Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN: 2548-964X
Vol. 4, No. 11, November 2020, hlm. 4154-4163 http://j-ptiik.ub.ac.id

Analisis Sentimen Ulasan Aplikasi *Mobile* menggunakan Algoritma Gabungan Naïve Bayes dan C4.5 berbasis Normalisasi Kata Levenshtein Distance

Arrofi Reza Satria[1], Sigit Adinugroho[2], Suprapto[3]

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: [1]arrofirezasatria@gmail.com, [2]sigit.adinu@ub.ac.id, [3]spttif@ub.ac.id

Abstrak

Google Play Store telah menjadi tempat pasar digital terbesar dengan lebih dari 10 juta produk didalamnya. Para pengembang aplikasi menjadikan kolom ulasan produk yang tersedia di Google Play Store menjadi salah satu cara untuk mengetahui kepuasan pengguna. Tapi tidak semua ulasan di aplikasi memiliki keselarasan antara rating dengan komentar, terdapat ulasan yang ambigu, yaitu ulasan yang ditandai oleh rating dan sentimen komentarnya tidak sama. Machine Learning (ML) telah sangat berguna dalam bidang analisis sentimen. Salah satu metode yang handal dan mudah digunakan adalah Naive Bayes. Metode C4.5 juga sangat populer dalam menyelesaikan permasalahan decesion tree yang dimana akan digunakan untuk proses klasifikasi sentimen. Sedangkan metode Levenshtein Disance digunakan untuk membandingkan antara dua buah string untuk proses normalisasi kata. Alur metode dimulai dengan memproses teks awal dataset dengan Levenstein Distance, kemudian dataset akan dibagi dua untuk proses klasifikasi Naïve Bayes dan C4.5. Dataset beratribut sentimen teks dan teks ulasan akan diproses oleh metode Naïve Bayes sedangkan rating dan sentiment teks akan diproses oleh C4.5. Hasil pengujian dengan metode evaluasi 10-Fold adalah 85,3%. Sedangkan klasifikasi sentimen tanpa menggunakan Levenshtein Distance adalah 85,6% selisih 0,3% menjadikan metode Levenshtein Distance tidak begitu signifikan mempengaruhi hasil klasifikasi. Hasil pengujian lainnya dengan penerapan batas limit Edit Distance 1, 2, 3 dan 4 masing-masing adalah 86,9%., 85,9%., 87,1% dan 86,1%. Pengujian algoritma Naïve Bayes tanpa C4.5 dalam mengklasifikasi teks ulasan memiliki hasil 85,3% selaras dengan pengujian sebelumnya. Hasil pengujian ini menggambarkan efektifitas program ini dalam klasifikasi sentiment aplikasi mobile.

Kata kunci: *analisis sentimen, aplikasi mobile, Naive Bayes, C45, Levenstein Distance*

Abstract

Google Play Store has become the largest digital market place with more than 10 million products in it. Application developers make the product review column available on the Google Play Store one of the ways to find out user satisfaction. But not all reviews on the app have an alignment between ratings and comments, there are ambiguous reviews, that are reviews marked by ratings and sentiment of comments are not the same. Machine Learning (ML) has been very useful in the field of sentiment analysis. One method that is reliable and easy to use is Naive Bayes. C4.5 method is also very popular in solving the decesion tree problem which will be used for the sentiment classification process. While the Levenshtein Disance method is used to compare two strings for the word normalization process. The method flow start with text preprocessing dataset with Levenstein Distance, then the dataset will be divided into two for the Naïve Bayes and C4.5 classification process. The sentiment text and text review will be processed by the Naïve Bayes method while the rating and sentiment text will be processed by C4.5.The test results using the 10-Fold evaluation method are 85.3%. While the sentiment classification without using Levenshtein Distance is 85.6%, the difference is 0,3%, making the Levenshtein Distance method not significantly affect the classification results. Other test results with the application of the Edit Distance 1, 2, 3 and 4 limits were 86.9%, 85.9%, 87.1% and 86.1%, respectively. Testing Naïve Bayes algorithm without C4.5 in classifying review texts has an 85.3% same result with previous test. The results of this test illustrate the effectiveness of this program in the classification of mobile application

Keywords: *sentiment analysis, mobile applications, Naive Bayes, C4.5, Levenstein Distance*

1. PENDAHULUAN

Google Play Store telah menjadi tempat pasar digital terbesar dengan lebih dari 10 juta produk didalamnya. Terdapat sangat banyak sekali media digital yang tersedia seperti aplikasi, ebook dan film. Salah satu penyumbang pembelian terbesar media digital adalah produk aplikasi mobile. Mengawali penjualan aplikasi dengan harga gratis merupakan salah satu sebab mengapa aplikasi mobile banyak diunduh. Seperti yang dilansir dari penelitian (AppBrain, 2019), 90% Aplikasi mobile dimulai dengan harga gratis dan lebih dari 90% keuntungan dari aplikasi mobile didapatkan dari aplikasi yang berlabelkan gratis. Sejauh ini lebih dari 76 milyar aplikasi di download pada tahun 2018 dan akan terus meningkat ditahun selanjutnya (Statista, 2018). Aplikasi mobile yang paling banyak diunduh adalah pada bidang sosial media, permainan dan produktivitas. Ketiga bidang ini diestimasikan mendapat keuntungan dari penjualan aplikasi sekitar $90 milyar di tahun 2018 (Freier, 2018), melampaui penjualan game konsol dan pc.

Pasar aplikasi mobile yang menjanjikan, membuat para developer berlomba-lomba untuk membuat aplikasi yang diminati oleh khalayak luas. Industri perangkat mobile harus memberikan pengalaman terbaik bagi pengguna aplikasi dengan cara meningkatkan kualitas dan pelayanan, Lebih jauh lagi mengetahui kebutuhan pengguna. Memperbanyak produk aplikasi yang di rilis juga salah satu bagian dari memperbanyak keuntungan industri aplikasi. Di sisi lain, masyarakat pengguna juga ikut terbantu dengan adanya aplikasi media digital.

Kolom ulasan produk yang tersedia di Google Play Store merupakan salah satu cara untuk mengetahui kepuasan pengguna. Ulasan yang tertulis bisa diketahui sentimennya melalui rating atau komentar. Sentimen adalah pengutaraan opini, sikap dan emosi seseorang terhadap suatu objek tertentu (Medhat, Hassan & Korashy, 2014). Ulasan tediri dari dua atribut, yaitu rating dan teks komentar pengguna. Contoh ulasan di Google Play Store (https://play.google.com) yang terdiri dari komentar dan jumlah bintang. Ulasan di steam game (https://store.steampowered.com) berupa rekomendasi biner dan teks komentar.

Tapi sayangnya, tidak seluruhnya ulasan aplikasi selaras rating dengan komentar. Terdapat ulasan yang ambigu, yaitu rating dan sentimen ulasannya tidak sama. Dalam hal ini ulasan memiliki dua arti sentimen, bisa positif dan negatif. Hal ini bisa menjadi masalah ketika pihak developer kesulitan untuk mengintrepertasi sentimen dari ulasan tersebut. Terlebih lagi sentimen yang tergolong campuran tidak bisa dihiruakan dikarenakan untuk menjaga agar informasi penting dari konsumen tidak hilang. Masalah ini diperlukan teks analisis dengan aturan tertentu untuk mendeteksi ulasan yang tergolong campuran.

Machine Learning (ML) telah sangat berguna khususnya dalam bidang klasifikasi, seperti analisis sentimen, klasifikasi dokumen dan deteksi spam. Salah satu metode yang handal dan mudah digunakan adalah Naive Bayes (NB), Terbukti dari beberapa penelitian sebelumnya oleh Sutabri, et al. (2018). Olbenjo, B. (2016) dan Febriyani, Nasrun & Setianingsih (2018). Ketiga penelitian tersebut meneliti mengenai bidang text mining menggunakan metode Naive Bayes dengan waktu penelitian kurang dari lima tahun yang lalu. Hasil dari ketiga penelitian tersebut memiliki nilai akurasi terbaik diatas 80 % yang dimana sesuai dengan permasalahan dan dataset yang tersedia dalam penelitian tersebut. Dengan nilai akurasi yang cukup tinggi, membuktikan bahwa metode Naive Bayes handal digunakan dalam bidang text mining.

Dalam bidang teks mining, terdapat beberapa variasi dan gabungan penggunaan metode klasifikasi. Salah satunya adalah metode gabungan Rule-Based dan Machine Learning seperti yang diusulkan oleh Chikersal, Poria & Cambria (2015) denga penelitiannya berjudul "SeNTU: Sentimen Analysis of Tweets by Combining a Rule-based Classifier with Supervised Learning". Penelitian ini menerapkan dua langkah klasifikasi. Jika klasifikasi dari Mahine Learning berkategori unknown maka akan memasuki klasifikasi secara Rule-Based. Penambahan metode Rule-Based memungkinkan untuk mengklasifikasikan kategori yang tidak bisa didefinisikan oleh machine learning. Penelitian ini juga menjadi acuan penulis yang juga sama menerapkan dua langkah klasifikasi.

Menangani permasalahan teks mining yang berhubungan dengan opini publik, sangat erat sekali terjadi kesalahan penulisan teks. Hal ini bisa menyebabkan sistem salah mengidentifikasi teks yang berakibat ketidak akuratan kategorisasi kelas data. Penerapan metode pencocokan dan pengecekan dalam kata sangat penting dalam menangani data yang terdapat banyak kesalahan eja. Salah satu metode yang banyak digunakan dalam masalah perbaikan kesalahan eja dalam kata adalah Levenshtein Distance. Penelitian sebelumnya yang menggunakan metode perbaikan kata ini pernah digunakan oleh Gunawan, Fauzi & Adikara (2018). Dalam penelitian ini membuktikan bahwa metode Levenshtein Distance bisa meningkatkan akurasi klasifikasi sebesar 2.5%.

Penelitian ini akan dilakukan klasifikasi dengan kategori sentimen positif, negatif dan campuran (mixed) dengan menggunakan metode gabungan Naive Bayes dan C4.5 (NB-C4.5). Sedangkan metode Levenshtein Distance sebagai metode perbaikan kesalahan eja dalam kata. Dataset ulasan aplikasi mobile akan dipilih dari empat kategori aplikasi mobile yang berbeda dengan jumlah unduhan diatas satu juta. Layaknya API pada Twitter dataset, Peneliti mengambil dataset sentimen ulasan dari API yang disediakan dari pihak ketiga, yaitu Appbot (https://appbot.co), seperti yang dilakukan oleh Bano, Zowghi & Kearney (2017). Algoritma Naive Bayes pada penelitian ini akan bertindak sebagai algoritma yang akan melakukan klasifikasi sentimen, khususnya sentimen pada teks komentar ulasan. Algoritma C4.5 sebagai klasifikasi sentimen keseluruhan antara nilai rating dan sentimen teks. Sedangkan algoritma Levenshtein Distance sebagai metode untuk memperbaiki kesalahan eja dalam kata.

Berdasarkan hipotesis diatas, peneliti mencoba untuk merealisasikannya bersamaan dikerjakannya penelitian ini yang berjudul "Analisis Sentimen Aplikasi Mobile Menggunakan Gabungan Naïve Bayes dan C4.5 Berbasis Normalisasi Kata Levenshtein Distance". Diharapkan dengan penelitian ini bisa membantu pihak developer dalam meningkatkan kualitas aplikasi.

2. TINJAUAN PUSTAKA

2.1 Analisis Sentimen

Sentiment Analyisis, *Sentiment Classifiation* (SC) atau *Opinion Mining* (OM) adalah salah satu studi untuk memahami opini, sikap dan emosi seseorang terhadap suatu objek tertentu (*entity*) (Medhat, Hassan & Korashy, 2014). Tidak jauh berbeda dengan definisi Sentimen menurut kamus bahasa Indonesia, yaitu "pendapat atau pandangan yang didasarkan pada perasaan yang berlebih-lebihan terhadap sesuatu bertentangan dengan pertimbangan pikiran" (Kemdikbud, 2019). Objek Sentimen bisa seperti produk, pelayanan, seseorang, peristiwa, organisasi atau topik tertentu.

2.2 Dataset Ulasan Komersial Appbot

Appbot (https://appbot.co/) adalah sebuah website sistem analisis yang disediakan untuk menganalisis ulasan aplikasi *mobile*. Kelebihan dari Appbot adalah menyediakan fitur teks analisis seperti analisis sentimen, NLP (*Natural Language Processing)* dan *Apps Monitoring*. Sejak Google Play Store tidak merilis *official* API (*Application Programming Interface*) untuk diluar pengembang aplikasi. Appbot adalah rujukan bagi penganalisis sentimen dengan kelebihan 14 hari gratis masa percobaan. Dengan algoritma yang sudah di latih dari 400 juta rekaman, Appbot menjadi bahan yang konkrit untuk mendapatkan *supervised* data seperti pada penelitian oleh Bano, Zowghi & Kearney (2017).

2.3 Rating dan Ulasan

Rating merupakan pandangan sekilas dari kualitas, kuantitas dan aspek lainnya sebuah produk. Umumnya rating berupa angka atau banyaknya lambang tertentu. Pada Google Play Store rating di representasikan dengan lambang bintang berjumlah satu sampai lima. Ulasan adalah Evaluasi pendapat pengguna aplikasi yang dituliskan berupa komentar penjelasan dan saran mengenai mengapa pengguna memberikan nilai rating tersebut.

2.4 Pemrosesan teks awal

Pemrosesan teks awal merupakan teknik Data Mining untuk mengolah data agar bisa dengan mudah di proses oleh sistem kecerdasan buatan. Beberapa teknik yang biasanya digunakan adalah *Case Folding*, Tokenizing, Hapus Huruf Berulang, Stopword *Removal* dan Stemming. Teknik pemrosesan teks awal yang perlu ditekankan dalam bidang analis sentimen adalah Negation Handling.

Negation Handling termasuk bidang pemrosesan teks awal yang menangani penambahan kata negasi, sehingga bisa mempertahankan sentimen asli katanya. Kata-kata seperti "tidak" akan sangat berpengaruh

pada klasifikasi analisis sentimen. Maka dari itu diperlukanan penanganan sederhana terhadap kalimat negasi dengan cara dilakukan penggabungan kata negatif dengan kata selanjutnya. Kata negatif seperti "tidak", "enggak" dan "gk" akan dirubah menjadi "negatif", Hal ini berguna untuk menyamakan kata negasi. Kemudian, penggabungan kata negasi dengan kata selanjutnya contoh seperti "tidak bisa" menjadi "negative_bisa".

2.5 Levenshtein Distance

Levenshtein Distance atau Edit Distance adalah matriks perbandingan untuk mengukur perbedaan diantara dua urutan oleh Vladimir Levenshtein (1966). Levenshtein Distance sering dipakai dalam membandingankan antara dua urutan String yang berguna untuk masalah memperbaiki kesalahan eja dalam kata. Secara rumus matematika Levenshtein Distance bisa dianotasikan seperti persamaan 1.
Keterangan variabel dan fungsi :

i,j : Angka iterasi posisi huruf kata
a,b : String kata
$lev_{a,b}(i,j)$:Matriks perbandingan dua String
$max(i,j)$: Memilih angka terbesar dari i dan j
$min(i,j)$:Memilih angka terkecil dari i dan j

Ada tiga macam operasi yang digunakan oleh algoritma ini yaitu :

1. Operasi Penggantian (*Subtitution*) aaaa
   Operasi ini menukar karakter terhadap kata yang diindikasikan terdapat kesalahan eja.
2. Operasi Penambahan (*Insertion*) aaaa
   Operasi ini menambah karakter pada huruf pada kata yang diindikasikan terdapat kesalahan eja.
3. Operasi Penghapusan (*Deletion*) aaaa
   Operasi ini menghapus karakter pada huruf pada kata yang diindikasikan terdapat kesalahan eja.

$$lev_{a,b}(i,j) = \begin{cases} max(i,j) \ \ if\ min(i,j) = 0 \\ min \begin{cases} lev_{a,b}(i-1,j) + 1 \\ lev_{a,b}(i,j-1) + 1 \\ lev_{a,b}(i-1,j-1) + 1_{a_i \neq b_j} \end{cases} \quad otherwise \end{cases} \tag{1}$$

2.6 Naïve Bayes

Naive Bayes adalah metode simpel dan paling banyak digunakan untuk proses klasifikasi. Klasifikasi Naive Bayes adalah berdasarkan Bayes Theorem. Teori Bayes menjelaskan probabilitas dan kejadian berdasarkan kondisi dari suatu kejadian tertentu . Teori Bayes didefinisikan sebagi berikut (Rish, 2001).

$$P(c_j|w_i) = \frac{P(c_j)\ x\ P(w_i|c_j)}{P(w_i)} \tag{2}$$

Keterangan :

$P(c_j|w_i)$:Posterior merupakan peluang kategori j ketika terdapat kemunculan kata i
$P(w_i|c_j)$:Conditional probability merupakan peluang sebuah kata i masuk ke dalam .kategori j
$P(c_j)$:Prior merupakan peluang kemunculan sebuah kategori j
$P(w_i)$:Peluang kemunculan sebuah kata

Navie Bayes Classifier (NBC) telah diterapkan di berbagai macam bidang klasifikasi dokumen, spam email dan analisis sentimen. Naive Bayes sangat diketahui akan kecepatan dan membutuhkan sedikit data untuk proses klasifikasi. Sejak klasifikasi dilakukan dalam proses komputasi, maka Naive Bayes adalah metode yang paling efisien dalam penggunaan memori.

2.7 C4.5

Algoritma C4.5 diusulkan oleh Quinlan, J. R. (2014) dalam bukunya berjudul "*C4. 5: programs for machine learning*" dalam memperbaiki algoritma pohon keputusan sebelumnya ID3. Walapun begitu, dasar algoritma ID3 dan C4.5 masih memiliki kesamaan. Terdapat beberapa rumus penting dalam membentuk pohon keputusan C4.5.

$$entropy(T) = \sum_{i=1}^{n} \left(\frac{T_i}{T}\right) \times Info(T_i) \tag{3}$$

Entropy adalah perhitungan untuk mengetahui ketidak teraturan data dalam dataset.

entropy(T) atau entropy dijelaskan pada persamaan 3, dimana T adalah data latih dan T_i adalah subset data latih yang di partisi berdasarkan atribut X.

$$Gain(T,x) = entropy(T) - entropy(T,x) \quad (4)$$

$$split\ entropy(rating) = -\sum_{i=0}^{n}\left(\frac{|T_i|}{|T|} x \log_2 \frac{|T_i|}{|T|}\right) (5)$$

Information Gain adalah nilai seberapa penting atribut pada vector dataset. Persamaan 4. ini, digunakan untuk memilih urutan atribut pada pohon keputusan C4.5. Information Gain didapatkan dari pengurangan entropy sebelum dipartisi atribut X dan entropy sesudah dipartisi varibel X. Split Entropy mempresentasikan potensi informasi yang didapat dari data T ke banyaknya n subset. Persamaan 6 membagi subset kasus terhadap seluruh banyknya data.

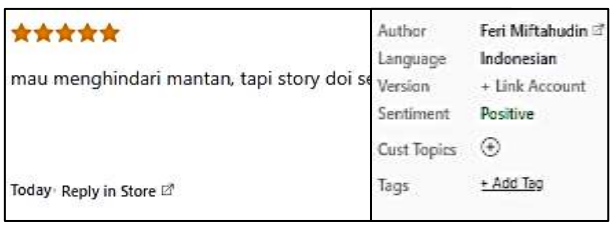
$$gain\ ratio(\mathrm{X}) = \frac{gain(X)}{split\ entropy(X)} \quad (6)$$

Gain ratio memperbaiki kekurangan dari information gain dengan cara membaginya dengan probabilitas banyaknya kasus. Perhitungan gain ratio diperlukan nilai information gain dan split entropy.

3. METODOLOGI PENELITIAN

3.1 Pengumpulan dan Pengolahan Data

Data analisis sentimen yang diambil didapatkan dari penyedia layanan teks analisis komersial Appbot (https://app.appbot.co). Data yang tersedia bisa diambil secara gratis dengan memanfaatkan masa percobaan 14 hari. Pengambilan data dari *website* tersebut dibantu dengan *extension web scraper* yang tersedia di Chrome secara umum dan gratis. Sebagai contoh, data ulasan yang diambil dari aplikasi Instagram dengan filter bahasa Indonesia yang bisa dilihat pada Gambar 1.



Gambar 1. Ulasan Appbot

Penjelasan lebih lanjut mengenai bagian data yang diambil dari Gambar 1 dijelaskan sebagai berikut:

1. Rating (X1) : Diindikasikan bintang 1-5
2. Teks (X3) : Teks ulasan
3. Sentimen (Y)..:.Sentimen keseluruhan ulasan (Positif, Negatif dan *Mixed*)

Sesuai dengan kebutuhan penelitian, data sentimen teks (X2) yang kosong akan diisi oleh penulis secara manual oleh peneliti dengan dua kategori sentimen positif dan negatif seperti pada Tabel 1.

Tabel 1. Dataset Ulasan Setelah Pengisian Manual Sentimen Teks

Y Sentimen	X1 Rat1ing	X2 Sentimen Teks	X3 Teks Ulasan
Positif	5	T_Negatif	Mau …
Positif	5	T_Positif	Instagram …
Negatif	1	T_Negatif	Habis di …
Mixed	5	T_Negatif	Di Instagram…
-	Rating uji	-	Contoh ulasan uji

Dateset ulasan akan di ditransformasikan menjadi dua dataset, yakni untuk dataset klasifikasi Naïve Bayes dan C4.5. Variabel sentimen teks (X2) akan menjadi variabel yang digunakan pada dua dataset, yang masing masing sebagai variabel independen pada klasifikasi C4.5 dan menja1di variabel dependen atau Y pada klasifikasi Naïve Bayes seperti pada Tabel 2. Pemecahan dataset juga termasuk data latih dan data uji.

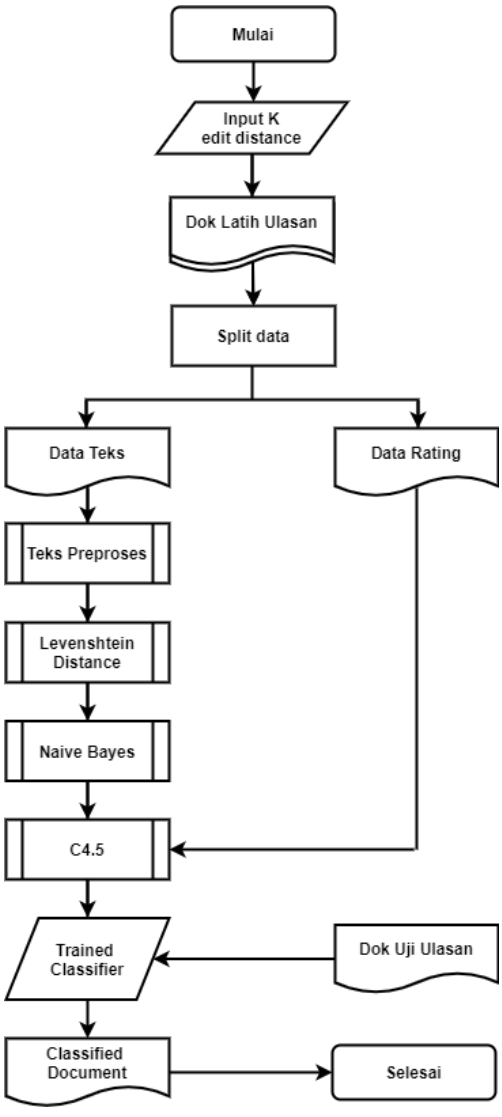
3.2 Alur Proses Klasifikasi Sistem Sentimen Analisis

Klasifikasi sentimen pada aplikasi Google Play Store dilakukan dengan menggunakan metode Naive Bayes dan C4.5, Sedangkan metode Levenshtein Distance sebagai metode pendukung pemrosesan teks awal. Sesuai Gambar 2, klasifikasi sentimen dimulai dengan membagi data latih menjadi dua melalui proses *split* data, teks dan rating. Data teks yang berisikan sentimen teks (X2) dan teks ulasan (X3) akan memasuki proses teks pemrosesan awal, Naïve Bayes dan Levenshtein Distance. Pelatihan dengan metode NB ini bertujuan untuk mengetahui sentimen teks ulasan. Selanjutnya, data rating akan memasuki proses klasifikasi

Tabel 2. Dataset Ulasan untuk Klasifikasi Naïve Bayes dan C4.5

C4.5			Naive Bayes	
Y	**X1**	**X2**	**X2**	**X3**
Positif	5	T_Negatif	T_Negatif	Mau menghindari mantan, tapi story doi selalu paling kiri
Positif	5	T_Positif	T_Positif	Instagram memang bagus dipakai
Negatif	1	T_Negatif	T_Negatif	Habis di update kok jadi gabisa bikin snap shot
Mixed	5	T_Negatif	T_Negatif	Di Instagram saya masa ga ada emoji2 gitu buat kirim pesan
-	Rating uji	-	-	Contoh teks ulasan uji

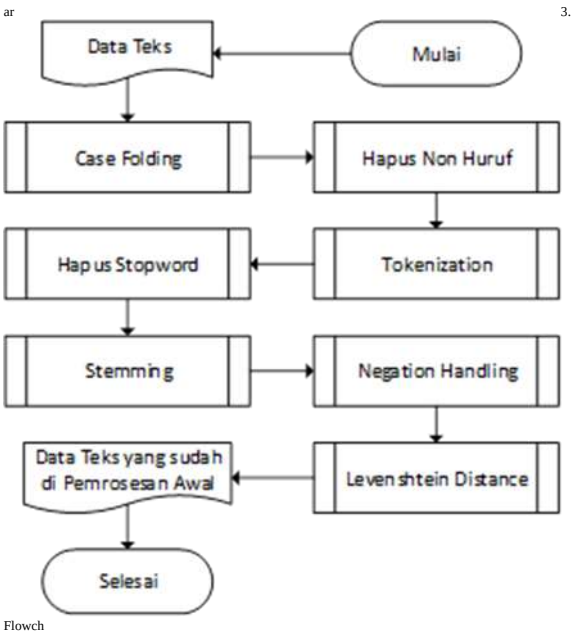
C45 untuk mengetahui aturan hubungan antara rating dengan sentimen komentar.



Gambar 3. Flowchart Tahapan Pemrosesan Teks Awal

3.3 Pemrosesan Teks Awal

Data teks akan diolah terlebih dahulu sebelum masuk ke proses *Machine Learning*. Perancangan tahapan pemrosesan teks awal akan disimulasikan pada Gambar 3. Proses penghapusan stopword atau kata yang tidak penting membutuhkan daftar kata, Peneliti menggunakan dataset kata stopword dari Tala, F.Z. (2003). Proses stemming membutuhkan algoritma untuk menghapus kata yang berimbuhan, peneliti peneliti menggunakan library Java dengan algoritma oleh Nazief B. dan Adriani M. (1996) yang tersdia di https://github.com/jsastrawi/jsastrawi. Gamb ar 3. Flowch



Gambar 3. Flowchart Tahapan Pemrosesan Teks Awal

Proses Negation Handling juga membutuhkan kata-kata yang berindikasikan kalimat negasi, berikut adalah daftar kata negasi yang peneliti buat.

Kata negasi : {tidak, tdk, gk, gak, ga, engga, enggak, tak}

3.4 Normalisasi Kata menggunakan Levenshtein Distance

Peneliti menggunakan teknik *brute force* untuk mendapatkan kata yang memiliki nilai kemiripan yang sesuai. Proses Levenshtein Distance akan dilakukan terhadap seluruh kamus

yang akan menghasilkan daftar kata yang mendekati sama. Sebagai contoh, kata "asik" akan dibandingankan dengan seluruh kata pada kamus kata dasar yang nantinya akan didapatkan daftar kata yang bisa memperbaiki kata yang salah eja.

Edit 0 =.[-]
Edit 1 =.[acik, adik, akik, alik, apik, arik, asak, …asi, .asid, asih, asil, asin, asyik, fasik, …pasik, .tasik, usik]
Edit 2 =.[abid, abis, absis, abuk, dan 283 …kata .lainnya]
Edit 3 =.[aba, abad, abah, abai, dan 1857 kata ...lainnya]

Semakin tinggi nilai edit maka daftar kata normalisasi juga akan semakin banyak. Dengan nilai minimum nilai edit **1** maka terdapat 17 kata normalisasi yang nantinya akan menjadi pengganti kata yang salah eja.

- **Memilih kandidat pengganti Normalisasi Kata Levenshtein Distance**

Jika kandidat kata pengganti lebih dari satu, maka cara untuk memilih kata pengganti dilakukan dengan cara membuat probabilitas sederhana kata berdasarkan banyaknya kata dalam dataset. Tujuan dari proses ini untuk meminimalisir kesalahan pemilihan kata pengganti. Contoh, kata "asyik" terdapat banyak dalam dataset, sedangkan kata "apik" sedikit dalam dataset maka kandidat yang terpilih adalah kata "asyik". Jika probabilitas seluruh kandidat sama, maka bisa diambil kandidat pertama.

3.5 Klasifikasi Naïve Bayes

Proses ini bertujuan untuk mengklasifikasi sentimen ulasan teks (X2) dari teks ulasan (X3) dengan dua tipe nilai, T_Positif dan T_Negatif. Sistem nantinya akan menyimpan probabilitas kondisional setiap kata dalam data latih yang nantinya akan digunakan untuk proses klasifikasi pengujian. Hasil klasifikasi ini nantinya akan digunakan kedalam proses klasifikasi C4.5. Jika sistem mendapatkan nilai probabilitas yang sama, maka sistem akan memilihnya secara acak dari dua tipe nilai.

3.6 Klasifikasi C4.5

Proses ini bertujuan untuk mengklasifikasi sentimen keseluruhan (Y) dari atribut *rating* (X1) dan sentimen teks (X2). Data sentiment teks didapatkan dari hasil klasifikasi Naïve Bayes sebelumnya. Hasil klasifikasi terdapat tiga tipe nilai Positif, Negatif dan *Mixed*. Sistem nantinya akan membentuk *decision tree* dari hasil pelatihan yang nantinya akan digunakan untuk proses klasifikasi pengujian.

4. PENGUJIAN

Peneliti dalam pengujiannya mengetahui beberapa kondisi diluar dari skenario pengujian yang memungkingkan bisa mempengaruhi hasil uji. Dari 1300 data yang diambil diketahui :

Jumlah data sentimen positif : 488
Jumlah data sentimen negatif : 411
Jumlah data sentimen mixed : 401

Jumlah data sentimen diatas merupakan perbandingan banyaknya sentimen keseluruhan dari positif, negatif dan Mixed.

Jumlah data sentimen T_Positif : 663
Jumlah data sentimen T_Negatif : 637

Kategori sentimen teks hanya terdapat dua T_Positif dan T_Negatif yang jumlah datanya ditampilkan seperti diatas.

Jumlah kata keseluruhan : 8149
Jumlah kata pada kelas T_Positif : 1789
Jumlah kata pada kelas T_Negatif : 6360
Jumlah kata unik seluruh kelas : 2385

4.1 Skenario Uji Naive Bayes – C4.5 dan Levenshtein Distance

Pengujian ini bertujuan untuk mengetahui nilai akurasi dari ketiga metode Levenshtein Distance dan Naive Bayes – C4.5 dalam mengklasifkasi sentimen ulasan aplikasi mobile. Pengujian dilakukan dengan menggunakan metode evaluasi K-Fold dengan Fold=10 dan nilai edit=2. Hasil pengujian yang dihasilkan ditampilkan pada Tabel 3.

Tabel 3. Pengujian Naïve Bayes-C4.5 dan Levenshtein Distance

Fold Ke-	Naive Bayes – C4.5 Levenshtein Distance
1	0,861
2	0,846
3	0,823
4	0,838
5	0,861
6	0,853
7	0,907
8	0,830
9	0,803
10	0,915
Rata-Rata	**0,853 atau 85.3%**

4.2 Skenario Uji Naive Bayes – C4.5 Tanpa Levenshtein Distance

Pengujian ini bertujuan untuk mengetahui seberapa jauh metode Levenshtein Distance dapat mempengaruhi nilai akurasi dari klasifikasi sentimen aplikasi mobile. Hasil pengujian yang dihasilkan ditampilkan pada Tabel 4.

Tabel 4. Pengujian Naive Bayes – C4.5 tanpa Levenshtein Distance

Fold Ke-	Naive Bayes – C4.5
1	0,810
2	0,823
3	0,884
4	0,800
5	0,930
6	0,861
7	0,876
8	0,876
9	0,823
10	0,884
Rata-Rata	**0,856 atau 85.6%**

4.3 Skenario Penerapan Perbedaan Nilai Edit

Pengujian ini menggunakn metode Levenshtein Distance dan Naive Bayes – C4.5 dengan nilai edit distance yang berbeda-beda. Hasil pengujian yang dihasilkan ditampilkan pada Tabel 5.

Tabel 5. Pengujian Levenshtein Distance dan Naive Bayes – C4.5 dengan Nilai Edit Distance Berbeda

Fold Ke-	Naive Bayes – C4.5 Levenshtein Distance			
	Edit 1	Edit 2	Edit 3	Edit 4
1	0,861	0,823	0,907	0,830
2	0,915	0,884	0,884	0,861
3	0,876	0,876	0,834	0,838
4	0,869	0,846	0,869	0,884
5	0,861	0,876	0,907	0,853
6	0,869	0,876	0,876	0,838
7	0,90	0,876	0,830	0,823
8	0,823	0,846	0,853	0,876
9	0,847	0,823	0,864	0,884
10	0,869	0,869	0,892	0,923
Rata-Rata	**0,869**	**0,859**	**0,871**	**0,861**
Persentase	**86.9%**	**85.9%**	**87.1%**	**86.1%**

4.4 Skenario Pengujian Nilai Akurasi Naive Bayes

Pengujian ini menguji kemampuan algoritma Naïve Bayes tanpa mengunakan C4.5 dalam mengetahui sentimen teks. Hasil pengujian ditampilkan pada Tabel 6.

Tabel 6. Pengujian Naive Bayes Tanpa C4.5

Fold Ke-	Levenshtein Distance
1	0,90
2	0,938
3	0,846
4	0,815
5	0,830
6	0,869
7	0,838
8	0,846
9	0,892
10	0,830
Rata-Rata	**0,853 atau 85.3%**

5. ANALISIS DAN KESIMPULAN

5.1 Analisis

Data yang diambil terjadi ketimpangan jumlah kata yang cukup besar yakni 1789 kata bersentimen positif dengan 6360 bersentimen negatif dari 8149 jumlah kata. Perbedaan ini berpengaruh pada hasil dari proses klasifikasi untuk kata bersentimen positif lebih cepat dideteksi daripada kata bersentimen negatif yang memiliki kata yang lebih banyak. Perbedaan jumlah kata ini memang sangat sering terjadi dikarenakan penulis ulasan yang puas dengan aplikasi hanya menuliskan pujian.

Kesalahan dalam memperbaiki kata yang salah eja sering terjadi. Pertama, terbatasnya dataset sebagai pemberi saran normalisasi kata. Jika terdapat dua atau lebih kandidat kata edit distance yang sama, maka dipilih kata yang paling banyak di dataset. Contoh “Mask iya lag mulu” kata “Mask ” akan di normalisasi menjadi kata “masuk” dikarenakan kata “masuk” memiliki jumlah lebih banyak dalam dataset. Kedua, kata baru yang tidak ada dalam kamus kata dasar akan ikut ternomaliasi. Contoh “good” menjadi “gua” yang dimana gua terdapat dalam dataset.

Hasil dari klasifikasi dengan atau tanpa normalisasi Levenshtein Distance paling besar hanya menambah akurasi sebesar 1.5% dari pengujian tertinggi Tabel 4 dengan Tabel 5 nilai edit 3. Penambahan metode Levenshtein Distance tidak berdampak signifikan terhadap akurasi. Disamping itu, pengujian tanpa normalisasi Levenshtein Distance justru menambah akurasi sebesar 0,3% pada Tabel 4. Ketidakefektifan dan hasil yang sedikit berubah-ubah dikarenakan banyaknya kata tidak baku dan salah eja yang terdapat di dataset ulasan.

Hasil dari klasifikasi sentimen teks dengan metode Naive Bayes (Tabel 3) sama dengan hasil klasifikasi sentimen keseluruhan dengan

metode Naïve Bayes - C4.5 (Tabel 6) yang keduanya memiliki nilai 85.3%. Dari hasil ini, dapat dianalisis bahwa hasil klasifikasi dengan atau tanpa metode C4.5 memiliki hasil yang selaras. Lebih sederhana, Jika pada klasifikasi Naïve Bayes terdapat kesalahan, maka pada klasifikasi C4.5 juga terdapat kesalahan.

Seluruh skenario pengujian peneliti mengenai penerapan metode Levenshtein Distance dan Naïve Bayes – C4.5 terbukti bisa menyelesaikan permasalahan klasifikasi ulasan aplikasi mobile dengan akurasi diatas 80%. Mengenai keefektifan dari metode LD dan NB-C4.5 terhadap permasalahan, peneliti tidak mempunyai sumber berapa prosentase akurasi jika bisa dikatakan bagus.

5.2 Kesimpulan

Berdasarakan analisa dari penggunaan metode Levenshtein Distance dan Naïve Bayes – C4.5 dalam klasifikasi sentimen aplikasi mobile terdapat banyak faktor yang mempengaruhi nilai akurasi klasifikas i. Faktor pertama adalah perbedaan yang signifikan antara jumlah data sentimen positif dan negatif. Dalam hal ini mempengarhui nilai waktu komputasi dan nilai akurasi dari masing masing sentimen tersebut. Faktor kedua adalah banyaknya kata yang terdapat salah eja, tidak baku, kata baru dan kata yang bukan dari bahasa indonesia. Kata – kata ini akan menimbulkan kesalahan dalam proses klasifikasi. Faktor ketiga adalah penerapan Levenstein Distance yang dinilai tidak efektif secara signifikan mempengaruhi hasil klasifikasi. Secara keseluruhan penelitian penggunaan metode Levenshtein Distance dan Naïve Bayes – C4.5 dalam menganalisis sentimen aplikasi mobile memiliki nilai akurasi diatas 85.3% dengan nilai akurasi tertinggi sebesar 87.1%.

6. DAFTAR PUSTAKA

AppBrain. (2019). Android and Google Play statistics.

Bano, M., Zowghi, D., & Kearney, M. (2017, July). Feature based sentiment analysis for evaluating the mobile pedagogical affordances of apps. In IFIP World Conference on Computers in Education (pp. 281-291).

Chikersal, P., Poria, S., & Cambria, E. 2015. SeNTU: sentiment analysis of tweets by combining a rule-based classifier with supervised learning. In Proceedings of the 9th International Workshop on Semantic Evaluation, (pp. 647-651).

Freier, A. (2018). App revenue reaches $92.1 billion in 2018 driven by mobile gaming apps. [online] Tersedia di: https://www.businessofapps.com/news/app-revenue-reaches-92-1-billion-in-2018-driven-by-mobile-gaming-apps/ [Diakses 12 Juli 2020]

Gunawan, F., Fauzi, M. A., & Adikara, P. P. (2017). Analisis Sentimen Pada Ulasan Aplikasi Mobile Menggunakan Naive Bayes dan Normalisasi Kata Berbasis Levenshtein Distance (Studi Kasus Aplikasi BCA Mobile). Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer.

Indonesia, K. B. B. (2016). Melalui http:// https://kbbi.kemdikbud.go.id. Diakses 10 Agustus 2019.

Levenshtein, V. I. (1966, February). Binary codes capable of correcting deletions, insertions, and reversals. In Soviet physics doklady (Vol. 10, No. 8, pp. 707-710).

Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. Ain Shams engineering journal, 5(4), 1093-1113.

Nazief C. and M. Adriani. Confix Stripping: Approach to Stemming Algorithm for Bahasa Indonesia. Technical report, Faculty of Computer Science, University of Indonesia, Depok, 1996.

Olabenjo, B. (2016). Applying Naive Bayes Classification to Google Play Apps Categorization. arXiv preprint arXiv:1608.08574.

Parveen, H., & Pandey, S. (2016). Sentiment analysis on Twitter Data-set using Naive Bayes algorithm. In 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATcT) (pp. 416-419).

Rish, I. (2001, August). An empirical study of the naive Bayes classifier. In IJCAI 2001 workshop on empirical methods in artificial intelligence (Vol. 3, No. 22, pp. 41-46).

Statista. (2018). Number of apps available in leading app stores as of 1st quarter 2018. Tersedia melalui https://www.statista.com/ statistics/276623/ number-of-apps-available-in-leading-app-stores/ [Diakses 12 Juli 2020].

Sutabri, T., Putra, S. J., Effendi, M. R., Gunawan, M. N., & Napitupulu, D. (2018, August). Sentiment Analysis for Popular e-traveling Sites in Indonesia using Naive Bayes. In 2018 6th International Conference on Cyber and IT Service Management (CITSM) (pp. 1-4).

Tala, Fadillah. Z., 2003, A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia, [e-journal], Tersedia melalui: eprints.illc.uva.nl/740/ Pereferensian Institute for Logic, Language and Computation Universite van Amsterdam [Diakses 12 Juli 2020]