Klasifikasi Ulasan Pengguna Aplikasi Mandiri Online di Google Play Store dengan Menggunakan Metode *Information Gain* dan *Naive Bayes Classifier*

Amalia Elma Sari[1], Sri Widowati[2], Kemas Muslim Lhaksmana[3]

[1,2,3]Fakultas Informatika, Universitas Telkom, Bandung
[1]amaliaelmasari@students.telkomuniversity.ac.id, [2]sriwidowati@telkomuniversity.ac.id,
[3]kemasmuslim@telkomuniversity.ac.id

Abstrak

Ulasan pengguna merupakan salah satu fitur yang disediakan oleh platform Google Play Store agar pengguna dapat memberikan *feedback* dalam bentuk rating dan ulasan untuk aplikasi yang diunduh. Fungsi ulasan disediakan untuk mengizinkan pengguna memberikan komentar dari aplikasi yang telah digunakan, sehingga menyediakan umpan balik bagi pengembang aplikasi. Penelitian ini membangun sebuah sistem yang dapat melakukan klasifikasi ulasan pengguna di Google Play Store termasuk kedalam ulasan positif atau negatif, serta mengklasifikasikan berdasarkan faktor kualitas perangkat lunak ISO/IEC 25010. Topik ini diangkat untuk memudahkan para pengembang perangkat lunak tersebut untuk memperbaiki kualitas dan kepuasan dari aplikasi berdasarkan ulasan yang diberikan oleh pengguna. Naive Bayes Classifier dipilih pada penelitian ini karena dikenal sebagai metode sederhana, cepat tetapi memiliki performansi yang tinggi dalam pengklasifikasian teks, dan untuk menanggulangi tingginya dimensi data maka Naive Bayes dikombinasikan dengan metode seleksi fitur yaitu Information Gain sebagai metode seleksi dalam memilih fitur-fitur yang berpengaruh untuk setiap label kelas. Hasil penelitian menunjukkan bahwa, akurasi dan f-measure yang didapat pada klasifikasi dengan seleksi fitur Information Gain yaitu 91,33% dan 89,18%.

Kata kunci : Ulasan Pengguna, ISO/IEC 25010, Naive Bayes Classifier, Information Gain, Klasifikasi Teks

Abstract

User reviews are one of the features provided by the Google Play Store platform so users can provide feedback in the form of ratings and reviews for downloaded applications. The review function is provided to allow users to comment on applications that have been used, thus providing feedback for application developers. This research builds a system that can classify user reviews on the Google Play Store including positive or negative reviews, and classifies based on ISO / IEC 25010 software quality factors. This topic is raised to make it easier for software developers to improve the quality and satisfaction of the application is based on user reviews. Naive Bayes Classifier was chosen in this study because it is known as a simple, fast method but has a high performance in classifying text, and to cope with high data dimensions, Naive Bayes is combined with the feature selection method, namely Information Gain as a selection method in selecting influential features. for each class label. The results showed that the accuracy and f-measure obtained in the classification by the Information Gain feature selection were 91.33% and 89.18%.

Keywords: User Reviews, ISO / IEC 25010, Naive Bayes Classifier, Information Gain, Text Classification

1. Pendahuluan

1.1 Latar Belakang

Perkembangan teknologi menyebabkan pertumbuhan pengguna perangkat *mobile* semakin meningkat. Hal tersebut mendorong para pengembang aplikasi untuk mengembangkan berbagai aplikasi. Terdapat beberapa platform distribusi aplikasi yang dikenal saat ini diantaranya adalah Apple Store, Windows Phone Store, dan Google Play Store yang memberikan kemudahan bagi *developer* untuk merilis aplikasi dan bagi pengguna (*user*) untuk men*download* aplikasi yang mereka butuhkan. Pada bulan Juni 2019 jumlah aplikasi yang tersedia di Google Play Store sebanyak 2,7 juta, salah satunya yaitu aplikasi Mandiri Online yang dikenal sebagai layanan mobile banking bagi nasabah Bank Mandiri.

Platform seperti Google Play Store menyediakan fitur agar pengguna dapat memberikan *feedback* dalam bentuk rating dan ulasan untuk aplikasi yang diunduh, yang mana hal tersebut sangat menarik dari perspektif perangkat lunak dan rekayasa kebutuhan [1]. Fungsi ulasan disediakan untuk mengizinkan pengguna untuk memberikan komentar dari aplikasi yang telah digunakan, yang tujuannya menyediakan mekanisme umpan balik dari *user* ke *developer* aplikasi [2].

Untuk menghasilkan informasi dari ulasan pengguna aplikasi yang ada, maka perlu dilakukan pengolahan data ulasan pengguna di Google Play Store. Ulasan pengguna sangat dibutuhkan karena dikenali sebagai sumber yang bernilai untuk memperbaiki aplikasi, peningkatkan nilai untuk pengguna, serta membantu *developer* untuk lebih memahami kebutuhan pengguna [3]. Rata-rata aplikasi di App Store menerima 23 ulasan pengguna setiap

harinya, dan untuk aplikasi popular menerima hampir 4275 ulasan setiap harinya [4]. Begitu banyak pengguna yang memberi ulasan sehingga pengembang mengalami kesulitan dalam memilah dan mengkategorikan ulasan pengguna tersebut [4]. Platform distribusi aplikasi yang ada menyediakan dukungan yang terbatas untuk *developer* untuk secara sistematis memfilter dan mengklasifikasikan ulasan untuk mendapatkan *requirement* tambahan dari pengguna [3]. Jika informasi kebutuhan dari ulasan pengguna dapat automatis diindentifikasi dan diklasifikasi, itu akan membantu pengembang untuk memperbaiki kualitas dan kepuasan dari aplikasi [2].

Maka dari itu pada tugas akhir ini akan dilakukan penelitian mengenai klasifikasi ulasan pengguna berdasarkan kombinasi antara kelas sentimen dan kelas kategori. Kelas sentimen terbagi menjadi dua polaritas yaitu sentimen positif dan negatif. Kelas kategori dipilih berdasarkan standar ISO/IEC 25010 untuk menilai kualitas perangkat lunak yang terdapat pada aplikasi Mandiri Online. Keenam dimensi pada kelas kategori tersebut adalah *Functional Suitability*, *Usability*, *Reliability*, *Security*, *Portability*,dan *Performance Efficiency.* Metode yang digunakan pada tugas akhir ini adalah metode Naive Bayes Classifier. Metode Naive Bayes Classifier dipilih karena dikenal sebagai metode sederhana, cepat tetapi memiliki performansi yang tinggi dalam pengklasifikasian teks [5]. Namun banyaknya kata yang dapat mendefinisikan dan membentuk suatu dokumen menimbulkan masalah pada proses pengklasifikasian teks yaitu tingginya dimensi fitur. Permasalahan yang muncul akibat banyaknya dimensi fitur pada kategorisasi teks dapat mengurangi kinerja klasifikasi dan memperlambat waktu komputasi. Maka dari itu, pada tugas akhir ini digunakan metode seleksi fitur yaitu Information Gain *(IG).* Information Gain (IG) merupakan salah satu metode seleksi fitur dengan menghitung jumlah informasi dari kata tersebut ada atau tidaknya dalam suatu dokumen [6].

1.2 Perumusan Masalah

Adapun rumusan masalah pada tugas akhir ini adalah:

1. Bagaimana mengimplementasikan metode Naive Bayes Classifier dan Information Gain pada klasifikasi sentimen dan klasifikasi kategori kualitas perangkat lunak dari ulasan pengguna aplikasi Mandiri Online di Google Play Store?
2. Bagaimana hasil penerapan sistem yang dibangun untuk membantu pengembang dalam menyimpulkan kualitas dari aplikasi Mandiri Online berdasarkan ulasan pengguna?

1.3 Batasan Masalah

Adapun batasan masalah dari tugas akhir ini adalah:

1. Dataset yang digunakan pada tugas akhir ini adalah ulasan pengguna berbahasa Indonesia dari aplikasi Mandiri Online di Google Play Store.
2. Metode yang digunakan pada tugas akhir ini adalah *Information Gain* dan *Naive Bayes Classifier*.
3. Tidak membandingkan metode *Naive Bayes Classifier* dengan metode klasifikasi lainnya.
4. Kelas klasifikasi kategori berdasarkan standar kualitas perangkat lunak ISO/IEC 25010.
5. Satu ulasan pengguna hanya dapat diklasifikasikan dalam satu kelas sentimen dan satu kelas kategori.
6. Terdapat dua kelas klasifikasi sentimen yaitu sentimen positif dan negatif.
7. Terdapat enam kelas klasifikasi kategori kualitas perangkat lunak yang digunakan pada tugas akhir ini yaitu *Functional Suitability*, *Usability*, *Reliability*, *Security*, *Portability*,dan *Performance Efficiency.*

1.4 Tujuan

Adapun tujuan dari tugas akhir ini yaitu:

1. Mengklasifikasikan ulasan pengguna aplikasi Mandiri Online yang terdapat di Google Play Store dengan menggunakan metode *Naive Bayes Classifier* dan metode *Information Gain* .
2. Mendapatkan hasil penerapan sistem dalam menyimpulkan kualitas dari aplikasi Mandiri Online.

1.5 Sistematika Penulisan Tugas Akhir

Untuk penulisan laporan ini adalah sebagai berikut: Bagian 2 menunjukan penelitian-penelitian terkait dengan tugas akhir ini. Sistem yang akan diajukan untuk analisis sentiment ulasan pengguna di Playstore mengenai aplikasi Mandiri Online dengan metode *Naive Bayes Classifier*. Cara kerja *Naive Bayes Classifier* akan dijelaskan pada Bagian 3. Pada bagian 4 akan didiskusikan mengenai hasil pengujian dan evaluasi sistem. Dan kesimpulan akan dijelaskan pada bagian 5.

2. Studi Terkait

2.1 Kualitas Perangkat Lunak (ISO/IEC 25010)

ISO/IEC 25010 adalah salah satu metode standar internasional yang digunakan untuk menilai model kualitas sebuah perangkat lunak yang dibuat oleh *International Organization for Standarization and International Electrotechincal Commision*. ISO/IEC 25010 memiliki 8 karakteristik yaitu:

a. *Functional Suitability*: Tingkatan dimana sistem sudah menyediakan fungsi yang baik untuk memenuhi kebutuhan *user*, baik yang telah ditentukan pada awal perancangan sistem maupun sebaliknya.
b. *Reliability:* Tingkatan dimana sistem atau elemen didalamnya mampu menjalankan fungsionalitas pada beberapa kondisi tertentu dan jangka waktu yang telah ditentukan.
c. *Security:* Tingkatan dimana sistem mampu melindungi informasi dan data yang ada didalamnya, sehingga *user* luar maupun sistem lain tidak bisa melakukan akses ilegal.
d. *Portability:* Tingkatan efektifitas dan efisiensi dimana sistem dapat dipindahkan dari satu *hardware*, *software* atau lingkungan operasional ke properti lainnya.
e. *Usability:* Tingkatan dimana sebuah sistem dapat digunakan untuk memenuhi tujuan tertentu yang sesuai dengan efektifitas, efisiensi dan kepuasan dalam konteks penggunaannya.
f. *Performance:* Performansi sistem yang berhubungan dengan jumlah sumber daya yang digunakan pada kondisi tertentu.
g. *Compatability*: Tingkatan dimana sistem maupun elemen didalamnya dapat bertukar informasi satu sama lain serta menjalankan fungsionalitasnya masing-masing ketika saling berbagi *hardware* atau *software* yang sama.
h. *Maintability:* Tingkatan efektifitas dan efisiensi dimana sistem dapat dimodifikasi sesuai yang diinginkan user terkait

2.2 Seleksi Fitur *Information Gain*

Fitur adalah seluruh kata yang muncul dalam *training set*. *Set* ini biasanya sangat besar yaitu satu dimensi untuk satu kata unik sehingga membutuhkan *feature space* yang cukup besar. Seleksi fitur adalah metode untuk mereduksi dimensi fitur dengan cara memilih kata-kata yang paling informative. *Information Gain* merupakan salah satu teknik pemilihan fitur dimana suatu kata diukur dengan menghitung jumlah bit informasi dengan mengamati kemunculan kata tersebut dalam dokumen [6]. *Information gain* digunakan pada tahapan seleksi fitur untuk mengurangi fitur yang tidak relevan atau mengurangi dimensi fitur pada data. Fitur yang digunakan penelitian ini adalah kata. Rumus yang digunakan untuk mencari Entropy:

$$info(D) = -\Sigma_{i=1}^{c} P(i) log_2 P(i) \quad (1)$$

dengan keterangan:

c = jumlah nilai pada atribut target (jumlah kelas klasifikasi)

P(i) = proporsi sample pada kelas i dengan sampel total

kemudian dicari nilai entropy setelah pembobotan dalam setiap fiturnya yaitu dengan rumus:

$$Info_A(D) = -\Sigma_{j=1}^{c} \frac{|D_j|}{|D|} InfoD_j \quad (2)$$

dengan keterangan:

A = Atribut

|D | = Jumlah seluruh sampel data

$|D_j|$ =Jumlah sampel untuk nilai partisi j

v = Jumlah nilai yang mungkin untuk atribut A

$InfoD_j$ = Entropy untuk setiap partisi j

Nilai information gain didapatkan dari pengurangan dari nilai dari persamaan (1) dengan nilai dari persamaan (2), seperti rumus dibawah ini:

$$InfoGain(A) = -Info(D) - |Info_A(D)| \quad (3)$$

2.3 Naive Bayes Classifier

Naive Bayes Classifier adalah sebuah metode klasifikasi sederhana yang mengaplikasikan teorema Bayes. Dalam perhitungan probabilitas, *Naive Bayes Classifier* mengasumsikan bahwa nilai atribut suatu kelas tidak memiliki keterkaitan dengan keberadaan atribut dikelas lain [5]. Meski dalam kenyataanya setiap atribut hampir dipastikan memiliki ketergantungan dengan atribut lain, namun dengan asumsi naif seperti ini membuat *Naive Bayes* mudah dalam perhitungan. Perhitungan *Naive Bayes* dirumuskan dengan menggunakan persamaan:

$$C_{map} = \frac{argmax}{c \in C} P(c|d) = \frac{argmax}{c \in C} P(c) \pi_{i=1}^{n} P(wi|c) \quad (4)$$

dengan keterangan:

P(c|d) = *Posterior probability* dokumen d terhadap kelas c

P(wi|c) = *Conditional probability* kemunculan sebuah kata dalam dokumen pada kelas c

P(c) = *Prior probability* kelas c

$$P(c) = \frac{Nc}{N} \quad (5)$$

Dimana Nc merupakan jumlah dokumen yang merupakan kelas c dan N merupakan jumlah dokumen pada data latih. Untuk perhitungan *conditional probability* dilakukan dengan menggunakan persamaan:

$$P(w,c) = \frac{count(w,c)+1}{count(c)+|v|} \quad (6)$$

Dimana *count*(w|c) merupakan jumlah kata yang diamati pada kelas c, *count*(c) adalah jumlah seluruh kata yang ada pada kelas c dan |v| merupakan kata unik yang terdapat pada data latih

2.4 Evaluasi Performansi

Evaluasi performansi dilakukan untuk menguji hasil klasifikasi dari sistem yang telah dibuat dengan menghitung nilai akurasi dan *f-measure* [8]. Akurasi merupakan perhitungan jumlah data yang diklasifikasikan dengan benar dibanding dengan seluruh data. *F-measure* merupakan nilai yang mewakili keseluruhan sistem, dimana dilakukan perhitungannya terhadap nilai *precision* dan *recall*. Perhitungan performansi menggunakan *confusion matrix* seperti pada Tabel 1 dibawah ini:

Tabel 1. *Confusion Matrix*

		Nilai Aktual	
		Positive	**Negative**
Nilai Prediksi	**Positive**	*True Positive*(TP)	*False Positive* (FP)
	Negative	*False Negative* (FN)	*True Negative* (TN)

Dimana TP (*True Positive*) jika prediksi positif dan nilai sebenarnya positif, FP (*False Posiive*) jika prediksi positif dan nilai sebenarnya negatif, FN (*False Negative*) jika prediksi negatif dan nilai sebenarnya positif, TN (*True Negative*) jika prediksi negatif dan nilai sebenarnya negatif. Nilai akurasi merupakan perbandingan antara data yang terklasifikasi benar dengan keseluruhan data, nilai akurasi dapat diperoleh dari persamaan (9) dan f-measure diperoleh dari persamaan (12) :

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

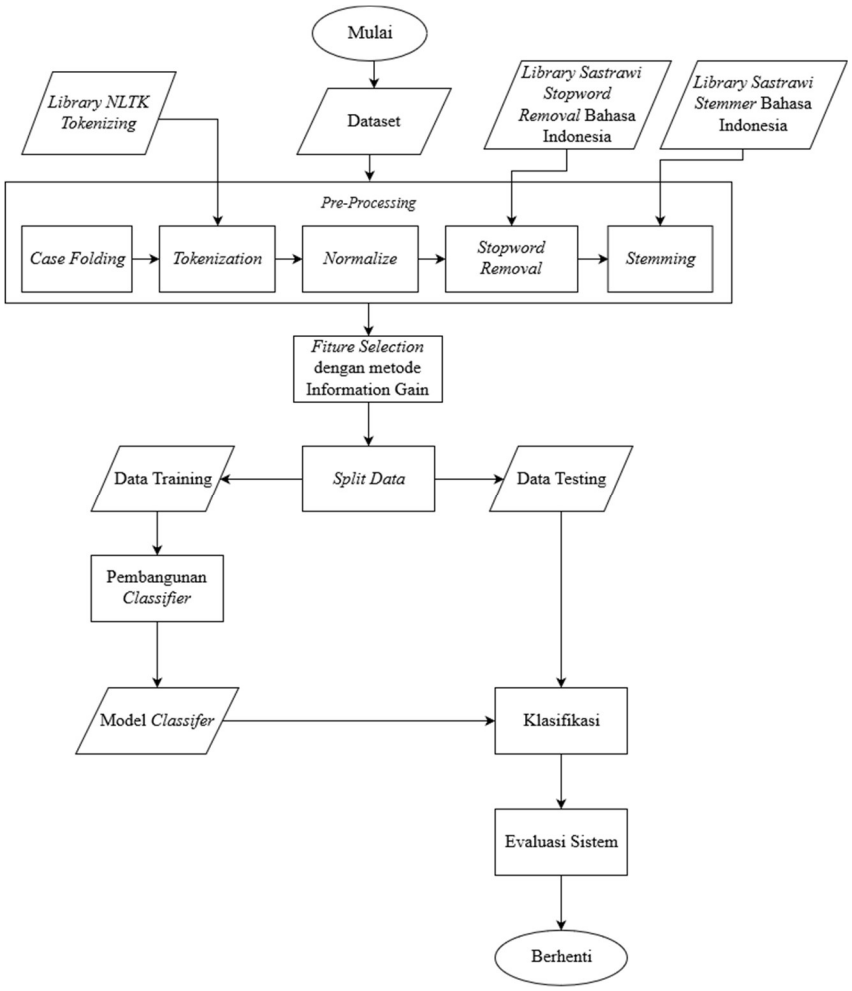
$$precision = \frac{TP}{FP+TP} \quad (9)$$

$$Recall = \frac{TP}{FN+T} \quad (8)$$

$$f-measure = \frac{2.precision.recall}{precision.recall} \quad (10)$$

3. Sistem yang Dibangun

Dalam tugas akhir ini akan dibangun sistem yang dapat mengklasifikasikan sentimen dan kategori mengenai ulasan pengguna aplikasi Mandiri Online di Google Play Store. Gambaran umum dari sistem yang akan dibangun pada tugas akhir ini dapat dilihat pada Gambar 1.



Gambar 1. Gambaran Rancangan Sistem

3.1 Pengumpulan Data

Dataset yang digunakan dalam penelitian diambil dari ulasan pengguna aplikasi Mandiri Online di Google Play Store yang belum diberi label. Kemudian dataset tersebut diberi label manual yang terdiri dari dua kelas sentimen (Positif dan Negatif) dan enam kelas kategori yang berdasarkan ISO 25010 (*Portability*, *Functional*, *Usability*, *Reliability*, *Performance*, *Security*) dan kelas kategori tambahan yaitu *Others*, dimana kelas kategori tersebut untuk ulasan pengguna yang tidak dapat dikelompokan berdasarkan ISO 25010. Contoh pelabelan dataset terdapat pada tabel 2.

Tabel 2. Contoh dataset

Ulasan	Sentimen	Kategori
"Tidak bisa diinstall di hp os kitkat (android 4.4.2), selalu tertulis aplikasi ini tidak kompatibel !"	Negatif	*Portability*
"Untuk menu multipayment itu tolong dikasih kolom search supaya tidak harus scroll jauh untuk yang mau payment karna scrollingnya itu makan waktu."	Negatif	*Usability*
"Aplikasi bagus, thanks mandiri sangat membantu."	Positif	*Others*

Jumlah dataset yang digunakan sejumlah 1500 ulasan pengguna dengan porsi kelas sentimen yaitu 638 ulasan positif dan 862 ulasan negatif, serta untuk kelas kategori yaitu 165 ulasan *Functional,* 62 ulasan *Usability, 221* ulasan *Security, 85* ulasan *Portability,* 229 ulasan *Performance Efficiency, 100* ulasan *Reliability* dan 638 ulasan *Others*.

3.2 *Pre-Processing*

Pada tahap ini dilakukan *preprocessing* data, hal ini dilakukan guna menghapus kata yang kurang relevan untuk proses pengklasifikasian sehingga dataset menjadi lebih sederhana serta untuk meningkatan performa dari sistem yang dibangun. Berikut adalah proses yang dilakukan dalam *pre-processing*:

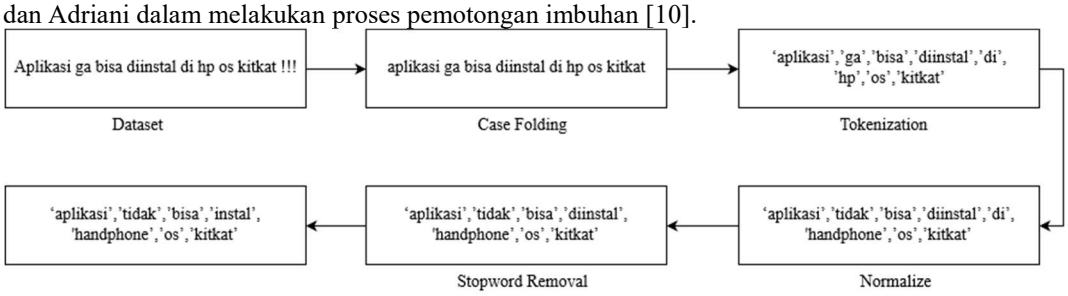
a. *Case Folding*
Case Folding adalah proses menyamakan semua huruf menjadi huruf kecil (*lowercase*). Tujuan digunakan *case folding* untuk menghilangkan ketidakonsistenan penggunaan huruf kapital dan huruf kecil pada ulasan pengguna. Pada tahap ini juga dilakukan pembersihan data dari karakter yang tidak akan digunakan seperti (&,/,*,0,(),dll).

b. *Tokenization*
Tokenization yaitu proses pemotongan string input berdasarkan pada tiap kata penyusunnya dengan menggunakan spasi sebagai pembatas. Tujuan dilakukan *tokenization* adalah memisahkan setiap kata yang menyusun pada suatu dokumen.

c. *Normalize*
Normalize (Normalisasi Kata) adalah proses yang digunakan untuk mengubah kata-kata yang tidak baku menjadi kata baku dan pengubahan kata singkatan menjadi kata asalnya. Perbaikan kata tidak baku dilakukan dengan proses normalisasi yang dilakukan dengan melakukan pengecekan token pada kalimat dengan kamus katabaku. Pendaftaran kata yang akan diubah mengacu pada dataset yang digunakan pada penelitian.

d. *Stopword Removal*
Stopword Removal yaitu proses menghilangkan kata-kata yang sering muncul namun tidak memiliki arti atau tidak relevan. Pada penelitian ini penulis menggunakan data *stopword* dalam Bahasa Indonesia dari hasil penelitian Fadhillah[9], Namun daftar *stopword removal* ini akan ditambahkan dan dimodifikasi sesuai dengan kebutuhan penelitian ini.

e. *Stemming*
Stemming yaitu proses mengubah kata berimbuhan menjadi kata dasar. Salah satu *library stemming* bahasa indonesia yang disering digunakan adalah *library* Sastrawi yang menerapkan algoritma Nazief dan Adriani dalam melakukan proses pemotongan imbuhan [10].



Gambar 2. Proses *Pre-processing*

3.3 Seleksi Fitur dengan *Information Gain*

Pada tahap ini dilakukan seleksi fitur dengan menggunakan metode *Information Gain*. Seleksi fitur bertujuan untuk mengurangi jumlah fitur, mempercepat proses klasifikasi serta meningkatkan performansi klasifikasi dengan cara menghilangkan fitur yang tidak relevan, redundan atau noise [6]. *Information gain* akan menghitung nilai dari setiap kata dalam sekumpulan kata. Kata-kata dengan nilai *information gain* yang tinggi akan menjadi fitur utama dari setiap data dalam proses klasifikasi. Berikut contoh perhitungan manual dari *information gain*:

Tabel 3. contoh koleksi data

Dokumen	Fitur			Sentimen
	Bagus	Susah	Bisa	
D1	Ya	Tidak	Ya	Positif

D2	Tidak	Ya	Tidak	Negatif
D3	Ya	Ya	Tidak	Negatif
D4	Tidak	Tidak	Ya	Positif
D5	Ya	Tidak	Tidak	Positif

Fitur yang terdapat pada tabel 4 merupakan potongan kata dari dokumen yang akan dihitung bobotnya. Pada kasus perhitungan bobot penulis akan mengambil contoh kata "Bagus" dengan menghitung entropy pada dataset dengan menggunakan persamaan (1) sebagai berikut:

$$Entropy(Set) = -[(\frac{3}{5})log^2(\frac{3}{5})+(\frac{2}{5})log^2(\frac{2}{5})] = 0.9709$$

Selanjutnya ambil contoh pada kata "Bagus" yang memiliki value "Ya" atau "Tidak" sehingga dapat dihitung dengan persamaan (1), setelah itu hitung Entropy (S_{Bagus}) dengan persamaan (2) dan menghasilkan perhitungan sebagai berikut:

$$Entropy(Ya) = -[(\frac{2}{3})log^2(\frac{2}{3})+(\frac{1}{3})log^2(\frac{1}{3})] = 0.9182$$

$$Entropy(Tidak) = -[(\frac{1}{2})log^2(\frac{1}{2})+(\frac{1}{2})log^2(\frac{1}{2})] = 1$$

$$Entropy(S_{Bagus}) = -[(\frac{3}{5}) * 0.9182+(\frac{2}{5}) * 1] = 0.9509$$

Langkah terakhir untuk mencari nilai *information gain* menggunakan persamaan berikut (3) :

$$\text{Gain}(S_{Bagus}) = 0.9709 - 0{,}9509 = 0.02$$

Semakin besar nilainya maka semakin menandakan bahwa kata tersebut berpengaruh pada kelas-kelas yang ada. Setelah itu, kata-kata dengan nilai diatas standar (*threshold*) dan yang paling tinggi yang akan dipilih sebagai fitur utama dalam melakukan klasifikasi.

3.4 Pembangunan *Classifier*

Pada tahap ini dataset hasil *preprocessing* yaitu data *training* akan digunakan dalam pembangunan model *classifier*. Dimana pembangunan model *classifier* ini dibangun dengan metode Naive Bayes Classifier. Pembangunan *classifier* dilakukan dengan menghitung nilai *prior probability* dan *conditional probability* pada data *training*. Hasil perhitung *prior probability* dan *conditional probability* adalah penyusun model Naive Bayes Classifier. Model ini nantinya akan digunakan untuk menilai *posterior probability* untuk menentukan kelas dari data *testing* yang digunakan dalam penelitian tugas akhir ini. Berikut ini merupakan perhitungan klasifikasi menggunakan Naive Bayes berdasarkan pada data pada Tabel 5.

Tabel 4. Contoh Data

Dokumen (d)	**Fitur**	**Kelas**	**Kategori**
1	tidak bisa install hp os kitkat android aplikasi tidak kompatibel	Negatif	*Portability*
2	menu multipayment kasih kolom search scroll jauh payment scroll makan waktu	Negatif	*Usability*
3	aplikasi bagus, thanks mandiri sangat membantu	Positif	*Others*
4	aplikasi mandiri bagus	?	?

1. Langkah pertama adalah menghitung nilai *prior probability* untuk masing masing kelas sentimen dan kategori dengan menggunakan persamaan (1):
   P(positif) = 1/3 = 0.33 P(*functional*) = 1/3 = 0.33 P(*security*) = 1/3 =0.33
   P(negatif)= 2/3 = 0.66 P(*usability*) = 1/3 =0.33
2. Selanjutnya melakukan perhitung *conditional probability* untuk masing-masing kata pada data latih berdasarkan kelas sentimen dan kelas kategori berdasarkan persamaan (2):
   P(aplikasi,positif)=2/6+24=0,067 P(mandiri, *usability*)=1/11+24=0,028

P(aplikasi,negatif)=1/21+24=0,022
P(aplikasi,*portability*)=2/10+24=0,058
P(aplikasi,*usability*) = 1/11+24 =0,028
P(aplikasi, *others*)=2/6+ 24=0,067
P(mandiri,positif) = 2/6+24=0,067
P(mandiri,negatif) = 1/21+24=0,022
P(mandiri, *portability*)=1/10+24=0,029
P(mandiri,*others*)=2/6+24=0,067
P(bagus,positif)= 1/6+24= 0,067
P(bagus,negatif)= 1/21+24=0,022
P(bagus,*portability*)=1/10+24=0,29
P(bagus,*usability*)=1/11+24=0,028
P(bagus,*others*)=2/6+24=0,067

3. Selanjutnya melakukan perhitungan *posterior probability* masing-masing kelas sentimen dan kategori untuk mengetahui kelas dokumen 5 dengan persamaan (3):
   P(positif|d4)=0.33*0,067*0,067*0,067=0,0000099251
   P(negatif|d4)=0.66*0.022*0.022*0.022=0,0000007027
   P(*portability*|d4)= 0.33*0,058*0,029*0,029=0,00000160967
   P(*usability*|d4)= 0.33*0,028*0,028*0,028=0,00000072441
   P(*others*|d4)= 0.33*0,067*0,067*0,067=0,00000992517
   Maka dari hasil perhitungan kelas sentimen untuk dokumen 4 adalah kelas positif. karena, memiliki nilai *posterior probability* tertinggi yaitu 0.00000099251 4dan untuk kelas kategori adalah *others*, karena memiliki nilai *posterior probability* tertinggi yaitu 0.000000992517

3.5 Evaluasi Performansi

Tahapan terakhir adalah menghitung nilai akurasi dan *f-measure* dari hasil klasifikasi yang telah dilakukan pada sentimen dan kategori. Hal ini dilakukan untuk mengukur performansi dari sistem yang dibangun

4. Evaluasi

Setelah sistem selesai dibuat, maka tahap selanjutnya adalah pengujian terhadap sistem. Pengujian ini bertujuan untuk mengetahui seberapa baik sistem yang telah dibuat dengan memastikan bahwa input akan memberikan hasil yang sesuai dengan kebutuhan dan bertujuan untuk mengetahui performansi dari sistem.

4.1 Hasil dan Analisis Hasil Pengujian Pengaruh *Information Gain* terhadap Klasifikasi Sentimen.

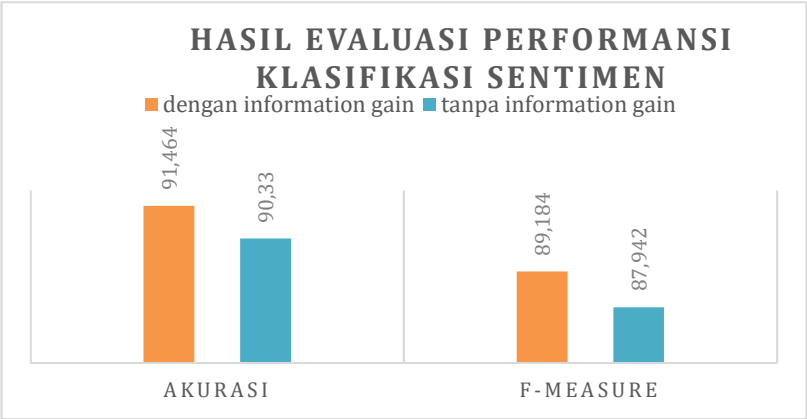
Pengujian yang dilakukan adalah pengklasifikasi *Naive Bayes Classifier* tanpa *feature selection Information Gain* dan dengan menggunakan *feature selection Information Gain* yang menerapkan K-Fold *cross validation* dengan nilai k=10. Pengujian ini dilakukan pada dataset Mandiri Online untuk dapat mengklasifikasikan sentimen dan kategorinya. Tujuan dari pengujian ini adalah untuk melihat seberapa pengaruh *Information Gain* terhadap performa klasifikasi.

Tabel 5 Hasil evaluasi pengaruh *information gain* terhadap akurasi dan f-measure klasifikasi sentimen

Fold	Akurasi		f-measure	
	tanpa information gain	dengan information gain	tanpa information gain	dengan information gain
1	90,66	92,00	86,27	91,3
2	90,00	91,33	87,6	89,07
3	90,66	93,33	89,23	92,42
4	84,66	93,33	79,64	90,74
5	92,00	87,33	92,2	84,03
6	91,33	90,66	89,6	88,13
7	89,33	94,00	85,96	91,89
8	89,33	95,33	88,57	93,33
9	93,33	87,33	91,07	83,76
10	92,00	90,00	89,28	87,17
Performansi	90,33	91,46	87,94	89,18

Berdasarkan hasil evaluasi pengujian pengaruh *information gain* pada proses klasifikasi sentimen dapat dilihat pada Tabel 5 menunjukan nilai akurasi dan f-measure performa klasifikasi ulasan pengguna menggunakan metode *naive bayes* sebelum menggunakan *information gain* adalah sebesar 90,33 % dan

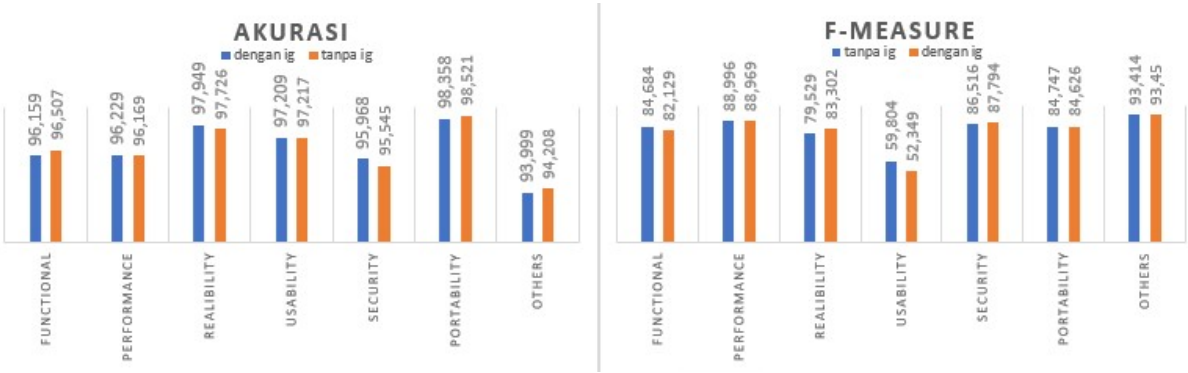
87,94%, sedangkan akurasi dan f-measure performa klasifikasi *naive bayes* setelah menggunakan *information gain* adalah 91,46% dan 89,18%.



Gambar 3 Perbandingan Hasil Performansi Klasifikasi Sentimen

Performa Naive Bayes mengalami peningkatan setelah menggunakan metode *feature selection information gain* dikarenakan *information gain* melakukan pemilihan fitur yang akan digunakan pada proses klasifikasi, proses seleksi dipilih berdasarkan hasil perhitungan gain, dimana fitur yang memiliki nilai gain lebih dari *threshold* akan dipilih, semakin tinggi nilai gain yang dimiliki maka semakin signifikan kata tersebut dalam proses klasifikasi, kata yang tidak terpilih akan dihilangkan oleh *information gain*, sehingga menyebabkan dimensi data berkurang. oleh karena itu dengan berkurangnya dimensi data, performa yang dihasilkan *naive bayes* juga akan meningkat.

4.2 Hasil dan Analisis Hasil Pengujian Pengaruh *Information Gain* terhadap Klasifikasi Kategori



Gambar 4 Perbandingan Hasil Performansi Klasifikasi Kategori

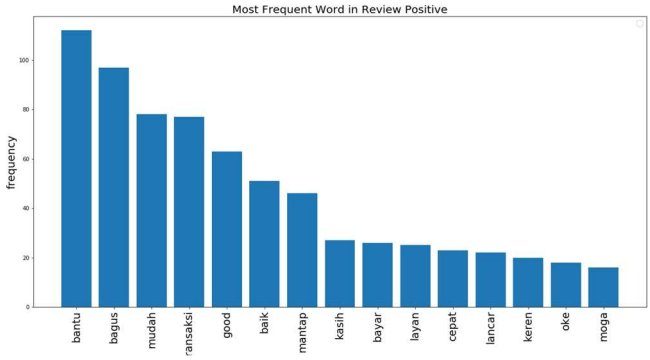
Pada Gambar 4 menunjukan bahwa pada klasifikasi kategori ulasan pengguna aplikasi Mandiri online menggunakan *Naive Bayes Classifier* tanpa dan dengan *feature selection* information gain tidak berpengaruh untuk proses klasifikasi kategori. Hal ini disebabkan karena perhitungan *feature selection* information gain hanya dilakukan berdasarkan kelas sentimen, yang artinya information gain digunakan untuk mengukur nilai ketergantungan antara fitur atau kata dengan kelas sentimen dan tidak berlaku untuk nilai ketergantungan fitur dengan kelas kategori.

4.3 Hasil dan Analisis Visualisasi Ulasan Pengguna

Visualisasi ulasan pengguna aplikasi Mandiri Online dilakukan dengan mengekstraksi informasi berupa topik yang sering dibicarakan oleh pengguna aplikasi tersebut. Sehingga hasil dari visualisasi ini dapat digunakan untuk mengetahui bagaimana kualitas aplikasi Mandiri Online dari sudut pandang konsumen, serta harapan konsumen terhadap Aplikasi Mandiri Online.

Visualisasi dilakukan dengan cara pembuatan diagram frekuensi kata dan *wordcloud*. Visualisasi tersebut digunakan agar dapat mengetahui frekuensi kata yang sering digunakan pengguna dalam mengulas kinerja aplikasi Mandiri Online pada setiap kelas sentimen (positif dan negatif) dan kelas kategori (*functional*, *security*, *portability*, *realibility*, *usability*, dan *performance*). Berikut akan dijelaskan visualisasi berupa diagram frekuensi kata dan *wordcloud* dari setiap ulasan pengguna aplikasi Mandiri Online.

4.3.1 Ulasan Kelas Sentimen Positif

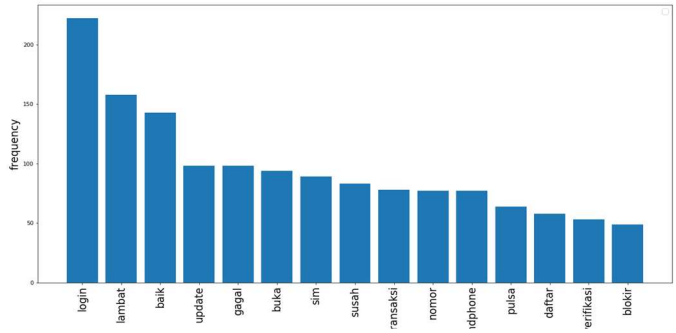


Gambar 5 Kata Frekuensi Terbanyak dari Kelas Sentimen Positif Mandiri Online

Gambar 6 *Word Cloud* Kelas Sentimen Positif Mandiri Online

Pada Gambar 5 merupakan grafik yang menunjukan 15 kata yang paling sering digunakan pada ulasan positif yang ditulis oleh pengguna Mandiri Online sehingga menunjukan perasaan pengguna aplikasi yang diwakilkan oleh kata tersebut. Pada visualisasi *wordcloud* dapat dilihat topik yang sering dibicarakan pengguna aplikasi Mandiri Online untuk memberikan ulasan positif. Ukuran kata yang terbentuk menunjukan banyaknya frekuensi kata tersebut, semakin besar kata tersebut menunjukan bahwa kata tersebut sering digunakan oleh pengguna untuk menjelaskan bagaimana aplikasi Mandiri Online bekerja. Pada gambar 6, *wordcloud* dengan kata "bantu" kata dasar dari "membantu" paling sering digunakan pada pembuatan ulasan, diikuti kata "bagus" kemudian kata "mudah". Kata tersebut menggambarkan bahwa terdapat beberapa ulasan positif dari pengguna yang menunjukan bahwa aplikasi Mandiri Online membantu, bagus dan mudah.

4.3.2 Ulasan Kelas Sentimen Negatif

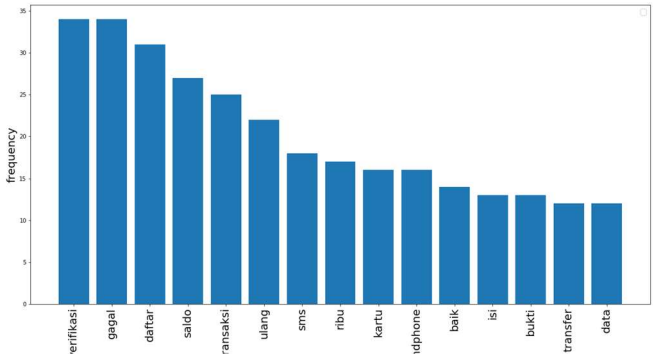


Gambar 7 Kata Frekuensi Terbanyak dari Kelas Sentimen Negatif Mandiri Online

Gambar 8 *Word Cloud* Kelas Sentimen Negatif Mandiri Online

Pada Gambar 7 merupakan grafik yang menunjukan 15 kata yang paling sering digunakan pada ulasan negatif yang ditulis oleh pengguna Mandiri Online sehingga menunjukan perasaan pengguna aplikasi yang diwakilkan oleh kata tersebut. Pada visualisasi *wordcloud* dapat dilihat topik yang sering dibahas pengguna aplikasi untuk memberikan ulasan negatif pada kinerja aplikasi. Pada gambar 8, *wordcloud* dengan pembahasan kata "login" paling sering digunakan pada pembuatan ulasan, diikuti kata "lambat" kemudian kata "baik" dari kata dasar "perbaiki/memperbaiki". Kata tersebut menggambarkan bahwa terdapat beberapa topik yang sering dibahas oleh pengguna yang menunjukan bahwa aplikasi Mandiri Online memiliki ulasan negatif pada bagian *login*, performansi aplikasi yang lambat serta perlunya perbaikan aplikasi.

4.3.3 Ulasan Kelas Kategori *Functional*

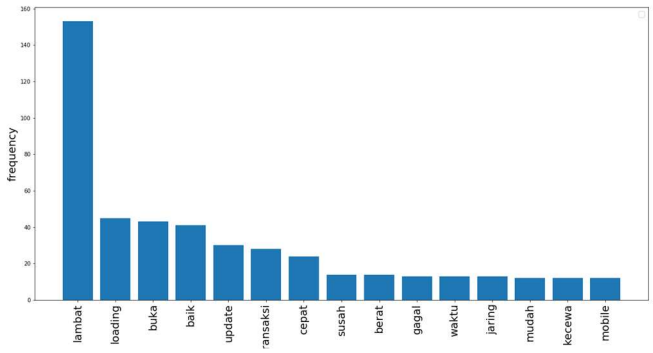


Gambar 9 Kata Frekuensi Terbanyak dari Kelas Kategori *Functional* Mandiri Online

Gambar 10 *Word Cloud* Kelas Kategori *Functional* Mandiri Online

Pada Gambar 9 merupakan grafik yang menunjukan 15 kata yang paling sering digunakan pada ulasan yang ditulis oleh pengguna Mandiri Online sehingga menunjukan perasaan pengguna dalam mengulas ulasan negatif permasalahan pada *functional* aplikasi yang diwakilkan oleh kata tersebut. Pada visualisasi *wordcloud* dapat dilihat topik yang sering dibahas pengguna aplikasi untuk memberikan ulasan negatif kategori *functional* pada kinerja aplikasi. Pada gambar 10, *wordcloud* dengan pembahasan kata "verifikasi" paling sering digunakan pada pembuatan ulasan, diikuti kata "gagal" kemudian kata "daftar". Kata tersebut menggambarkan bahwa terdapat beberapa topik yang sering dibahas oleh pengguna yang menunjukan bahwa aplikasi Mandiri Online memiliki ulasan negatif pada kategori *functional* yaitu permasalahan verifikasi dan pendaftaran akun yang gagal.

4.3.4 Ulasan Kelas Kategori *Performance*

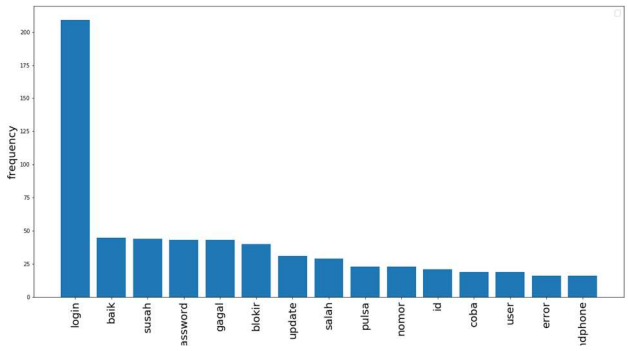


Gambar 11 Kata Frekuensi Terbanyak dari Kelas Kategori *Performance* Mandiri Online

Gambar 12 *Word Cloud* Kelas Kategori *Performance* Mandiri Online

Pada Gambar 11 merupakan grafik yang menunjukan 15 kata yang paling sering digunakan pada ulasan yang ditulis oleh pengguna Mandiri Online sehingga menunjukan perasaan pengguna dalam mengulas ulasan negatif permasalahan pada *performance* aplikasi yang diwakilkan oleh kata tersebut. Pada visualisasi *wordcloud* dapat dilihat topik yang sering dibahas pengguna aplikasi untuk memberikan ulasan negatif kategori *performance* pada kinerja aplikasi. Pada gambar 12, *wordcloud* dengan pembahasan kata "lambat" paling sering digunakan pada pembuatan ulasan, diikuti kata "loading" kemudian kata "buka". Kata tersebut menggambarkan bahwa terdapat beberapa topik yang sering dibahas oleh pengguna yang menunjukan bahwa aplikasi Mandiri Online memiliki ulasan negatif yaitu pada performansi yang lambat, permasalahan *loading* dan permasalahan pada saat membuka aplikasi.

4.3.5 Ulasan Kelas Kategori *Security*

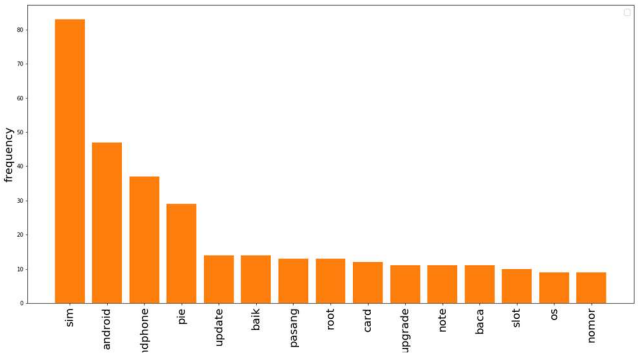


Gambar 13 Kata Frekuensi Terbanyak dari Kelas Kategori *Security* Mandiri Online

Gambar 14 *Word Cloud* Kelas Kategori *Security* Mandiri Online

Pada Gambar 13 merupakan grafik yang menunjukan 15 kata yang paling sering digunakan pada ulasan yang ditulis oleh pengguna Mandiri Online sehingga menunjukan perasaan pengguna dalam mengulas ulasan negatif permasalahan pada *Security* aplikasi yang diwakilkan oleh kata tersebut. Pada visualisasi *wordcloud* dapat dilihat topik yang sering dibahas pengguna aplikasi untuk memberikan ulasan negatif kategori *security* pada kinerja aplikasi. Pada gambar 14, *wordcloud* dengan pembahasan kata "login" paling sering digunakan pada pembuatan ulasan, diikuti kata "baik" kata dasar "perbaiki" atau "memperbaiki" kemudian kata "susah". Kata tersebut menggambarkan bahwa terdapat beberapa topik yang sering dibahas oleh pengguna yang menunjukan bahwa aplikasi Mandiri Online memiliki ulasan negatif yaitu pada permasalahan *login* yang susah dan perlu diperbaiki.

4.3.6 Ulasan Kelas Kategori *Portability*

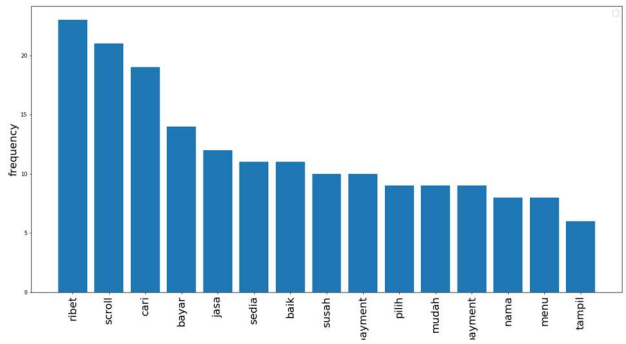


Gambar 15 Kata Frekuensi Terbanyak dari Kelas Kategori *Portability* Mandiri Online

Gambar 16 *Word Cloud* Kelas Kategori *Portability* Mandiri Online

Pada Gambar 15 merupakan grafik yang menunjukan 15 kata yang paling sering digunakan pada ulasan yang ditulis oleh pengguna Mandiri Online sehingga menunjukan perasaan pengguna dalam mengulas ulasan negatif permasalahan pada *portability* aplikasi yang diwakilkan oleh kata tersebut. Pada visualisasi *wordcloud* dapat dilihat topik yang sering dibahas pengguna aplikasi untuk memberikan ulasan negatif kategori *portability* pada kinerja aplikasi. Pada gambar 16, *wordcloud* dengan pembahasan kata "sim", diikuti kata "handphone" kemudian kata "android". Kata tersebut menunjukan bahwa aplikasi Mandiri Online memiliki ulasan negatif yaitu pada permasalahan pada penyesuaian aplikasi terhadap perangkat lunak, perangkat keras, dan lingkungan yang berbeda. Permasalahan *portability* terdapat pada ketentuan dalam pemasangan kartu sim, os android, dan tipe handphone.

4.3.7 Ulasan Kelas Kategori *Usability*

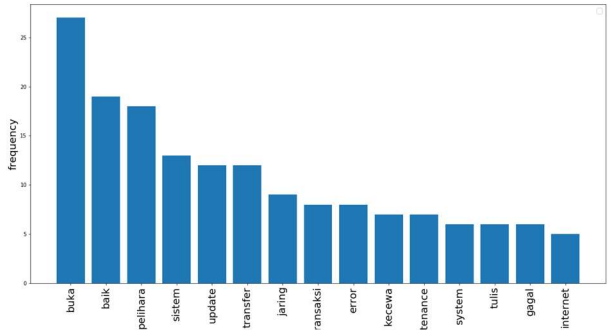


Gambar 17 Kata Frekuensi Terbanyak dari Kelas Kategori *Usability* Mandiri Online

Gambar 18 *Word Cloud* Kelas Kategori *Usability* Mandiri Online

Pada Gambar 17 merupakan grafik yang menunjukan 15 kata yang paling sering digunakan pada ulasan yang ditulis oleh pengguna Mandiri Online sehingga menunjukan perasaan pengguna dalam mengulas ulasan negatif permasalahan pada *usability* aplikasi yang diwakilkan oleh kata tersebut. Pada visualisasi *wordcloud* dapat dilihat topik yang sering dibahas pengguna aplikasi untuk memberikan ulasan negatif kategori *usability* pada kinerja aplikasi. Pada gambar 18, *wordcloud* dengan pembahasan kata "ribet", diikuti kata "*scroll*" kemudian kata "cari". Kata tersebut menunjukan bahwa aplikasi Mandiri Online memiliki ulasan negatif yaitu pada permasalahan *usability* yaitu aplikasi yang ribet atau sulit digunakan oleh pengguna, dan permasalahan *interface* aplikasi untuk proses pencarian.

4.3.8 Ulasan Kelas Kategori *Realibility*



Gambar 19 Kata Frekuensi Terbanyak dari Kelas Kategori *Realibility* Mandiri Online

Gambar 20 *Word Cloud* Kelas Kategori *Realibility* Mandiri Online

Pada Gambar 19 merupakan grafik yang menunjukan 15 kata yang paling sering digunakan pada ulasan yang ditulis oleh pengguna Mandiri Online sehingga menunjukan perasaan pengguna dalam mengulas ulasan negatif permasalahan pada *realibility* aplikasi yang diwakilkan oleh kata tersebut. Pada visualisasi *wordcloud* dapat dilihat topik yang sering dibahas pengguna aplikasi untuk memberikan ulasan negatif kategori *realibility* pada kinerja aplikasi. Pada gambar 20, *wordcloud* dengan pembahasan kata "buka", diikuti kata "baik" kata dasar "perbaiki" atau "memperbaiki" kemudian kata "pelihara" dari kata "pemeliharaan". Kata tersebut menunjukan bahwa aplikasi Mandiri Online memiliki ulasan negatif yaitu pada permasalahan *realibility* yaitu permasalahan dalam membuka aplikasi dan waktu pemeliharaan/*maintenance* sehingga aplikasi perlu diperbaiki.

5. Kesimpulan

Berdasarkan hasil pengujian dan analisis yang telah dilakukan maka dapat diambil kesimpulan sebagai berikut:

1. Klasifikasi sentimen menggunakan *Naive Bayes Classifier* dengan *feature selection Information Gain* akan menghasilkan akurasi dan f-measure lebih baik daripada tanpa *feature selection Information Gain* yaitu sebesar 91,33% dan 90,37% untuk ulasan Mandiri Online karena fitur-fitur diseleksi terlebih dahulu sebelum proses klasifikasi sehingga fitur-fitur yang kurang penting tidak digunakan pada proses klasifikasi dan untuk klasifikasi kategori menggunakan *Naive Bayes Classifier* tanpa dan dengan *feature selection information gain* tidak berpengaruh untuk proses klasifikasi kategori. Hal ini disebabkan karena perhitungan *feature selection* information gain hanya dilakukan berdasarkan kelas sentimen , yang

artinya *information gain* digunakan untuk mengukur nilai ketergantungan antara fitur atau kata dengan kelas sentimen dan tidak berlaku untuk nilai ketergantungan fitur dengan kelas kategori.

2. Dengan adanya proses klasifikasi serta visualisasi pada *wordcloud*, didapatkan informasi mengenai topik-topik yang paling sering dibicarakan oleh *user* berkaitan dengan kualitas Aplikasi Mandiri Online, sehingga pihak Bank Mandiri dapat dengan mudah mengetahui aspek kelebihan dan kekurangan produk aplikasinya dari sudut pandang konsumen secara tepat. Agar dapat segera dilakukan penanganan terhadap aspek kualitas dari aplikasi. Hal ini tentu diharapkan dapat meningkatkan kualitas kinerja aplikasi dan meningkatkan jumlah pelanggan bank Mandiri karena mudahnya pengaksesan transaksi bank secara online dengan menggunakan aplikasi Mandiri Online.

Saran

Adapun saran dari tugas akhir ini adalah:

1. Melakukan perhitungan *feature selection* information gain berdasarkan kelas kategori, karena untuk melihat pengaruh dari information gain terhadap klasifikasi kategori dapat dilakukan berdasarkan nilai ketergantungan antara fitur dengan kelas kategori tersebut.
2. Penelitian ini hanya mengklasifikasikan beberapa kategori dari standar kualitas perangkat lunak ISO 25010, sementara masih banyak metode lain (seperti dokumen ISO lainnya, IEEE, dan lain-lain). Penelitian selanjutnya perlu adanya implementasi kasus evaluasi kualitas perangkat lunak dengan menggunakan metode lain yang disesuaikan dengan kebutuhan yang diperlukan untuk mendapatkan penilaian secara menyeluruh dan mendalam, serta apabila melakukan pengklasifikasian manual berdasarkan standar kualitas perangkat lunak, disarankan untuk dilakukan validasi kategori oleh *expert* dibidangnya.

DAFTAR PUSTAKA

[1] Dennis Pagano, W. M. (2013). User Feedback in the Appstore: An Empirical Study.
[2] Hui Yang, P. L. (2015). Indentification and Classification of Requirements from App User Reviews.
[3] Abelein, U., Sharp, H., & Paech, B. (2013). Does Involving Users in Software Development Really Influence System Success?
[4] Mengmeng Lu, P. L. (2017). Automatic Classification of Non-Functional Requirements from Augmented App User Reviews.
[5] Manning, C. D., Raghavan, P., dan Schutze, H.. 2009. Introduction to Information Retrieval. England: Cambridge University Press.
[6] Deng, Houtao, and George Runger. 2012. "Feature Selection via Regularized Trees,"
[7] *ISO/IEC 25010*. (n.d.). Retrieved from ISO 25000: https://iso25000.com/index.php/en/iso-25000-standards/iso-25010
[8] Olson, D. L., dan Delen, D.. 2008. Advanced Data Mining Techniques. Berlin Heidelberg: Springer.
[9] Z Tala, F., 2003. A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia.
[10] Agusta, Ledy, "Perbandingan Algoritma Stemming Porter dengan Algoritma Nazief & Adriani untuk Stemming Dokumen Teks Bahasa Indonesia", Prosiding Konferensi Nasional Sistem dan Informatika, November 14th, 2009, Bali, Indonesia, pp: 196-201