

OPINION MINING UNTUK ULASAN PRODUK DENGAN KLASIFIKASI NAIVE BAYES

Albert Jeremy¹⁾ Viny Christanti M²⁾ Bagus Mulyawan³⁾

Teknik Informatika, Fakultas Teknologi Informasi Universitas Tarumanagara
Jalan Letjen S Parman 1, Jakarta Barat 11440, Indonesia
email : albert.jeey@gmail.com

Teknik Informatika, Fakultas Teknologi Informasi Universitas Tarumanagara
Jalan Letjen S Parman 1, Jakarta Barat 11440, Indonesia
email : viny@untar.ac.id

Teknik Informatika, Fakultas Teknologi Informasi Universitas Tarumanagara
Jalan Letjen S Parman 1, Jakarta Barat 11440, Indonesia
email : bagus@untar.ac.id

ABSTRACT

Nowadays, micro blogs have become the most used tools for users to share many things: from just updating things to telling their conditions or thoughts. Some popular micro blogs mostly used to give comments and opinions are facebook, instagram, and twitter. Twitter has 259 million active users each month as for January until April 2017. This made twitter one of the best micro blogs to know the most updated opinions.

The system uses Naive Bayes Classification to classify opinions about smartphone and computer from twitter. The sentiments are divided to positive, neutral, and negative. After that, Confusion Matrix is used to evaluate the algorithm and count the accuracy. Naive Bayes Classification gives 77.7% accuracy for Unigram, 50.7% for Bigram, and 31.7% for Trigram.

Key words

Confusion Matrix, Naive Bayes, Opinion Mining, R Language, Sentiment Analysis.

1 Pendahuluan

Perkembangan internet saat ini telah membuat media sosial menjadi salah satu tempat yang paling sering digunakan pengguna untuk menyampaikan banyak hal, mulai dari sekadar *update status* hingga curhatan mengenai kondisi atau pemikiran pengguna saat itu juga. *Twitter* adalah media sosial yang dapat digunakan untuk menyampaikan pesan-pesan pendek dengan maksimal 140 karakter yang biasa disebut *tweet*.

Pengguna aktif *twitter* pada bulan Januari sampai April tahun 2017 adalah 259 juta orang per bulan yang merupakan kenaikan hingga 60% dari jumlah pengguna aktif pada tahun 2016 [1].

Sistem yang dirancang berguna untuk menggali dan menganalisis sentimen dari kalimat opini atau komentar pada produk *smartphone* dan komputer di *twitter*. Metode yang digunakan untuk mengolah dan melakukan klasifikasi pada data *twitter* adalah Klasifikasi *Naive Bayes*. Sistem analisis sentimen *twitter* yang dirancang sebelumnya oleh Nuke Y. A. Faradhillah et. al. menggunakan *Naive Bayes* memberikan tingkat akurasi sebesar 78,77% [2].

2 Dasar Teori

Beberapa teori dan konsep yang digunakan dalam perancangan sistem adalah *Opinion Mining*, *Pre-processing* dan Klasifikasi *Naive Bayes*.

2.1 *Opinion Mining*

Opinion Mining merupakan sebuah bidang pembelajaran yang menganalisis opini, sentimen, penilaian, sikap, dan emosi terhadap suatu entitas seperti produk, pelayanan, organisasi, peristiwa, dan lain-lain. Istilah *opinion mining* pertama kali muncul dalam penelitian Daye et al. (2003), tetapi penelitian mengenai hal ini telah muncul sejak tahun 2000, hanya saja pada saat itu belum terdapat istilah *opinion mining*[3]. Secara garis besar, opini diklasifikasikan menjadi tiga kelas,

yaitu positif, negatif, dan netral. Data-data yang digunakan untuk melakukan *opinion mining* biasanya diambil dari *blog* (*blogspot*, *wordpress*, dll.) dan juga *microblog* (*twitter*, *facebook*, *youtube*, dan lain-lain) [3].

2.2 Pre-processing

Pre-processing adalah salah satu tahap awal yang harus dikerjakan sebelum teks dapat diproses lebih lanjut. Kumpulan data teks sering kali sulit untuk diproses karena beberapa faktor seperti banyaknya tanda baca, pengulangan kata-kata yang tidak terlalu penting atau tidak relevan, dan lain-lain. Tahapan-tahapan yang ada dalam *pre-processing*: [4]

1. Lexical Analysis

Lexical Analysis merupakan tahap yang mengubah rangkaian kalimat menjadi kata per kata atau token. Lalu, semua huruf pada teks diubah menjadi huruf kecil dan segala tanda baca dihilangkan. Untuk angka, terdapat algoritma yang mengizinkan angka untuk menjadi token (tidak dihilangkan) dan ada algoritma yang menghilangkan angka, bergantung pada kebutuhan data yang akan diproses [5].

2. Stopword Removal

Pada tahap ini, setiap kata yang dianggap tidak terlalu penting dibuang atau tidak digunakan untuk proses selanjutnya. Hal ini berguna untuk mengurangi beban proses dengan harapan tidak mengganggu atau merusak makna dari teks yang sudah dihilangkan *stopword*-nya. Contoh beberapa *stopword* dalam bahasa Inggris adalah *the*, *as*, *be*, *and*, dan lain-lain [5].

3. Stemming

Banyak kata yang merupakan penurunan dari kata dasar yang sama dan dapat dikatakan memiliki konsep arti kata yang sama [6]. Sebagai contoh kata *stem*, *stemmer*, *stemming*, *stemmed*, dan sebagainya. Penurunan kata ini dihasilkan dengan menambahkan imbuhan (awalan, sisipan, dan akhiran) pada sebuah kata dasar. Karena itu salah satu tujuan *stemming* adalah mengubah tiap kata menjadi kata dasarnya sehingga mempermudah proses-proses lainnya. Tujuan lain dari *stemming* berhubungan langsung dengan proses *information retrieval*, karena dengan memiliki kata-kata dasar akan meningkatkan performa pada beberapa tahap *information retrieval*. Contohnya adalah mempermudah pengindeksan dokumen-dokumen sesuai dengan topik karena dapat dibagi sesuai dengan kata dasarnya [6].

2.3 Klasifikasi Naive Bayes

Klasifikasi *Naive Bayes* merupakan salah satu metode dari *machine learning*. *Naive Bayes* merupakan metode klasifikasi berdasarkan teorema *Bayes* yang banyak digunakan. Metode ini mengasumsikan bahwa pengaruh nilai atribut pada suatu kelas adalah independen terhadap nilai dari atribut-atribut lain. Berikut adalah teorema *Bayes*: [7]

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)} \quad (1)$$

Dengan mengasumsikan bahwa keberadaan satu fitur tidak bergantung pada fitur-fitur lain: [7]

$$P(x_{1-n}) = P(x_1, x_2, x_3, \dots, x_n) \quad (2)$$

Didapati persamaan sebagai berikut:

$$P(y|x_1, \dots, x_n) = \frac{P(x_1, \dots, x_n)P(y)}{P(x_1, \dots, x_n)} \quad (3)$$

Dalam klasifikasi *Naive Bayes*, rumus menghitung probabilitas untuk melakukan klasifikasi adalah sebagai berikut:[8]

$$P(c|d) = P(c) \prod_{1 \leq k \leq n_d} P(t_k|c) \quad (4)$$

Keterangan :

- c : kelas untuk data yang ingin diklasifikasi
- d : data yang ingin diklasifikasi
- t_k : daftar kata-kata yang berbeda dalam pelatihan
- n_d : jumlah dari kata t_k
- $P(c)$: probabilitas *prior* data kelas c
- $P(t_k|c)$: probabilitas kondisional kata t_k kelas c

Berikut adalah tahap-tahap Klasifikasi *Naive Bayes* untuk mengklasifikasi dan memberikan probabilitas pada data: [8]

1. Melakukan perhitungan *prior* untuk tiap kelas. Ini dapat dilakukan dengan persamaan:

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

Keterangan :

- N_c : jumlah data dalam kelas c
 - N : jumlah data dalam pelatihan
2. Melakukan perhitungan probabilitas kondisional pada tiap kata untuk tiap kelas. Ini dapat dilakukan dengan persamaan:

$$P_{(t|c)} = \frac{T_{ct}+1}{\sum_{t' \in V} (T_{ct'}+B')} \quad (6)$$

Keterangan :

- T_{ct} : jumlah kemunculan kata t dalam data kelas c
- V : tiap kata dalam data latih

B' : jumlah kata dalam V

- Melakukan perhitungan probabilitas kata dari data uji pada tiap kelas.

Semakin tinggi nilai probabilitas data uji terhadap suatu kelas, data tersebut dianggap semakin relevan dengan kelas itu.

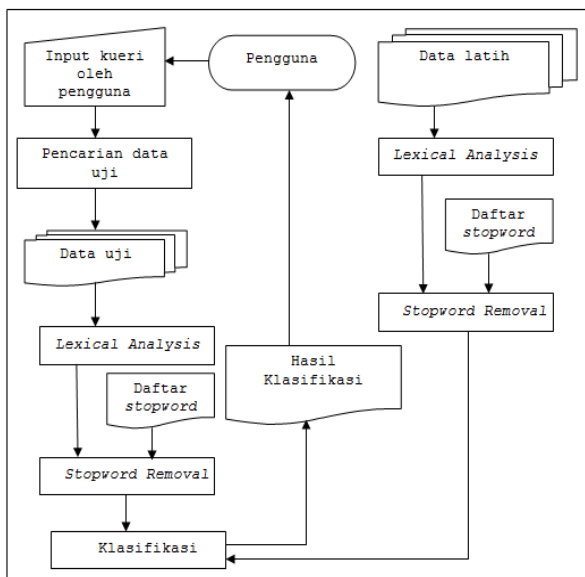
2.4 Bahasa Pemrograman R

R merupakan bahasa pemrograman yang memiliki keunggulan dalam perhitungan statistik, eksplorasi data, analisis, dan visualisasi. Kegunaan utama bahasa R adalah untuk memanipulasi dan menganalisis data statistik. Bahasa pemrograman R biasanya digunakan untuk melakukan *text processing*, *opinion mining*, *web scrapping*, *statistical analysis*, dan lain-lain [9].

3 Rancangan dan Pembuatan

3.1 Rancangan Sistem

Sistem yang dirancang adalah aplikasi yang melakukan *crawling* pada komentar-komentar mengenai produk *smartphone* dan *laptop* pada *twitter*, lalu dengan menggunakan Klasifikasi *Naive Bayes* yang telah dilatih, data-data dari *twitter* tersebut akan diklasifikasikan untuk menentukan jumlah sentimen positif, netral, dan negatif untuk komentar-komentar tersebut. Rancangan sistem dapat dilihat pada Gambar 1.



Gambar 1 Rancangan Sistem *Opinion Mining*

3.1.1 Perencanaan

Perencanaan merupakan tahap awal dalam merancang atau mengembangkan sebuah sistem. Tahap ini bertujuan untuk mendefinisikan tujuan dan ruang lingkup dari sistem yang akan dibuat. Tahap ini juga dapat mengidentifikasi masalah-masalah apa yang dapat diselesaikan oleh sistem.

Rancangan yang dibuat adalah sebuah aplikasi yang dapat menganalisis data dari *twitter* mengenai opini terhadap *smartphone* dan *laptop*. Data yang dianalisis berdasarkan tipe *smartphone* dan *laptop* sebagai *input* dari *end user*. Hasil analisis opini dari *twitter* ini dapat digunakan baik oleh produsen produk-produk tersebut untuk memantau dan mengetahui respon masyarakat terhadap produknya, maupun orang lain yang ingin mengetahui produk apa yang sedang menerima banyak sentimen positif atau negatif. Metode yang digunakan adalah Klasifikasi *Naive Bayes*. Klasifikasi ini akan mengelompokkan data-data dari *twitter* ke kelompok sentimen yang sesuai (positif, netral, atau negatif).

Pertama-tama, dilakukan *crawling* data dari *twitter*. Data-data tersebut kemudian diberi label secara manual, apakah itu merupakan *tweet* yang memiliki sentimen positif, netral, atau negatif. Lalu dalam tahap *pre-processing*, akan lakukan *lexical analysis* dan *stopword removal*. *Lexical Analysis* mengubah semua huruf besar menjadi huruf kecil dan menghilangkan tanda-tanda baca, sedangkan *stopword removal* menghilangkan kata-kata yang dianggap tidak diperlukan atau tidak penting. Data uji juga akan melewati *pre-processing*. Lalu, Klasifikasi *Naive Bayes* dilatih menggunakan data latih yang sudah diberi label.

Selanjutnya adalah pengklasifikasian data uji dengan menggunakan metode Klasifikasi *Naive Bayes*. Pada tahap ini, akan dihitung probabilitas kata-kata pada tiap *tweet* untuk tiap kelas. Lalu, data-data tersebut akan dikelompokkan sesuai dengan nilai probabilitasnya yang tertinggi. Hasil analisisnya akan ditampilkan dalam bentuk grafik.

3.1.2 Analisis

Pada tahap ini dilakukan analisis terhadap tiap metode yang akan digunakan dalam aplikasi yaitu Klasifikasi *Naive Bayes*. Selain itu, juga dilakukan analisis terhadap spesifikasi perangkat lunak dan perangkat keras yang digunakan dalam merancang aplikasi.

Hasil analisis pada metode Klasifikasi *Naive Bayes* menunjukkan bahwa metode ini memiliki pendekatan yang baik dalam pemberian nilai probabilitas data uji dan data latih. Dalam sebuah jurnal yang meneliti klasifikasi sentimen pada *Big Data* menggunakan Klasifikasi *Naive Bayes*, akurasi klasifikasi ini meningkat dan mencapai 82% saat jumlah data latihnya ditingkatkan [10]. Dalam penelitian lain yang melakukan klasifikasi untuk dokumen, Klasifikasi *Naive Bayes* berhasil mengklasifikasikan 2717 dokumen dengan benar dengan akurasi mencapai 97% [11].

Analisis selanjutnya adalah spesifikasi perangkat lunak dan perangkat keras yang digunakan dalam merancang aplikasi. Spesifikasi perangkat lunak sebagai berikut:

1. Sistem Operasi Windows 7 Professional 32-bit
2. Search Engine Mozilla Firefox
3. Enginesite Perl Editor
4. Visual Studio 2012
5. RStudio
6. Microsoft Office Excel 2017
7. Notepad

Spesifikasi perangkat keras sebagai berikut:

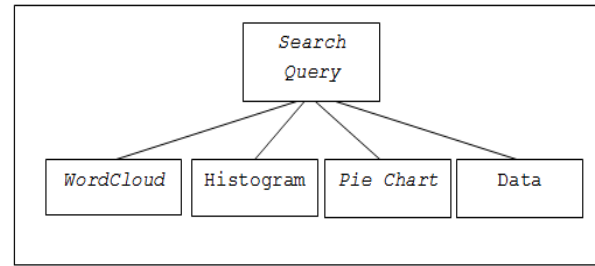
1. Processor Intel® Core™ i3-2330M CPU @ 2.20GHz 2.20 GHz
2. RAM 2.00 GB (1.84 GB yang dapat dipakai)
3. Harddisk 406 GB
4. Mouse Logitech M100r

3.1.3 Perancangan

Pada tahap ini dibuat rancangan sistem, diagram hirarki, *flowchart*, dan rancangan *interface*.

3.1.3.1 Diagram Hirarki

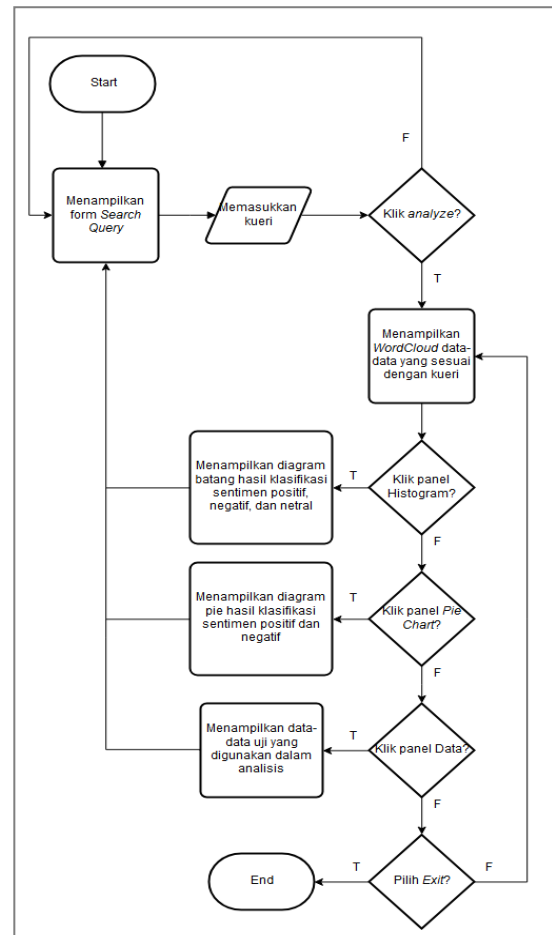
Diagram Hirarki berfungsi untuk memperlihatkan hirarki dari aplikasi yang akan dibuat. Dengan diagram hirarki, dapat dilihat struktur dan gambaran awal dari aplikasi. Diagram hirarki dapat dilihat pada Gambar 2.



Gambar 2 Diagram Hirarki

3.1.3.2 Flowchart

Flowchart berguna untuk menggambarkan proses-proses berdasarkan urutan langkah dari satu proses ke proses lainnya. *Flowchart* juga memdahkan pengguna untuk memahami alur dari aplikasi yang dibuat. *Flowchart* dapat dilihat pada Gambar 3.



Gambar 3 Flowchart Opinion Mining

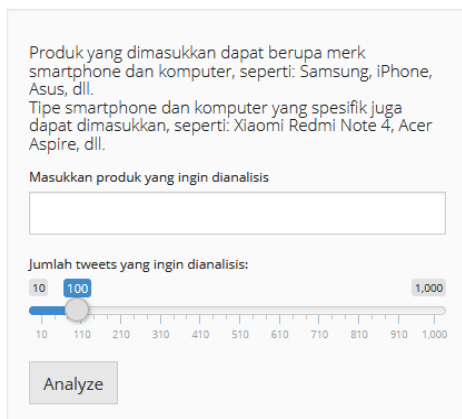
3.2 Pembuatan Interface dan Sistem

Interface berguna untuk memudahkan pengguna dalam mengoperasikan aplikasi sekalipun pengguna tidak mengerti metode-metode yang digunakan.

Pembuatan sistem diawali dengan melakukan *crawling* data dari *twitter* untuk dijadikan *corpus*. Lalu, masuk ke tahap mendesain *interface*. Setiap modul dibuat sesuai dengan rancangan *interface* dengan tampilan *Graphical User Interface* (GUI). Setelah itu, dilakukan *coding* untuk *pre-processing*, Klasifikasi *Naive Bayes*, evaluasi *Confusion Matrix*, tampilan *WordCloud*, *Histogram*, *Pie Chart*, dan *Data*. Tampilan *interface* yang dibuat adalah sebagai berikut:

1. Modul *Search Query*

Modul *Search Query* merupakan tampilan awal dari aplikasi. Pada modul ini terdapat *textfield* yang dapat diisi dengan produk yang ingin dianalisis. Lalu, terdapat *sidebar* yang berguna untuk menentukan jumlah *tweets* yang ingin dianalisis dan tombol *analyze* yang berguna untuk menganalisis *query* yang dimasukkan dalam *textfield*. Pada modul ini juga terdapat tombol *WordCloud*, *Histogram*, *Pie Chart*, *Data*, *Help*, dan *About*. Tampilan modul *Search Query* dapat dilihat pada Gambar 4.



Gambar 4 Tampilan Modul *Search Query*

2. Modul *WordCloud*

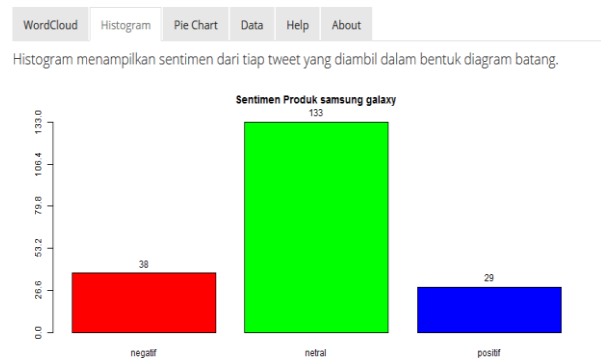
Modul ini berguna untuk menampilkan frekuensi kata-kata yang didapatkan dari hasil analisis. Semakin besar frekuensi suatu kata, ukuran kata tersebut akan ditampilkan semakin besar pada *WordCloud*. Tampilan modul *WordCloud* dapat dilihat pada Gambar 5.



Gambar 5 Tampilan Modul *WordCloud*

3. Modul *Histogram*

Modul ini berguna untuk menampilkan hasil analisis sentimen dalam bentuk diagram batang. Tampilan modul *Histogram* dapat dilihat pada Gambar 6.



Gambar 6 Tampilan Modul *Histogram*

4. Modul *Pie Chart*

Pada modul ini, pengguna dapat melihat pembagian sentimen positif, negatif, dan netral dalam bentuk diagram pie. Tampilan modul *Pie Chart* dapat dilihat pada Gambar 7.



Gambar 7 Tampilan Modul *Pie Chart*

5. Modul *Data*

Modul ini menunjukkan data-data uji yang dianalisis sesuai dengan kueri yang dimasukkan. Tampilan modul *Data* dapat dilihat pada Gambar 8.

	Sentimen	Tweet	Waktu
1	negatif	@Lilin_shop SamsungGalaxy A7 (2017)	2017-12-28 14:38:27
2	netral	@Lilin_shop Samsung galaxy a5	2017-12-28 14:35:45
3	negatif	Saya suka video @YouTube Unboxing Samsung Galaxy Tab A (Indonesia)	2017-12-28 14:35:13
4	netral	@Lilin_shop Samsung Galaxy A7	2017-12-28 14:34:49
5	netral	@Lilin_shop Samsung Galaxy A 2017	2017-12-28 14:33:11
6	negatif	@UnboxTherapy Samsung Galaxy grand neo	2017-12-28 14:31:46
7	negatif	Saya suka video @YouTube Review Samsung Galaxy TAB 3 8.0	2017-12-28 14:30:44
8	netral	samsung galaxy j5 pro garansi resmi : Saya baru beli produk keren di aplikasi Akulaku dengan minimum pembayaran ci	2017-12-28 14:30:41

Gambar 8 Tampilan Modul Data

6. Modul Help

Modul ini berguna untuk memberikan keterangan mengenai petunjuk cara penggunaan aplikasi. Rancangan modul Help dapat dilihat pada Gambar 9.

O;Pinia adalah program yang berguna untuk mengamati sentimen produk smartphone dan komputer dari twitter.

- Pada textfield yang tersedia di sisi kiri, masukkan merk smartphone atau komputer yang ingin diketahui sentimennya.
- Lalu, masukkan jumlah tweet yang ingin dicari pada slider.
- Setelah itu, tekan tombol 'analize' dan tunggu hasil ditampilkan pada tab wordcloud, histogram, piechart, dan data.

Gambar 9 Rancangan Modul Help

7. Modul About

Modul ini berguna untuk memberikan informasi mengenai pembuat aplikasi. Rancangan modul About dapat dilihat pada Gambar 10.

Nama: Albert Jeremy
 NIM: 535130024
 Pembimbing 1: Viny Christanti M., M.Kom
 Pembimbing 2: Bagus Mulyawan, MM

Gambar 10 Rancangan Modul About

4. Pengujian dan Pembahasan

4.1 Metode Pengujian

Terdapat dua hal yang diuji, yaitu modul-modul dan data output hasil klasifikasi. Pengujian yang dilakukan terhadap hasil modul-modul menggunakan *black box*

testing. *Black box testing* adalah pengujian perangkat lunak yang dilakukan dengan melihat *input* dan *output* tanpa memperhatikan struktur coding perangkat lunak tersebut. Tujuan dari pengujian ini adalah untuk memastikan bahwa tiap modul dapat berfungsi dengan baik ketika digunakan oleh pengguna. Pengujian yang dilakukan terhadap data *output* adalah akurasi dari klasifikasi sentimennya. Terdapat dua tipe pembagian data latih dan uji dalam pengujian ini. Tipe 1, yaitu menggunakan 255 data latih dan 63 data uji dengan mengikutsertakan data uji dalam data latih. Tipe 2, yaitu menggunakan 192 data latih dan 63 data uji dengan tidak mengikutsertakan data uji dalam data latih. Matriks dokumen dibentuk menjadi *unigram*, *bigram*, dan *trigram*. Akurasi dari masing-masing matriks dokumen dihitung menggunakan *Confusion Matrix*.

4.2 Hasil Pengujian dan Pembahasan

Pengujian program dari kedua metode pengujian, yaitu *Black Box Testing* untuk modul-modul program dan evaluasi *Confusion Matrix* untuk *output* data hasil klasifikasi dapat dilihat pada bagian berikut ini.

4.2.1 Evaluasi Confusion Matrix

Pengujian ini membentuk *Confusion Matrix* berupa tabel yang membandingkan kelas/sentimen yang sebenarnya dengan hasil klasifikasi *Naive Bayes* dan menghitung akurasinya.

4.2.2 Evaluasi Tipe Data 1

1. Unigram

Tabel 1 *Confusion Matrix* Unigram Tipe Data 1

Kelas Hasil Prediksi	Kelas Sebenarnya		
	negatif	netral	Positif
negatif	17	0	0
netral	0	27	0
positif	0	1	18

Dari 255 data latih dan 63 data uji, sistem berhasil mengklasifikasi 62 data dengan tepat dan 1 data dengan keliru. Akurasi yang didapat adalah $(62/63) \times 100\% = 98,4\%$. Pada Tabel 1, sistem keliru mengklasifikasikan kalimat berikut: “ya kalo kamera emang bagus oppo. tapi speknya kalah jauh ama xiaomi dia mah menang kamera doang wkwk”. Kalimat tersebut beropini netral, tetapi sistem mengklasifikasikannya sebagai

positif karena terdapat kata “bagus” pada kalimat tersebut.

2. Bigram

Tabel 2 *Confusion Matrix* Bigram Tipe Data 1

Kelas Hasil Prediksi	Kelas Sebenarnya		
	negatif	netral	positif
Negatif	17	0	0
Netral	0	28	0
Positif	0	0	18

Dari 255 data latih dan 63 data uji, sistem berhasil mengklasifikasi 63 data dengan tepat dan 0 data dengan keliru. Akurasi yang didapat adalah $(63/63) \cdot 100\% = 100\%$. Sistem berhasil mengklasifikasikan seluruh kalimat dengan kelas yang tepat.

3. Trigram

Tabel 3 *Confusion Matrix* Trigram Tipe Data 1

Kelas Hasil Prediksi	Kelas Sebenarnya		
	negatif	netral	positif
negatif	17	0	0
netral	0	28	0
positif	0	0	18

Dari 255 data latih dan 63 data uji, sistem berhasil mengklasifikasi 63 data dengan tepat dan 0 data dengan keliru. Akurasi yang didapat adalah $(63/63) \cdot 100\% = 100\%$. Sistem berhasil mengklasifikasikan seluruh kalimat dengan kelas yang tepat.

4.3.2 Evaluasi Tipe Data 2

1. Unigram

Tabel 4 *Confusion Matrix* Unigram Tipe Data 2

Kelas Hasil Prediksi	Kelas Sebenarnya		
	negatif	Netral	positif
Negatif	16	4	0
Netral	4	14	2
Positif	1	3	19

Dari 192 data latih dan 63 data uji, sistem berhasil mengklasifikasi 49 data dengan tepat dan 14 data dengan keliru. Akurasi yang didapat adalah $(49/63) \cdot 100\% = 77,7\%$. Pada Tabel 4, sistem keliru mengklasifikasikan kalimat berikut: “Sony Xperia XZ Premium Smartphone ini benar-benar punya kamera yang patut diacungi jempol”. Kalimat tersebut beropini positif, tetapi sistem mengklasifikasikannya sebagai netral karena tidak mengandung kata positif yang terdapat pada data latih.

2. Bigram

Tabel 5 *Confusion Matrix* Bigram Tipe Data 2

Kelas Hasil Prediksi	Kelas Sebenarnya		
	negatif	netral	positif
Negatif	11	4	3
Netral	8	13	10
Positif	2	4	8

Dari 192 data latih dan 63 data uji, sistem berhasil mengklasifikasi 21 data dengan tepat dan 42 data dengan keliru. Akurasi yang didapat adalah $(32/63) \cdot 100\% = 50,7\%$. Pada Tabel 5, sistem keliru mengklasifikasikan kalimat berikut: “Aku enakan oppo ramnya lebih gede, ga lemot, bateranya cepet penuh + awet, terus kameranya lebih bagus juga ya walaupun aku ga peduli soal”. Kalimat tersebut beropini positif, tetapi sistem mengklasifikasikannya sebagai negatif.

3. Trigram

Tabel 6 *Confusion Matrix* Trigram Tipe Data 2

Kelas Hasil Prediksi	Kelas Sebenarnya		
	negatif	Netral	positif
negatif	2	1	1
netral	19	16	18
positif	0	4	2

Dari 192 data latih dan 63 data uji, sistem berhasil mengklasifikasi 20 data dengan tepat dan 43 data dengan keliru. Akurasi yang didapat adalah $(20/63) \cdot 100\% = 31,7\%$. Pada Tabel 4, sistem keliru mengklasifikasikan kalimat berikut: “Selama ini aku pake seri asus zenfone max sih. Dari 1 upgrade ke 2, rencana upgrade ke 3, soalnya kalo buat streaming-sosmed doang bateranya kuat2 aja. Cuman kameranya b aja sih ga jelek ga istimewa juga, sama kalo dipake gaming agak ga kuat.”. Kalimat tersebut beropini netral, tetapi sistem mengklasifikasikannya sebagai negatif karena terdapat kata “jelek” pada kalimat tersebut.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Berdasarkan hasil pengujian yang dilakukan terhadap hasil prediksi dengan matriks dokumen unigram, bigram, dan trigram, dan modul-modul yang ada, diperoleh kesimpulan sebagai berikut:

1. Program dapat mengambil dan menganalisis data sesuai dengan produk yang dicari. Produk-produknya adalah *smartphone* Samsung, iPhone,

Oppo, Lenovo, Huawei, Sony, dan komputer Acer dan Asus.

2. Hasil prediksi memberikan akurasi terbaik pada matriks dokumen unigram, yaitu sebesar 77,7%.
3. Pengguna dapat melihat hasil klasifikasi sentimen yang ditampilkan dari *tweets* yang ada.

5.2 Saran

Saran penulis untuk perancangan berikutnya sebagai berikut:

1. Menggunakan *corpus* dengan jumlah data yang lebih besar.
2. Menambah proses *pre-processing* pada data latih, seperti *stemming*, *thesaurus*, dan mengubah tiap kata pada *corpus* dengan kata baku.

Viny Christanti M., M.Kom, memperoleh gelar S.Kom dari Universitas Tarumanagara, Jakarta tahun 2004 dan gelar M.Kom dari Universitas Indonesia, Depok tahun 2008. Saat ini sebagai Staf Pengajar program studi teknik Informatika Fakultas Teknologi Informasi Universitas Tarumanagara.

Bagus Mulyawan, MM, Memperoleh gelar S.Kom dari Universitas Gunadarma tahun 1992 dan gelar MM dari Universitas Budi Luhur tahun 2008. Saat ini sebagai Staf Pengajar program studi teknik Informatika Fakultas Teknologi Informasi Universitas Tarumanagara.

REFERENSI

- [1] Zephoria, The Twitter Top Ten, <https://zephoria.com/twitter-statistics-top-ten/>, diakses pada tanggal 7 September 2017.
- [2] Nuke Y. A. Faradhillah et. al., Eksperimen Sistem Klasifikasi Analisa Sentimen Twitter pada Akun Resmi Pemerintah Kota Surabaya Berbasis Pembelajaran Mesin, is.its.ac.id/pubs/oajis/index.php/file/download_file/1645, diakses pada tanggal 7 September 2017.
- [3] Bing Liu, *Sentiment Analysis and Opinion Mining*, (Chicago: Morgan & Claypool Publishers, 2012), h. 1.
- [4] Aditya, *Text Processing*, <http://informatikalogi.com/text-preprocessing/>, diakses pada tanggal 10 September 2017.
- [5] Christopher Fox, Chapter 7: Lexical Analysis and Stoplists, <http://orion.lcg.ufrj.br/Dr.Dobbs/books/book5/chap07.htm>, diakses pada tanggal 10 September 2017.
- [6] Cristian Moral et al., A survey of stemming algorithms in information retrieval, <http://www.informationr.net/ir/19-1/paper605.html#WL7qg7nttX0>, diakses pada tanggal 10 September 2017.
- [7] Charu C. Aggarwal, *Data Classification Algorithms and Applications*, (New York: CRC Press, 2015), h. 438.
- [8] Christopher D. Manning, *Foundations of Statistical Natural Language Processing*, (Massachusetts: The MIT Press, 2003), h. 237.
- [9] Kun Ren, *Learning R Programming*, (Birmingham: Packt Publishing, 2016), h.7.
- [10] Bingwei Liu et. al., "Scalable sentiment classification for Big Data analysis using Naive Bayes Classifier", *IEEE 2013 International Conference on Big Data*, (Oktober, 2013), h. 99.
- [11] Ting, S. L., W. H. Ip, dan Albert HC Tsang, "Is Naive Bayes a good classifier for document classification", *International Journal of Software Engineering and Its Applications*, Vol. 5, Nomor 3, (Juli, 2011), h. 44.

Albert Jeremy, mahasiswa S1, program studi Teknik Informatika, Fakultas Teknologi Informasi Universitas Tarumanagara.