yang pernah gue beli ini kayanya but it worth it sih and will definitely buy again sukanya sama suncreen ini high spf nggak meninggalkan white cast perfectly blends into the skin nggak membuat muka berminyak very light does not clog pores produk ini berhasil membuat gue jadi mau pakai sunscreen	sunscreen termahal gue beli kayanya worth sih definitely buy sukanya suncreen high spf nggak meninggalkan white cast perfectly blends skin nggak muka berminyak light not clog pores produk berhasil gue pakai sunscreen

5. *Stemming*, merupakan tahapan praproses yang bertujuan untuk menghapus imbuhan yang terdapat pada kata atau mengubah kata menjadi bentuk dasarnya. Berikut contoh penerapan *stemming* pada salah satu ulasan:

Tabel 6. Contoh *Stemming*

Kalimat Masukan	Hasil *Stemming*
sunscreen termahal gue beli kayanya worth sih definitely buy sukanya suncreen high spf nggak meninggalkan white cast perfectly blends skin nggak muka berminyak light not clog pores produk berhasil gue pakai sunscreen	sunscreen mahal gue beli kaya worth sih definitely buy suka suncreen high spf nggak tinggal white cast perfectly blends skin nggak muka minyak light not clog pores produk hasil gue pakai sunscreen

6. Normalisasi, merupakan tahapan praproses yang bertujuan untuk mengubah kata tidak baku menjadi kata baku. Terdapat 427 kamus data normalisasi yang dibuat secara manual. Berikut contoh penerapan normalisasi:

Tabel 7. Contoh Normalisasi

Kalimat Masukan	Hasil Normalisasi
sunscreen mahal gue beli kaya worth sih definitely buy suka suncreen high spf nggak tinggal white cast perfectly blends skin nggak muka minyak light not clog pores produk hasil gue pakai sunscreen	sunscreen mahal saya beli seperti worth sih definitely buy suka suncreen high spf tidak tinggal white cast perfectly blends skin tidak muka minyak light not clog pores produk hasil saya pakai sunscreen

3.3 *Term Frequency and Inverse Document Frequency* (TF-IDF)

Salah satu metode ekstraksi fitur yang paling populer untuk kasus klasifikasi teks adalah TF-IDF. Metode ini digunakan untuk memberi bobot nilai berdasarkan tingkat kepentingan kata terhadap dokumen atau kategori dalam suatu kumpulan dokumen [11]. Hasil dari TF-IDF adalah berupa matriks yang didapatkan dari hasil perkalian *Term Frequency* dengan *Inverse Document Frequency*.

Term Frequency atau TF merupakan bobot nilai dari frekuensi *term* atau kata yang sering muncul pada dokumen. Sedangkan, *Inverse Document Frequency* atau IDF adalah jumlah dokumen terkait yang mengandung kata tertentu. Perhitungan TF-IDF dapat dilakukan menggunakan persamaan sebagai berikut [13]:

$$TFIDF_a = TF_a \times \log\left(\frac{N}{DF_a}\right) \quad (1)$$

Keterangan:

- TF*a* : banyaknya kemunculan setiap kata pada suatu dokumen *a*
- *N* : jumlah seluruh dokumen
- *DFa* : jumlah dokumen keseluruhan yang memuat kata *a*

Diberikan suatu kalimat dari dokumen 1 yaitu ”produk biasa aja” dan dokumen 2 ”produk ini mahal”, perhitungan bobot dengan TF-IDF dapat dilihat pada Tabel 8.

Tabel 8. Perhitungan TF-IDF

term	tf d1	tf d2	df	Idf	TF-IDF d1	TF-IDF d2
produk	1	1	2	log(2/2)= 0	0	0
biasa	1	0	1	log(2/1) = 0,3	0,3	0
aja	1	0	1	log(2/1) = 0,3	0,3	0
ini	0	1	1	log(2/1) = 0,3	0	0,3

mahal	0	1	1	log(2/1) = 0,3	0	0,3

3.4 *Information Gain* (IG)

IG merupakan salah satu metode terbaik dalam pemilihan fitur [16]. Pada analisis sentimen, IG digunakan untuk mengetahui dependensi antara dua variabel atau dikatakan untuk mencari keterkaitan antara suatu fitur dengan kelas [16]. Selain itu, IG dapat digunakan untuk meningkatkan kinerja dari sistem yang dibangun [3, 18]. Proses seleksi fitur didapatkan setelah pembobotan fitur dengan TF-IDF. Sebelum dilakukan seleksi fitur, dilakukan pemisahan antara kelas dan fitur sebagai parameter perhitungan MI. Pada sistem ini menggunakan perhitungan *Mutual Information* (MI) karena MI memiliki kinerja metode yang mirip dan perhitungan yang sama dengan IG [6, 20]. Berikut perhitungan IG berdasarkan rumus MI[1, 6, 16]:

$$MI(Y,X)= \frac{N11}{N}\log_2\frac{N.N11}{N_1.N_1} + \frac{N01}{N}\log_2\frac{N.N01}{N_0.N_1} + \frac{N10}{N}\log_2\frac{N.N10}{N_1.N_0} + \frac{N00}{N}\log_2\frac{N.N00}{N_0.N_0} \quad (2)$$

Keterangan:

- N : jumlah semua *term* yang ada
- N_{10} : jumlah kalimat yang memiliki nilai *ex = 1 (term x)* namun tidak di kelas *y*
- N_{01} : jumlah kalimat yang memiliki nilai *ex = 0 (term x)* namun berada di kelas *y*
- N_{00} : jumlah kalimat yang memiliki nilai *ex = 0 (term x)* dan tidak di kelas *y*
- N_{11} : jumlah kalimat yang memiliki nilai *ex = 1 (term x)* dan berada di kelas *y*

Contoh perhitungan MI pada aspek *price* dapat dilihat pada tabel 9:

Tabel 9. Contoh ulasan perhitungan MI

index	ulasan	jumlah kata	*price*
83	tonernya ringan banget sih	4	0
64	harganya mahal, produknya biasa aja	5	-1
36	aku suka banget sama produk ini, harganya sangat affordable	9	1

Berdasarkan ulasan pada tabel 9, dapat diperoleh tabel kebenaran yang ditunjukkan pada Tabel 10.

Tabel 10. Tabel Kebenaran *term* harga pada Aspek *price* terhadap label Positif

	ey = epositif = 1	*ey = epositif = 0*
ex = harga = 1	1	1
ex = harga = 0	8	8

Tabel 11. Perhitungan *term* harga menggunakan MI

Perhitungan	Hasil MI
(*harga*, *positif*) = $\frac{1}{18}\log_2\frac{18 \times 1}{(1+1)(8+1)} + \frac{8}{18}\log_2\frac{18 \times 8}{(8+8)(8+1)} + \frac{1}{18}\log_2\frac{18 \times 1}{(1+1)(1+8)} + \frac{8}{18}\log_2\frac{18 \times 1}{(8+8)(1+8)} = 0{,}004704$	0,004704

Tabel 12. Contoh hasil seleksi fitur pada aspek Harga

index	*term*	nilai MI
251	mahal	1.0
154	harga	0.8921208047225897
289	murah	0.7956005462154673

Tabel 13. Contoh hasil seleksi fitur pada aspek *Packaging*

index	*term*	nilai MI
199	kemasan	1.0
198	kemas	0.8295905694219587
319	packaging	0.7055088874941323

Tabel 14. Contoh hasil seleksi fitur pada aspek Produk

index	*term*	nilai MI
191	kapas	1.0
452	toner	0.9978382095313082
445	tidak	0.9006909102810929

Tabel 15. Contoh hasil seleksi fitur pada aspek Aroma

index	*term*	nilai MI
476	wangi	1.0
19	aroma	0.396260420067444
114	enak	0.38701613803247575

Berdasarkan contoh hasil perhitungan seleksi fitur, dapat dilihat bahwa kata atau *terms* yang memiliki nilai paling tinggi merupakan fitur yang berelevan dengan kelasnya. Selanjutnya, *terms* yang mempunyai nilai IG lebih dari *threshold* untuk setiap aspek, nilainya diganti dengan nilai TF-IDF. Sehingga, *terms* yang mempunyai bobot digunakan sebagai inputan dalam proses klasifikasi.

3.5 *K-Nearest Neighbor* (KNN)

KNN merupakan algoritma klasifikasi *supervised learning*. KNN adalah salah satu algoritma sederhana dalam *machine learning* yang berbasis jarak [18]. Perhitungan jarak dengan metode KNN dilakukan dengan cara

menghitung jarak dari satu data yang berasal dari data *test* dengan seluruh data dari data *train* menggunakan *Euclidean Distance* [3]. Pada penelitian ini, pembuatan model dengan KNN dilakukan berdasarkan aspek. Setelah melalui proses ekstraksi fitur dengan TF-IDF dan seleksi fitur dengan IG, tahapan selanjutnya yaitu klasifikasi. Terdapat beberapa tahapan untuk menentukan kelas atau label pada data, yaitu:

1. Menentukan nilai *k*
2. Menghitung jarak terdekat dengan persamaan *Euclidean Distance*
3. Menjumlahkan hasil perhitungan jarak
4. Mengurutkan hasil penjumlah secara *ascending*
5. Memilih kategori tetangga terdekat sebanyak *k*
6. Memilih kelas berdasarkan frekuensi tertinggi terhadap *k* tertangga terdekat

Untuk menghitung jarak, dapat menggunakan persamaan *Euclidean Distance*[9]:

$$deuclid = \sqrt{\sum_{i=1} |P_i - Q_i|^2} \quad (3)$$

Keterangan:

- d_{euclid} : jumlah fitur atau dimensi
- P_i : fitur ke *i* pada data uji
- Q_i : fitur ke *i* pada data latih

3.6 Evaluasi

Menghitung kinerja dari model klasifikasi adalah cara untuk mengetahui efektivitas dari sistem yang dibangun [7]. Salah satu metrik untuk menghitung kinerja dan efektivitas sistem adalah *confusion matrix*. Hasil dari *confusion matrix* berupa nilai *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Evaluasi yang akan digunakan yaitu *accuracy*, *f1-score*, *precision*, dan *recall*. *F1-Score* adalah kelengkapan dan ketepatan dari model dilihat berdasarkan rata-rata label. Nilai *F1-Score* didapatkan dari nilai rata-rata *recall* dan *precision*. Untuk mendapatkan nilai *Recall*, dapat dihitung menggunakan formula sebagai berikut:

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

Sedangkan untuk mendapatkan nilai *Precision*, dapat diketahui dengan menggunakan formula sebagai berikut:

$$Precision = \frac{TP}{TP+FP} \quad (5)$$

Sehingga nilai *F1-Score* dapat diformulasikan sebagai berikut:

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

Sedangkan akurasi dipilih untuk melihat keakuratan data uji terhadap data latih. Perhitungan akurasi dirumuskan sebagai berikut:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (7)$$

4. Evaluasi

Dataset dibagi menjadi 80% sebagai data *train* dan 20% sebagai data *test* dengan persebaran kelas yang dapat dilihat pada Gambar 2. Hasil evaluasi didapatkan dari rata-rata akurasi dan *f1-score* tiap aspek. Nilai *f1-score* diambil dari *macro average* yang berasal dari *library* sklearn metrics. Penelitian ini berfokus pada 3 tahapan skenario pengujian, yaitu tahap praproses, seleksi fitur, dan tahap klasifikasi.

4.1 Evaluasi Tahap *Preprocessing*

Skenario pertama adalah mengetahui dan menganalisis pengaruh tahapan *preprocessing* pada *dataset female daily*. Pengaruh praproses pertama yang diuji adalah penerapan *stemming* dan tanpa *stemming* pada *dataset female daily*. Sistem pada penelitian ini dibangun menggunakan *library* Sastrawi untuk proses *stemming*. Penelitian ini menggunakan IG sebagai seleksi fitur dan KNN sebagai metode klasifikasinya. Berikut hasil akurasi dari percobaan:

Tabel 16. Hasil Akurasi Tahap *Preprocessing* dengan *Stemming*

nilai k	*threshold*	tanpa *stemming*	dengan *stemming*
k = 3	0,1	53,88%	63,07%
	0,2	60,26%	64,14%
	0,3	62,75%	63,54%
	0,4	64,17%	65,44%
	0,5	65,15%	65,62%
k = 5	0,1	59,60%	67,39%
	0,2	64,33%	69,07%
	0,3	65,37%	69,67%
	0,4	67,99%	68,94%
	0,5	68,72%	68,91%
k = 7	0,1	62,56%	68,75%
	0,2	67,87%	70,64%
	0,3	69,32%	69,48%
	0,4	70,27%	70,83%
	0,5	70,61%	70,77%
k = 9	0,1	64,20%	70,55%
	0,2	69,10%	71,28%
	0,3	70,86%	71,05%
	0,4	71,21%	71,15%
	0,5	71,37%	71,50%
k = 11	0,1	66,26%	71,56%

	0,2	70,64%	72,32%
	0,3	71,72%	72,66%
	0,4	72,19%	72,79%
	0,5	72,10%	**73,17%**
rata-rata		66,90%	69,37%

Berdasarkan performansi yang disajikan pada tabel 16, hasil akurasi tertinggi didapatkan dari dataset yang menerapkan *stemming* dengan *threshold* 0,5 yaitu sebesar 73,17%. Penerapan *stemming* pada *dataset female daily* terbukti lebih baik. Hal ini dikarenakan penggunaan algoritma Sastrawi menghapus sufiks maupun prefiks setiap kata atau mengubah kata menjadi kata dasar. Sehingga, kata dasar yang diperoleh dapat berguna untuk mencari atau mencocokan semua variasi kata yang terdapat pada suatu dokumen. Kemudian, kata-kata tersebut dapat menjadi fitur yang relevan yang nantinya digunakan sebagai inputan pada proses klasifikasi.

Tahapan *preprocessing* kedua yang akan diuji yaitu penggunaan normalisasi dan tanpa normalisasi. Sistem pada penelitian dibangun menggunakan IG sebagai seleksi fitur dan KNN sebagai metode klasifikasi. Normalisasi adalah proses untuk mengubah kata-kata yang mempunyai makna yang sama namun dituliskan dengan berbagai macam gaya tulisan. Sehingga, kata tersebut dianggap berbeda dan menyebabkan kesalahan klasifikasi.

Tabel 17. Hasil Prediksi dengan normalisasi

	teks ulasan	Harga	Aroma
normalisasi	...gentle cleanser lumayan mahal toner sih hiks	-1	0

Tabel 18. Hasil Prediksi tanpa normalisasi

	teks ulasan	Harga	Aroma
tanpa normalisasi	...gentle cleanser lumayan mehong toner sih hiks	1	0

Pada tabel 18, terdapat kata ”mehong” pada tabel tanpa normalisasi. Kata tersebut merupakan bahasa gaul atau *slang* yang memiliki makna ”mahal”. Akibatnya, kata ”mehong” diklasifikasikan salah karena kata tersebut dianggap memiliki arti yang berbeda dengan ”mahal”. Berikut hasil performansi dari penerapan normalisasi dan tanpa normalisasi:

Tabel 19. Hasil Akurasi Tahap *Preprocessing* dengan Normalisasi

nilai k	*threshold*	tanpa normalisasi	dengan normali-sasi
k = 3	0,1	57,99%	63,07%
	0,2	63,01%	64,14%
	0,3	65,47%	63,54%
	0,4	67,17%	65,44%
	0,5	66,92%	65,62%
k = 5	0,1	62,56%	67,39%

	0,2	67,01%	69,07%
	0,3	68,21%	69,67%
	0,4	67,65%	68,94%
	0,5	68,02%	68,91%
k = 7	0,1	65,78%	68,75%
	0,2	69,51%	70,64%
	0,3	69,79%	69,48%
	0,4	69,85%	70,83%
	0,5	69,85%	70,77%
k = 9	0,1	67,61%	70,55%
	0,2	70,52%	71,28%
	0,3	70,93%	71,05%
	0,4	71,12%	71,15%
	0,5	71,05%	71,50%
k = 11	0,1	68,75%	71,56%
	0,2	70,93%	72,32%
	0,3	71,28%	72,66%
	0,4	71,97%	72,79%
	0,5	71,97%	**73,17%**
average		68,20%	69,37%

Dapat dilihat pada tabel 19, penerapan normalisasi tertinggi berhasil mendapatkan akurasi sebesar 73,17% pada *threshold* dengan nilai *k* = 11. Hal ini terjadi karena proses normalisasi membantu proses seleksi fitur dalam menentukan fitur yang berelevan dengan kelasnya. Sehingga, dapat disimpulkan bahwa penerapan normalisasi dapat mempengaruhi hasil prediksi dari klasifikasi.

4.2 Evaluasi Tahap Seleksi Fitur

Pada pengujian skenario tahap *preprocessing*, hasil terbaik didapatkan oleh *dataset* yang menerapkan normalisasi dan *stemming*. Oleh karena itu, *dataset* tersebut digunakan untuk mencari nilai *threshold* yang terbaik. Untuk hasil pengujian dapat dilihat pada tabel 20.

Tabel 20. Hasil Akurasi Tahap Seleksi Fitur

threshold	Jumlah Fitur	k=3	k=5	k=7	k=9	k=11	rata-rata
0,1	301	63,07%	67,39%	68,75%	70,55%	71,56%	68,24%
0,2	148	64,14%	69,07%	70,64%	71,28%	72,32%	69,49%
0,3	93	63,54%	69,67%	69,48%	71,05%	72,66%	69,28%
0,4	49	65,44%	68,94%	70,83%	71,15%	72,79%	69,83%
0,5	25	65,62%	68,91%	70,77%	71,50%	**73,17%**	69,99%

Berdasarkan hasil pengujian di tabel 20, didapatkan hasil terbaik pada *threshold* 0,5 yang menggunakan nilai *k* = 11. Hal ini membuktikan bahwa semakin tinggi nilai *threshold*, maka ketepatan data *test* terhadap data *train* semakin baik. Namun, hasil akurasi masih belum maksimal karena hanya ditemukan sebanyak 25 dari 490 fitur yang terpilih pada percobaan *threshold* 0,5. Hal ini dikarenakan jumlah *dataset* yang sedikit menghasilkan jumlah fitur atau *term* sedikit.

4.3 Evaluasi Tahap Klasifikasi

Dari hasil skenario sebelumnya, akurasi tertinggi didapatkan oleh *dataset* yang menggunakan *stemming*, normalisasi, dan nilai IG lebih dari 0,5. Sistem dibangun menggunakan KNN sebagai metode klasifikasi. Tahapan klasifikasi digunakan untuk menganalisis dan mengetahui nilai *k* yang optimal pada KNN. Berdasarkan *elbow method*, nilai *k* yang optimal didapatkan pada rentang 3 sampai dengan 25. Nilai *k* merupakan representasi dari kedekatan antar data. Perhitungan akurasi model didapatkan dari akurasi rata-rata tiap aspek. Berikut hasil performansi yang didapatkan dari pengujian:

Tabel 21. Hasil Akurasi Tahap Klasifikasi

Nilai *k*	Akurasi
k=3	65,62%
k=5	68,91%
k=7	70,77%
k=9	71,50%
k=11	73,17%
k=13	73,77%
k=15	73,64%
k=17	73,23%
k=19	73,58%
k=21	73,93%
k=23	**74,21%**
k=25	73,96%

Nilai *k* optimal didapatkan adalah 23 dimana nilai akurasi yang diperoleh sebesar 74,21% artinya sebanyak 586 dari 792 data terklasifikasi dengan benar. Hasil percobaan menunjukkan bahwa semakin tinggi nilai *k* yang digunakan, maka akurasi yang didapatkan juga meningkat. Namun, saat model sudah mendapatkan hasil akurasi yang optimal, akurasi selanjutnya cenderung menurun. Hal ini dikarenakan nilai *k* yang semakin besar maka jumlah tetangga yang digunakan semakin banyak. Sehingga, kemungkinan data yang *noise* juga semakin tinggi.

Aspek *Price* mendapatkan performansi *f1-score* paling tinggi yaitu 50,66%. Hasil ini dikarenakan distribusi label pada aspek *price* lebih seimbang daripada aspek lainnya. Aspek *price* memiliki label netral sebanyak 451, positif sebesar 228, dan label negatif sebanyak 113. Pada kelas *packaging* dan aroma mendapatkan *f1-score* yang rendah karena label netral merupakan label terbanyak daripada label positif maupun negatif pada *dataset*. Hal ini menyebabkan banyaknya data yang diprediksi netral daripada positif maupun negatif.

Hasil akurasi dan *f1-score* dari tiap aspek berkebalikan, dimana akurasi yang tinggi namun *f1-score* rendah. Hal ini dapat terjadi karena akurasi hanya dapat bekerja dengan baik apabila jumlah label persebarannya seimbang. Sedangkan *f1-score*, bekerja dengan cara menghitung kelengkapan dan ketepatan model dalam memprediksi data. Dapat disimpulkan bahwa penentuan performansi paling baik untuk data yang tidak seimbang adalah *f1-score*.

Tabel 22. Performansi tiap Aspek

Aspek	Akurasi	*F1-Score*
Price	64,90%	50,66%
Packaging	83,96%	41,42%
Product	69,82%	48,99%
Aroma	78,16%	39,70%
Average	74,21%	45,19%

Tabel 23. Akurasi tiap Aspek Data *Train* Data *Test*

Aspect	Data *Train*	Data *Test*
Price	64,61%	64,90%
Packaging	84,97%	83,96%
Product	70,90%	69,82%
Aroma	79,29%	78,16%
Average	74,94%	74,21%

Berdasarkan tabel 23, akurasi rata-rata pada data *train* sebesar 74,94% dan data *test* sebesar 74,21% dimana selisih akurasinya hanya sebesar 0,73%. Hal ini membuktikan bahwa model yang dibangun tidak menimbulkan *overfitting* karena model dapat memprediksi dengan baik baik pada data *train* maupun data *test*. Dikarenakan model tidak *overfitting*, maka tidak dilakukan proses penanganan data yang tidak seimbang atau *imbalanced data*.

5. Kesimpulan

Berdasarkan hasil pengujian dari keseluruhan skenario, dapat disimpulkan bahwa data yang tidak seimbang mendapatkan akurasi paling tinggi karena model cenderung memprediksi label mayoritas. Pada tahapan *preprocessing*, penerapan *stemming* dan normalisasi terbukti memberikan pengaruh yang signifikan karena dapat mengurangi kata yang memiliki satu makna tetapi mempunyai lebih dari satu kata atau *term*. Kemudian, pada seleksi fitur, nilai IG lebih dari 0,5 mempunyai tingkat relevansi yang tinggi terhadap kelasnya. Nilai *k* yang optimal diperoleh sebesar 23. Sehingga, dari ketiga skenario yang dilakukan, akurasi tertinggi diperoleh oleh *dataset* yang menerapkan *stemming*, normalisasi, IG dengan *threshold* 0,5, dan $k = 23$ dengan akurasi sebesar 74,21%.

Saran untuk penelitian selanjutnya adalah menambahkan kamus kata untuk normalisasi dan menambahkan jumlah *dataset*.

Referensi

[1] Mutual information. https://nlp.stanford.edu/IR-book/html/htmledition/ mutual-information-1.html. Accessed: 2021-07-12.

[2] C. H. Y. Adiwijaya, Said Al Faraby. Analisis sentimen berbasis aspek pada review female daily menggunakan tf-idf dan na¨ıve bayes. volume 5, pages 1–9, April 2021.

[3] N. O. F. D. Adiwijaya. Sentiment analysis on movie reviews using information gain and k-nearest neighbor. *Journal of Data Science and Its Applications*, 3(1):1–7, 2020.

[4] N. P. A. Adiwijaya, Mahendra D P. Aspect-based sentiment analysis in beauty product reviews using tf-idf and svm algorithm. *ICoICT 2021 The 9th International Conference on Information and Communication Technology*, pages 1–6, 2020.

[5] C. C. Aggarwal. *Opinion Mining and Sentiment Analysis*, pages 413–434. Springer International Publishing, Cham, 2018.

[6] W. A. Alfath Noverio, Adiwijaya. Sentiment analysis pada movie review menggunakan feature selection information gain dan decision tree classifier. pages 1–18, 2021.

[7] M. Bakar, K. Adiwijaya, and S. Faraby. Multi-label topic classification of hadith of bukhari (indonesian language translation)using information gain and backpropagation neural network. pages 344–350, 11 2018.

[8] B. Bickart and R. Schindler. Internet forums as influential sources of consumer information. *Journal of Interactive Marketing*, 15:31 – 40, 06 2001.

[9] S.-H. Cha. Comprehensive survey on distance/similarity measures between probability density functions. *Int. J. Math. Model. Meth. Appl. Sci.*, 1, 01 2007.

[10] M. Fauzi. Word2vec model for sentiment analysis of product reviews in indonesian language. *International Journal of Electrical and Computer Engineering (IJECE)*, 9:525, 07 2018.

[11] D. Hidayati, S. Faraby, and A. Adiwijaya. Klasifikasi topik multi label pada hadis shahih bukhari menggunakan k-nearest neighbor dan latent semantic analysis. *JURIKOM (Jurnal Riset Komputer)*, 7:140, 02 2020.

[12] M. S. Mubarok, Adiwijaya, and M. D. Aldhi. Aspect-based sentiment analysis to review products using na¨ıve bayes. In *AIP Conference Proceedings*, volume 1867, page 020060. AIP Publishing LLC, 2017.

[13] J.-C. Na, H. Sui, C. Khoo, S. Chan, and Y. Zhou. Effectiveness of simple linguistic processing in automatic sentiment classification of product reviews. In *Conference of the International Society for Knowledge Organization (ISKO)*, pages 49–54, 2004.

[14] M. S. Neethu and R. Rajasree. Sentiment analysis in twitter using machine learning techniques. In *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, pages 1–5, 2013.

[15] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. *ArXiv*, cs.CL/0205070, 2002.

[16] A. I. Pratiwi et al. On the feature selection and classification based on information gain for document sentiment analysis. *Applied Computational Intelligence and Soft Computing*, 2018, 2018.

[17] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34:1–47, 04 2001.

[18] P. Soucy and G. W. Mineau. A simple knn algorithm for text categorization. In *Proceedings 2001 IEEE International Conference on Data Mining*, pages 647–648. IEEE, 2001.

[19] D. M. D. Sunitha C, Amal Ganesh. Sentiment analysis:a comparative study on different approaches. pages 1–6, 10 2012.

[20] J. L. B. W. C. S. Yan Xu, Gareth Jones. A study on mutual information-based feature selection for text categorization. *Journal of Computational Information Systems 3:3 (2007) 1007-1012*, pages 1–6, 2007.

Lampiran

Lampiran dapat berupa detil data dan contoh lebih lengkapnya, data-data pendukung, detail hasil pengujian, analisis hasil pengujian, detail hasil survey, surat pernyataan dari tempat studi kasus, screenshot tampilan sistem, hasil kuesioner dan lain-lain.