

## Klasifikasi Ulasan Aplikasi Jenius pada *Google Play Store* Menggunakan Algoritma *Naive Bayes*

Raihan Adyatma Subagja<sup>1</sup>, Yuni Widiastiwi<sup>2</sup>, Nurul Chamidah<sup>3</sup>

<sup>1,2,3</sup>Program Studi Informatika, Universitas Pembangunan Nasional Veteran Jakarta  
<sup>1,2,3</sup>Jl. RS. Fatmawati Raya, Pd. Labu, Kec. Cilandak, Kota Depok, Jawa Barat 12450  
 email: <sup>1</sup>[raihaanas@upnvj.ac.id](mailto:raihaanas@upnvj.ac.id), <sup>2</sup>[widiastiwi@upnvj.ac.id](mailto:widiastiwi@upnvj.ac.id), <sup>3</sup>[nurul.chamidah@upnvj.ac.id](mailto:nurul.chamidah@upnvj.ac.id)

**Abstrak.** Dengan perkembangan teknologi, semakin banyak aplikasi inovatif mempermudah kebutuhan manusia, salah satunya aplikasi *m-banking*. Aplikasi *m-banking* memudahkan kebutuhan pengguna untuk transaksi perbankan dan membuat penggunaannya dapat melakukan transaksi langsung dari *smartphone*. Jenius merupakan kartu debit yang dirilis oleh Bank BTPN dalam bentuk aplikasi *m-banking*. Aplikasi Jenius telah diunduh sebanyak 5 juta kali di *Google Play Store*. Dengan banyaknya pengguna Jenius, pastinya banyak opini masyarakat terhadap aplikasi tersebut. Opini tersebut dapat disalurkan melalui ulasan aplikasi pada *Google Play Store*. Untuk mempermudah mengolah informasi yang didapat dari ulasan tersebut, diperlukan klasifikasi terhadap ulasan aplikasi Jenius yang terdapat pada *Google Play Store*. Algoritma klasifikasi yang digunakan pada penelitian ini adalah *Naive Bayes*. Data diambil dengan cara *scraping*. Data yang diambil sudah memiliki label sesuai dengan rating yang diberikan pengguna. Selanjutnya data ulasan yang terkumpul dibagi ke dalam data latih sebesar 70% dan data uji sebesar 30%. Penelitian ini menghasilkan nilai *accuracy* sebesar 57%.

**Kata kunci:** Klasifikasi Teks, Jenius, *Google Play Store*, *Naive Bayes*.

### 1. PENDAHULUAN

Perkembangan teknologi menyebabkan pertumbuhan pengguna *smartphone* semakin meningkat. Hal tersebut menyebabkan para *developer* aplikasi untuk mengembangkan berbagai aplikasi. Banyaknya aplikasi tersebut semakin memudahkan para pengguna dalam berbagai macam hal, salah satu aplikasi yang memudahkan pengguna adalah aplikasi *m-banking*. Aplikasi *m-banking* merupakan aplikasi yang dapat membantu pengguna dalam hal transaksi perbankan. Dengan adanya aplikasi *m-banking*, para pengguna tidak perlu repot untuk berkunjung ke *teller* bank atau ke ATM. Pada *Google Play Store*, terdapat beberapa aplikasi *m-banking*, salah satunya aplikasi Jenius.

Jenius adalah sebuah aplikasi perbankan digital, yang diluncurkan oleh Bank BTPN pada 11 Agustus 2016. Aplikasi ini membantu penggunaannya melakukan aktivitas finansial seperti menabung, bertransaksi, atau mengatur keuangan serta memungkinkan nasabah untuk memiliki rekening bank. Semua dilakukan dari satu tempat, dari ponsel, baik yang berbasis *iOS* maupun *Android*. [1] Di Indonesia, Jenius sudah mulai banyak digemari masyarakat, bisa dilihat pada *Google Play Store*, aplikasi Jenius telah diunduh oleh 5 juta pengguna. Jenius banyak diminati banyak orang karena kemudahannya dalam membuka sebuah rekening. Berbeda dengan umumnya membuka rekening di Bank yang harus tatap muka, membuka rekening Jenius dilakukan secara online. Hanya dengan mengunduh aplikasi Jenius, mengisi data diri, serta mengunggah foto dan KTP, semuanya dilakukan secara *online*. Setelah selesai mendaftar, untuk mengaktifkan kartu juga dilakukan secara online, melalui *video call*.

Dengan banyaknya pengguna Jenius, pastinya banyak opini-opini mengenai aplikasi tersebut. Opini tersebut bisa disalurkan melalui ulasan pada *google play store*, yang umumnya berisi berupa saran maupun keluhan. Dari ulasan-ulasan tersebut, bisa didapatkan berbagai macam informasi yang berkaitan dengan aplikasi tersebut. Untuk menghasilkan informasi yang didapat dari ulasan pengguna aplikasi Jenius, dibutuhkan pengolahan data pada ulasan yang diberikan pengguna aplikasi Jenius. Ulasan tersebut berguna untuk membantu *developer* untuk menyempurnakan aplikasi tersebut. Untuk mempermudah pengolahan data ulasan yang ada, maka diperlukan klasifikasi terhadap ulasan aplikasi Jenius yang terdapat pada *Google Play Store*.

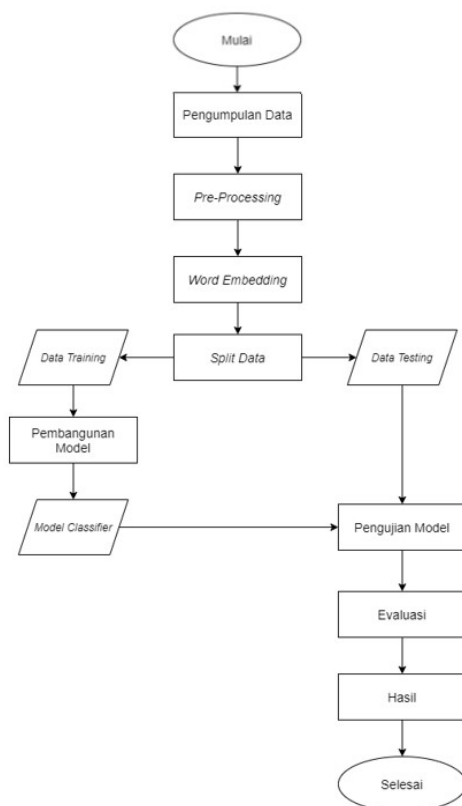
Data mining merupakan kombinasi beberapa cabang ilmu komputer yang merupakan sebuah proses menemukan pola pada suatu kumpulan data yang besar. Tujuan dari data mining yaitu mengambil intisari dari suatu pengetahuan untuk mengumpulkan data dan menghasilkan struktur data yang lebih mudah dimengerti. [2] Salah satu metode data mining adalah klasifikasi teks. Klasifikasi teks merupakan teknik dari *data mining* yang

digunakan untuk mencari atau mengatur *class* dalam sebuah dokumen, yang dibedakan dengan beberapa fungsi dengan tujuan untuk memungkinkan model dapat digunakan pada data pengujian. [3] Salah satu algoritma klasifikasi adalah *Naive Bayes*. Klasifikasi *Naive Bayes* adalah salah satu metode klasifikasi yang dapat digunakan untuk memprediksi probabilitas keanggotaan suatu class, klasifikasi ini didasarkan pada teorema *Bayes* yang memiliki akurasi dan kecepatan yang tinggi saat diimplementasikan kedalam database dengan dataset yang besar. [4] Teorema Bayesian adalah pendekatan statistik dasar untuk pengenalan pola. Teorema Bayes digunakan untuk melakukan perhitungan probabilitas dari suatu hipotesis dari probabilitas *prior* yang diberikan. Algoritma dasar untuk melakukan perhitungan probabilitas pada setiap kemungkinan hipotesis untuk penentuan kemungkinan terdekat dengan menggunakan Teorema Bayes. [5]

Terdapat banyak penelitian terdahulu yang bersangkutan dengan dilakukannya penelitian ini seperti penelitian [6] yang melakukan penelitian menerapkan metode klasifikasi *Naive Bayes* dan dikombinasikan dengan metode seleksi fitur *Information Gain* sebagai metode seleksi untuk memilih fitur mana saja yang berpengaruh untuk setiap label. Penelitian ini menghasilkan nilai akurasi sebesar 91,33%. Pada penelitian [7] dilakukan penelitian menggunakan metode *Naive Bayes* untuk mengklasifikasikan dokumen pada webiste pemerintah Kabupaten Semarang. Hasil penelitian ini menghasilkan nilai akurasi sebesar 85%. Pada penelitian [8] dilakukan penerapan algoritma klasifikasi *Naive Bayes*. Penelitian ini menghasilkan perbandingan antara ulasan positif dengan ulasan negatif. Pada penelitian [9] dilakukan penelitian menggunakan algoritma klasifikasi *Naive Bayes* yang dikombinasikan dengan *n-gram*. Penelitian ini diuji sebanyak lima kali, dengan akurasi tertinggi yang didapat sebesar 78,66%.

Dalam penelitian ini, teknik untuk pengambilan data pada *Google Play Store* menggunakan teknik *scraping*. Sedangkan untuk proses klasifikasi ulasan akan dilakukan dengan menggunakan metode *Naive Bayes* pada ulasan aplikasi *Jenius*. Harapannya dengan penelitian ini mampu mengklasifikasikan ulasan dengan baik, sehingga nantinya informasi yang terdapat di dalamnya bisa berguna bagi pihak yang membutuhkan.

## 2. METODE PENELITIAN



Gambar 1. Alur Penelitian

## 2.1 Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data untuk penelitian. Proses pengumpulan data dilakukan dengan teknik *scraping* pada *Google Play Store*. *Scraping* dilakukan dengan menggunakan *google-play-scraper*. Data ulasan diambil pada tanggal 1 November 2020 sampai dengan 1 Januari 2021. Data yang digunakan dalam penelitian ini diambil dengan cara *web scraping* menggunakan *python*, yang kemudian diuraikan sedemikian rupa sehingga dapat langsung diolah secara mudah. Data yang digunakan merupakan ulasan dan *rating* dari pengguna aplikasi tersebut dengan jumlah 1200 ulasan yang terdiri dari 200 ulasan dengan *rating* 1, 200 ulasan dengan *rating* 2, 400 ulasan dengan *rating* 3, 200 ulasan dengan *rating* 4, dan 200 ulasan dengan *rating* 5. *Class* yang digunakan dalam penelitian ini adalah *rating* atau nilai pada ulasan yang diberikan pengguna, dengan jumlah *class* sebanyak 5 *class*, yaitu satu sampai lima.

## 2.2 Pra-Proses Data

Setelah data didapatkan pada proses sebelumnya, dilakukan proses untuk menyiapkan data yang akan diolah nanti. Pada penelitian ini, dilakukan beberapa tahap *pre-processing* data, mulai dari *case folding*, *tokenizing*, *filtering*, *special character*, *punctuation and number removal*, dan *stemming*. Untuk pelabelan data, menggunakan *rating* dari ulasan yang diberikan pengguna terhadap aplikasi Jenius. Pelabelan dibagi ke dalam lima *class*, yaitu *class* dengan *rating* satu sampai lima. Data yang telah didapat sebelumnya akan dilakukan *pre-processing* dengan berbagai macam metode, diantaranya *case folding*, *special character and number removal*, *tokenizing*, *filtering*, *punctuation removal*, dan *stemming*.

### 2.2.1 Case Folding

Pada tahap ini, dilakukan praproses yang mengkonversikan seluruh isi dokumen teks ke dalam bentuk *lower case* atau huruf kecil

### 2.2.2 Tokenizing

Setelah melakukan *case folding*, selanjutnya akan dilakukan *tokenizing*. Pada tahap ini, teks yang telah selesai melewati *case folding* akan dilakukan proses pemecahan perkata.

### 2.2.3 Filtering

*Filtering* bertujuan untuk mengambil kata-kata penting pada *token* yang dihasilkan pada proses sebelumnya. Kata umum yang biasa muncul dalam jumlah besar dan dianggap tidak memiliki makna disebut *stopword*. Contoh *stopword* dalam bahasa Indonesia adalah “yang”, “di”, “dan”, “dari”, dan lain-lainnya.

### 2.2.4 Special Character, Punctuation, and Number Removal

Pada tahap *pre-processing* ini, teks yang berisi *non-alphabetic*, akan dihapus, contohnya seperti emoji, tanda baca, dan angka

### 2.2.5 Stemming

Pada tahap ini, dilakukan proses pemetaan dan penguraian bentuk dari suatu kata yang berimbuhan menjadi kata dasar

## 2.3 Word Embedding

Setelah data yang yang didapatkan telah melewati tahap praproses, selanjutnya dilakukan proses *word embedding*. *Word embedding* merupakan metode untuk memetakan kata atau frasa dari kosakata kedalam vektor yang sesuai dengan bilangan riil yang digunakan untuk menemukan prediksi kata, kesamaan kata atau semantik. [6] Metode *word embedding* yang digunakan pada penelitian ini adalah *CountVectorizer*. Metode ini mengubah fitur teks menjadi sebuah representasi vektor berdasarkan frekuensi kemunculan kata pada setiap teks. Contoh data yang akan dilakukan proses *word embedding* menggunakan *CountVectorizer* dapat dilihat pada Tabel 1.

**Tabel 1.** Sampel Data

Ulasan	Rating
aplikasi bagus mudah transaksi	5
fitur lengkap aplikasi sedikit lambat	4
banyak fitur lama aktivasi	3
fitur lengkap susah login	2
aktivasi akun lama	1

Berdasarkan Tabel 1, teks dalam data ulasan akan ditransformasikan ke dalam bentuk vektor menggunakan *CountVectorizer*, hasil transformasi dapat dilihat pada Tabel 2.

**Tabel 2.** Hasil Transformasi

	Dokumen				
	Dok 1	Dok 2	Dok 3	Dok 4	Dok 5
aktivasi	0	0	1	0	1
akun	0	0	0	0	1
aplikasi	1	1	0	0	0
bagus	1	0	0	0	0
banyak	0	0	1	0	0
fitur	0	1	1	1	0
lama	0	0	1	0	1
lambat	0	1	0	0	0
lengkap	0	1	0	1	0
login	0	0	0	1	0
mudah	1	0	0	0	0
sedikit	0	1	0	0	0
susah	0	0	0	1	0
transaksi	1	0	0	0	0

Tabel 2 merupakan hasil transformasi dari data dalam data ulasan yang sudah ditransformasikan ke dalam bentuk vektor menggunakan *CountVectorizer*.

## 2.4 Pembentukan Model dengan *Naïve Bayes*

Data yang telah melewati proses *word embedding* akan digunakan dalam pembentukan model. Pada tahap ini, proses pembentukan model dilakukan menggunakan algoritma *Naive Bayes Classifier*. Dengan pembagian data 70% sebagai data latih, dan 30% sebagai data uji. Proses pembentukan model dilakukan dengan menggunakan algoritma *Naive Bayes* dengan persamaan sebagai berikut: [7]

$$p(c|X) = \frac{p(X|c)p(c)}{p(X)} \quad (1)$$

Keterangan formula di atas sebagai berikut :

$x$  = Data dengan class yang belum diketahui

$c$  = Hipotesis data merupakan suatu class spesifik

$p(c|X)$  = Probabilitas berdasarkan kondisi (posterior probability)

$P(x|c)$  = Probabilitas berdasarkan kondisi pada hipotesis  
 $P(c)$  = Probabilitas hipotesis (prior probability)  
 $P(x)$  = Probabilitas x

Saat pembentukan model akan dihitung probabilitas kemunculan suatu dokumen pada *class* yang tersedia, dengan menggunakan persamaan sebagai berikut.[8]

$$P(c) = \frac{d|c|}{\text{TotalDokumen}} \quad (2)$$

Keterangan :

$P(c)$  : Probabilitas *class* C  
 $d|c|$  : Jumlah dokumen pada class C  
 C : Class 1 sampai 5

Setelah itu akan dihitung nilai probabilitas kemunculan setiap kata pada suatu *class* j dengan menggunakan persamaan berikut.[9]

$$P(w_i | c_j) = \frac{n(w_i, j) + 1}{|C| + n(\text{kosakata})} \quad (3)$$

Keterangan :

$P(w_i | c_j)$  : probabilitas dari kemunculan kata *wi* pada suatu *class* j  
 $n(w_i, j)$ : jumlah kemunculan kata *wi* dalam suatu *class* j  
 $|C|$  : semua kata yang terdapat pada *class* j  
 $n(\text{kosakata})$  : jumlah keseluruhan kata

Pada saat proses klasifikasi data ulasan, pendekatan metode ini akan menyaring *class* dengan nilai probabilitas tertinggi ( $c_{MAP}$ ) sesuai persamaan berikut:[10]

$$(c_{MAP}) = \text{argmax}_{c_i} P(c_j) \prod_i P(w_i | c_j) \quad (4)$$

Keterangan :

$P(c_j)$  : probabilitas munculnya suatu dokumen dalam *class* j  
 $P(w_i | c_j)$  : probabilitas munculnya kata *wi* pada *class* j

## 2.5 Evaluasi

Setelah pembentukan model menggunakan klasifikasi *Naive Bayes*, maka dilakukan evaluasi menggunakan *confusion matrix*. Biasanya, pada *confusion matrix* terdapat 4 representasi hasil dari proses klasifikasi, yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). [11] Namun pada penelitian ini, menggunakan 5 *class*, sehingga *confusion matrix* yang digunakan berbentuk 5x5. *Confusion matrix* yang digunakan dapat dilihat pada Tabel 3.

Tabel 3. Confusion Matrix

Aktual	Prediksi					
	Rating 5	Rating 4	Rating 3	Rating 2	Rating 1	Total
Rating 5	$TR_5$ (True R5)	$FR_{4a}$ (False R4)	$FR_{3a}$ (False R3)	$FR_{2a}$ (False R2)	$FR_{1a}$ (False R1)	$T_5$
Rating 4	$FR_{5a}$ (False R5)	$TR_4$ (True R4)	$FR_{3b}$ (False R3)	$FR_{2b}$ (False R2)	$FR_{1b}$ (False R1)	$T_4$
Rating 3	$FR_{5b}$ (False R5)	$FR_{4b}$ (False R4)	$TR_3$ (True R3)	$FR_{2c}$ (False R2)	$FR_{1c}$ (False R1)	$T_3$
Rating 2	$FR_{5c}$ (False R5)	$FR_{4c}$ (False R4)	$FR_{3c}$ (False R3)	$TR_2$ (True R2)	$FR_{1d}$ (False R1)	$T_2$
Rating 1	$FR_{5d}$ (False R5)	$FR_{4d}$ (False R4)	$FR_{3d}$ (False R3)	$FR_{2d}$ (False R2)	$FR_1$ (True R1)	$T_1$

Untuk menghitung *accuracy*, *precision*, dan *recall* dapat dihitung menggunakan rumus berikut.

$$accuracy = \frac{TR_5+TR_4+TR_3+TR_2+TR_1}{TR_5+TR_4+TR_3+TR_2+TR_1+\sum FR_{1a}+\sum FR_{1b}+\sum FR_{1c}+\sum FR_{1d}} \quad (5)$$

$$precision R_n = \frac{TR_n}{TR_n+FR_{n+1}+FR_{n+2}+FR_{n+3}+FR_{n+4}} \quad (6)$$

$$recall R_n = \frac{TR_n}{T_n} \quad (7)$$

### 3. HASIL DAN PEMBAHASAN

#### 3.1 Pembagian Data

Pada tahap ini, dilakukan pembagian data menjadi data latih dengan data uji. Metode pembagian data yang digunakan adalah metode *hold out validation*. *Hold out validation* merupakan metode pembagian data secara acak. Pada penelitian ini, komposisi pembagian data dibagi menjadi 70% untuk data latih dan 30% untuk data uji. Dengan presentase tersebut, 1200 data dibagi menjadi 840 data sebagai data latih, dan 360 data sebagai data uji. Jumlah *class* yang ada pada data latih dan data uji dapat dilihat pada Tabel 4.

Tabel 4. Rincian Data Latih dan Data Uji

	Rating 5	Rating 4	Rating 3	Rating 2	Rating 1	Jumlah
Data Latih	140	140	280	140	140	840
Data Uji	60	60	120	60	60	360

Berdasarkan Tabel 4 data latih untuk *class* dengan *rating* 5 terdapat sebanyak 140 data, *class* dengan *rating* 4 sebanyak 140 data, *class* dengan *rating* 3 sebanyak 280 data, *class* dengan *rating* 2 sebanyak 140 data, dan *class* dengan *rating* 1 sebanyak 140 data. Sedangkan data uji untuk *class* dengan *rating* 5 terdapat sebanyak 60 data, *class* dengan *rating* 4 sebanyak 60 data, *class* dengan *rating* 3 sebanyak 120 data, *class* dengan *rating* 2 sebanyak 60 data, dan *class* dengan *rating* 1 sebanyak 60 data.

#### 3.2 Pembentukan Model

Berdasarkan banyaknya sampel data yang terdapat pada Tabel 4, data tersebut akan diolah dalam proses *training*, yang bertujuan untuk membentuk sebuah model dengan algoritma *Naive Bayes* menggunakan persamaan (1), (2) dan (3). Model *Naive Bayes* tersebut nantinya akan digunakan pada proses *testing* dan evaluasi. Untuk sampel

data *training* yang sudah di praproses, dan sudah ditransformasi akan digunakan pada pembangunan model, data yang sudah ditransformasi dapat dilihat pada Tabel 1.

Langkah pertama dalam melakukan proses *training* dengan *Naive Bayes* yaitu menghitung *prior probability* dengan menggunakan persamaan (2) yang akan diterapkan pada sampel data yang terdapat pada Tabel 2.

1. Probabilitas untuk *class 5*

$$P(5) = \frac{d|5|}{\text{Total dokumen}}$$

$$P(5) = \frac{1}{5}$$

$$P(5) = 0.2$$

2. Probabilitas untuk *class 4*

$$P(4) = \frac{d|4|}{\text{Total dokumen}}$$

$$P(4) = \frac{1}{5}$$

$$P(4) = 0.2$$

3. Probabilitas untuk *class 3*

$$P(3) = \frac{d|3|}{\text{Total dokumen}}$$

$$P(3) = \frac{1}{5}$$

$$P(3) = 0.2$$

4. Probabilitas untuk *class 2*

$$P(2) = \frac{d|2|}{\text{Total dokumen}}$$

$$P(2) = \frac{1}{5}$$

$$P(2) = 0.2$$

5. Probabilitas untuk *class 1*

$$P(1) = \frac{d|1|}{\text{Total dokumen}}$$

$$P(1) = \frac{1}{5}$$

$$P(1) = 0.2$$

Setelah mendapat nilai probabilitas setiap class, langkah selanjutnya adalah menghitung nilai probabilitas setiap kata berdasarkan *class* menggunakan persamaan (3). Berdasarkan rincian pada Tabel 3, diketahui :

$n(\text{kosakata}) = 14$   
 Kata pada *class 5* = 4  
 Kata pada *class 4* = 4  
 Kata pada *class 3* = 3  
 Kata pada *class 2* = 2  
 Kata pada *class 1* = 1

Probabilitas kata “aktivasi” berdasarkan frekuensi kemunculan kata pada Tabel 4

$$P(\text{"aktivasi"}|5) = \frac{(\text{"aktivasi"}|5)+1}{|5|+n(\text{kosakata})}$$

$$= \frac{0+1}{4+14}$$

$$= 0.056$$

$$P(\text{"aktifasi"} | \text{"4"}) = \frac{(\text{"aktifasi"} | \text{"4"}) + 1}{|\text{"4"}| + n(\text{kosakata})}$$

$$= \frac{0 + 1}{4 + 14}$$

$$= 0.056$$

$$P(\text{"aktifasi"} | \text{"3"}) = \frac{(\text{"aktifasi"} | \text{"3"}) + 1}{|\text{"3"}| + n(\text{kosakata})}$$

$$= \frac{1 + 1}{3 + 14}$$

$$= 0.12$$

$$P(\text{"aktifasi"} | \text{"2"}) = \frac{(\text{"aktifasi"} | \text{"2"}) + 1}{|\text{"2"}| + n(\text{kosakata})}$$

$$= \frac{0 + 1}{2 + 14}$$

$$= 0.0625$$

$$P(\text{"aktifasi"} | \text{"1"}) = \frac{(\text{"aktifasi"} | \text{"1"}) + 1}{|\text{"1"}| + n(\text{kosakata})}$$

$$= \frac{1 + 1}{1 + 14}$$

$$= 0.13$$

Perhitungan diatas merupakan contoh untuk mendapatkan hasil probabilitas setiap kata berdasarkan frekuensi kemunculan kata pada Tabel 4. Hasil seluruh probabilitas setiap kata yang dihitung menggunakan persamaan (3) dapat dilihat pada Tabel 5.

**Tabel 5.** Hasil Probabilitas Tiap Kata

$P(w_i   c_j)$	Hasil	$P(w_i   c_j)$	Hasil
$P(\text{"aplikasi"}   \text{"5"})$	0.11	$P(\text{"lengkap"}   \text{"5"})$	0.056
$P(\text{"aplikasi"}   \text{"4"})$	0.11	$P(\text{"lengkap"}   \text{"4"})$	0.11
$P(\text{"aplikasi"}   \text{"3"})$	0.058	$P(\text{"lengkap"}   \text{"3"})$	0.058
$P(\text{"aplikasi"}   \text{"2"})$	0.0625	$P(\text{"lengkap"}   \text{"2"})$	0.125
$P(\text{"aplikasi"}   \text{"1"})$	0.067	$P(\text{"lengkap"}   \text{"1"})$	0.067
$P(\text{"bagus"}   \text{"5"})$	0.11	$P(\text{"login"}   \text{"5"})$	0.056
$P(\text{"bagus"}   \text{"4"})$	0.056	$P(\text{"login"}   \text{"4"})$	0.056
$P(\text{"bagus"}   \text{"3"})$	0.058	$P(\text{"login"}   \text{"3"})$	0.058
$P(\text{"bagus"}   \text{"2"})$	0.0625	$P(\text{"login"}   \text{"2"})$	0.125
$P(\text{"bagus"}   \text{"1"})$	0.067	$P(\text{"login"}   \text{"1"})$	0.067
$P(\text{"banyak"}   \text{"5"})$	0.056	$P(\text{"mudah"}   \text{"5"})$	0.11
$P(\text{"banyak"}   \text{"4"})$	0.056	$P(\text{"mudah"}   \text{"4"})$	0.056
$P(\text{"banyak"}   \text{"3"})$	0.12	$P(\text{"mudah"}   \text{"3"})$	0.058
$P(\text{"banyak"}   \text{"2"})$	0.0625	$P(\text{"mudah"}   \text{"2"})$	0.0625
$P(\text{"banyak"}   \text{"1"})$	0.067	$P(\text{"mudah"}   \text{"1"})$	0.067



$P(\text{"fitur"}   \text{"5"})$	0.056	$P(\text{"sedikit"}   \text{"5"})$	0.056
$P(\text{"fitur"}   \text{"4"})$	0.11	$P(\text{"sedikit"}   \text{"4"})$	0.11
$P(\text{"fitur"}   \text{"3"})$	0.12	$P(\text{"sedikit"}   \text{"3"})$	0.058
$P(\text{"fitur"}   \text{"2"})$	0.125	$P(\text{"sedikit"}   \text{"2"})$	0.0625
$P(\text{"fitur"}   \text{"1"})$	0.067	$P(\text{"sedikit"}   \text{"1"})$	0.067
$P(\text{"lama"}   \text{"5"})$	0.056	$P(\text{"susah"}   \text{"5"})$	0.056
$P(\text{"lama"}   \text{"4"})$	0.056	$P(\text{"susah"}   \text{"4"})$	0.056
$P(\text{"lama"}   \text{"3"})$	0.12	$P(\text{"susah"}   \text{"3"})$	0.058
$P(\text{"lama"}   \text{"2"})$	0.0625	$P(\text{"susah"}   \text{"2"})$	0.125
$P(\text{"lama"}   \text{"1"})$	0.13	$P(\text{"susah"}   \text{"1"})$	0.067
$P(\text{"lambat"}   \text{"5"})$	0.056	$P(\text{"transaksi"}   \text{"5"})$	0.11
$P(\text{"lambat"}   \text{"4"})$	0.11	$P(\text{"transaksi"}   \text{"4"})$	0.056
$P(\text{"lambat"}   \text{"3"})$	0.058	$P(\text{"transaksi"}   \text{"3"})$	0.058
$P(\text{"lambat"}   \text{"2"})$	0.0625	$P(\text{"transaksi"}   \text{"2"})$	0.0625
$P(\text{"lambat"}   \text{"1"})$	0.067	$P(\text{"transaksi"}   \text{"1"})$	0.067

Tabel berikut merupakan hasil perhitungan untuk mendapatkan hasil probabilitas setiap kata berdasarkan frekuensi kemunculan kata seluruh probabilitas setiap kata yang dihitung menggunakan persamaan (3).

### 3.3 Pengujian Model

Pengujian model dilakukan setelah pembentukan model selesai. Pada tahap pengujian model, data yang digunakan adalah data uji. Untuk sampel data uji yang sudah dilakukan pra proses dapat dilihat pada Tabel 6.

Tabel 6. Sampel Data Uji

Ulasan	Rating
aplikasi bagus mudah transaksi	?

Sampel data uji yang terdapat pada Tabel 6 dapat diklasifikasikan menggunakan persamaan (4).

$$\begin{aligned}
 &1. \quad P(5|Data\ Uji) \\
 &P(5|Data\ Uji) = P(5) \times P(\text{aplikasi}|5) \times P(\text{bagus}|5) \times \\
 &P(\text{mudah}|5) \times P(\text{transaksi}|5) \\
 &= 0.2 \times 0.11 \times 0.11 \times 0.11 \times 0.11 \\
 &= \mathbf{2.928 \times 10^{-5}}
 \end{aligned}$$

$$\begin{aligned}
 &2. \quad P(4|Data\ Uji) \\
 &P(4|Data\ Uji) = P(4) \times P(\text{aplikasi}|4) \times P(\text{bagus}|4) \times \\
 &P(\text{mudah}|4) \times P(\text{transaksi}|4) \\
 &= 0.2 \times 0.11 \times 0.056 \times 0.0625 \times 0.067 \\
 &= \mathbf{5.159 \times 10^{-6}}
 \end{aligned}$$

$$\begin{aligned}
 &3. \quad P(3|Data\ Uji) \\
 &P(3|Data\ Uji) = P(3) \times P(aplikasi|3) \times P(bagus|3) \times \\
 &P(mudah|3) \times P(transaksi|3) \\
 &= 0.2 \times 0.058 \times 0.058 \times 0.058 \times 0.058 \\
 &= 2.263 \times 10^{-6}
 \end{aligned}$$

$$\begin{aligned}
 &4. \quad P(2|Data\ Uji) \\
 &P(2|Data\ Uji) = P(2) \times P(aplikasi|2) \times P(bagus|2) \times \\
 &P(mudah|2) \times P(transaksi|2) \\
 &= 0.2 \times 0.0625 \times 0.0625 \times 0.0625 \times 0.0625 \\
 &= 3.052 \times 10^{-6}
 \end{aligned}$$

$$\begin{aligned}
 &5. \quad P(1|Data\ Uji) \\
 &P(1|Data\ Uji) = P(1) \times P(aplikasi|1) \times P(bagus|1) \times \\
 &P(mudah|1) \times P(transaksi|1) \\
 &= 0.2 \times 0.067 \times 0.067 \times 0.067 \times 0.067 \\
 &= 4.030 \times 10^{-6}
 \end{aligned}$$

Berdasarkan pengujian yang dilakukan, hasil probabilitas tertinggi adalah  $2.928 \times 10^{-5}$ . Hasil tersebut merupakan nilai probabilitas dari  $P(5|Data\ Uji)$ , sehingga data uji yang digunakan dapat diklasifikasikan kedalam *class* 5.

### 3.4 Evaluasi Hasil

Setelah model berhasil dibentuk, selanjutnya uji model tersebut menggunakan data uji, hasil pengujian tersebut dapat dilihat pada Tabel 7.

Tabel 7. Hasil *Confusion Matrix*

Aktual	Prediksi				
	Rating 5	Rating 4	Rating 3	Rating 2	Rating 1
Rating 5	30	8	23	3	2
Rating 4	11	17	30	1	3
Rating 3	9	5	87	1	1
Rating 2	4	4	31	11	11
Rating 1	0	0	5	4	59

Dengan nilai pada Tabel 7 diatas, dapat dihitung *Accuracy* dengan persamaan (5), *Precision* dengan persamaan (6), dan *Recall* dengan persamaan (7) sebagai berikut :

$$accuracy = \frac{30 + 17 + 87 + 11 + 59}{30 + 17 + 87 + 11 + 59 + 24 + 17 + 89 + 9 + 17} = \frac{204}{360} = 0,57$$

$$precision\ P_5 = \frac{30 + 11 + 9 + 4 + 0}{30} = \frac{54}{30} = 0,56$$

$$precision\ P_4 = \frac{17 + 8 + 5 + 4 + 0}{17} = \frac{34}{17} = 0,5$$

$$precision\ P_3 = \frac{87 + 23 + 30 + 31 + 5}{87} = \frac{176}{87} = 0,49$$

$$precision\ P_2 = \frac{11 + 3 + 1 + 1 + 4}{11} = \frac{20}{11} = 0,55$$

$$\begin{aligned}
 \text{precision } R_1 &= \frac{59}{59 + 2 + 3 + 1 + 11} = \frac{59}{86} = 0,78 \\
 \text{recall } R_5 &= \frac{30}{30 + 8 + 23 + 3 + 2} = \frac{30}{76} = 0,45 \\
 \text{recall } R_4 &= \frac{17}{17 + 11 + 30 + 1 + 3} = \frac{17}{62} = 0,27 \\
 \text{recall } R_3 &= \frac{11}{87 + 9 + 5 + 1 + 1} = \frac{11}{103} = 0,84 \\
 \text{recall } R_2 &= \frac{11}{11 + 4 + 4 + 31 + 11} = \frac{11}{59} = 0,10 \\
 \text{recall } R_1 &= \frac{59}{59 + 0 + 0 + 5 + 4} = \frac{59}{68} = 0,87
 \end{aligned}$$

Berdasarkan evaluasi yang dilakukan untuk melihat seberapa baik performa model yang dibentuk menggunakan algoritma *Naive Bayes*, didapatkan hasil akurasi sebesar 0.57, sementara untuk hasil *precision* dan *recall* pada tiap *class* dapat dilihat pada Tabel 8.

Tabel 8. Hasil *Precision* dan *Recall*

	R5	R4	R3	R2	R1
<i>Precision</i>	0.56	0.5	0.49	0.55	0.78
<i>Recall</i>	0.45	0.27	0.84	0.18	0.87

Berdasarkan Tabel 8, nilai *precision* dan *recall* terbesar ada pada *class* dengan *Rating* bintang 1, sebesar 0.78 dan 0.87.

#### 4. KESIMPULAN DAN SARAN

Berdasarkan penelitian yang telah dilakukan dengan melakukan klasifikasi ulasan pengguna terhadap aplikasi Jenius, dapat disimpulkan bahwa Data yang telah diambil dengan cara *scraping* pada *google play store*, dilakukan pra proses data dengan melakukan *case folding*, *special character and number removal*, *tokenizing*, *filtering*, *punctuation removal*, *stemming* dan *word embedding*, yang kemudian dibagi kedalam 70% data untuk membangun model *Naive Bayes*, dan 30% untuk menguji model tersebut. Pengujian model *Naive Bayes Classifier* menghasilkan akurasi sebesar 0.57, hasil *precision* dan *recall* terbesar terdapat pada *class rating* 1, sebesar 0,78 dan 0,87. Performa dari *Naive Bayes Classifier* dalam mengklasifikasikan ulasan pengguna pada aplikasi Jenius kurang baik, karena akurasi yang dihasilkan relatif kecil

Saran untuk penelitian ini diantaranya menambahkan data latih dan data uji lebih banyak lagi, menambahkan metode lain dalam pra proses data, seperti normalisasi kata agar hasil yang didapatkan lebih optimal, menggunakan metode *word embedding* lainnya, seperti TF-IDF, Word2Vec, atau yang lainnya, serta luaran bisa dikembangkan menjadi sebuah aplikasi yang dapat memprediksi atau merekomendasikan *rating* ulasan berdasarkan ulasan yang ditulis.

#### 5. DAFTAR PUSTAKA

- [1] Liputan 6 (11 Agustus 2016). BTPN Luncurkan Jenius, Aplikasi Cerdas untuk Mengatur Keuangan, dari <https://www.liputan6.com/bisnis/read/2575011/btpn-luncurkan-jenius-aplikasi-cerdas-untuk-mengatur-keuangan>
- [2] Wijaya, A. P., & Santoso, H. A. (2016). Naive Bayes Classification pada Klasifikasi Dokumen Untuk Identifikasi Konten E-Government Naive Bayes Classification on Document Classification to Identify E-Government Content. *Journal of Applied Intelligent System*, 1(1), 48–55.
- [3] Suyanto. (2017). *Data Mining Untuk Klasifikasi dan Klasterisasi Data*. Penerbit Informatika.
- [4] Fitri, E., Yuliani, Y., Rosyida, S., & Gata, W. (2020). Analisis Sentimen Terhadap Aplikasi Ruangguru Menggunakan Algoritma Naive Bayes, Random Forest Dan Support Vector Machine. *Jurnal Transformatika*, 18(1), 71. <https://doi.org/10.26623/transformatika.v18i1.2317>
- [5] Prasetyowati, E. (2017). *DATA MINING Pengelompokan Data untuk Informasi dan Evaluasi*. Duta Media Publishing.
- [6] Sari, A. E., Widowati, S., & Lhaksana, K. M. (2019). *Klasifikasi Ulasan Pengguna Aplikasi Mandiri Online di*

- Google Play Store dengan Menggunakan Metode Information Gain dan Naive Bayes Classifier. E-Proceeding of Engineering, 6(2), 9143–9157.
- [7] Wijaya, A. P., & Santoso, H. A. (2016). Naive Bayes Classification pada Klasifikasi Dokumen Untuk Identifikasi Konten E-Government Naïve Bayes Classification on Document Classification to Identify E-Government Content. *Journal of Applied Intelligent System*, 1(1), 48–55.
- [8] Hidayat, M. A. J., Primajati, G., & Amrullah, A. Z. (2020). Klasifikasi Data Ulasan Positif dan Negatif Dengan Menggunakan Algoritma Naive Bayes. *Jurnal*, 2(1), 52–56. <https://doi.org/10.30812/bite.v2i1.809>
- [9] Chandra, D. N., Indrawan, G., & Sukajaya, I. N. (2019). Klasifikasi Berita Lokal Radar Malang Menggunakan Naïve Bayes Dengan Fitur N-Gram. 10(2), 11–19.
- [10] Trianto, R. B., Triyono, A., & Arum, D. M. P. (2020). Klasifikasi Rating Otomatis pada Dokumen Teks Ulasan Produk Elektronik Menggunakan Metode N-gram dan Naïve Bayes. *Jurnal Informatika Universitas Pamulang*, 5(3), 295. <https://doi.org/10.32493/informatika.v5i3.6110>
- [11] Herlambang, Admaja., & Wijoyo, Satrio. (2019) . Algoritma Naïve Bayes Untuk Klasifikasi Sumber Belajar Berbasis Teks Pada Mata Pelajaran Produktif Di Smk Rumpun Teknologi Informasi Dan Komunikasi. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, 6(4), 431-436.