

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/320234777>

Analisis Sentimen Tentang Opini Film pada Dokumen Twitter Berbahasa Indonesia Menggunakan Naive Bayes dengan Perbaikan Kata Tidak Baku

Article · December 2017

CITATIONS

9

READS

4,695

3 authors, including:



Rizal Setya Perdana
Brawijaya University

41 PUBLICATIONS 119 CITATIONS

[SEE PROFILE](#)



Muhammad Ali Fauzi
Brawijaya University

112 PUBLICATIONS 334 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Smart Mobile Navigation System [View project](#)



Optimasi Sisa Bahan Baku Pada Industri Mebel Menggunakan Algoritma Genetika [View project](#)

Analisis Sentimen Tentang Opini Film Pada Dokumen Twitter Berbahasa Indonesia Menggunakan Naive Bayes Dengan Perbaikan Kata Tidak Baku

Prananda Antinasari¹, Rizal Setya Perdana², M. Ali Fauzi³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹nandaopra@gmail.com, ²rizalespe@ub.ac.id, ³moch.ali.fauzi@ub.ac.id

Abstrak

Pertumbuhan media sosial yang sangat pesat tidak membuat Twitter ditinggalkan oleh penggunanya. Twitter merupakan salah satu media sosial yang memungkinkan penggunanya untuk melakukan interaksi, berbagi informasi, atau bahkan untuk mengutarakan perasaan dan opini, termasuk juga dalam mengutarakan opini film. Komentar atau Tweet mengenai film yang ada pada Twitter dapat dijadikan sebagai evaluasi dalam menonton film dan meningkatkan produksi film. Untuk mengetahui hal tersebut, analisis sentimen dapat digunakan untuk mengklasifikasikan kedalam sentimen negatif atau positif. Didalam Tweet terkandung banyak ragam bahasa yang digunakan, yaitu diantaranya bahasa dalam bentuk tidak baku seperti bahasa *slang*, penyingkatan kata, dan salah eja. Oleh sebab itu dibutuhkan penanganan khusus pada Tweet. Pada penelitian ini digunakan kamus kata tidak baku dan normalisasi Levenshtein Distance untuk memperbaiki kata yang tidak baku menjadi kata baku dengan pengklasifikasian Naive Bayes. Berdasarkan hasil pengujian yang telah dilakukan didapatkan akurasi tertinggi dengan nilai *accuracy*, *precision*, *recall*, dan *f-measure* sebesar 98.33%, 96.77%, 100%, dan 98.36%.

Kata kunci: *twitter, tweet, analisis sentimen, perbaikan kata tidak baku, normalisasi levenshtein distance, naive bayes*

Abstract

The rapid growth of social media does not make Twitter left by its users. Twitter is one of the social media that allows user to interact each other, share information, or even to express feelings and opinions, including in expressing film opinions. Comments or Tweets about movies that exist on Twitter can be used as an evaluation in watching movies and increasing film production. To figure it out, sentiment analysis can be used to classify into negative or positive sentiments. In Tweets contain many languages used in the form of non-standard languages such as *slang*, word-outs, and misspellings. Therefore it takes special handling on Twitter comments. In this research used non-standard word dictionary and Levenshtein Distance normalization to improve non-standard word to standard word by classification Naive Bayes. Based on the result of the test, the highest *accuracy*, *precision*, *recall*, and *f-measure* value are 98.33%, 96.77%, 100%, and 98.36%.

Keywords: *twitter, tweet, sentiment analysis, non-standard word improvement, levenshtein distance normalization, naive bayes*

1. PENDAHULUAN

Twitter merupakan salah satu media sosial yang memungkinkan penggunanya untuk berbagi informasi dengan sesama secara *realtime*. Informasi yang dibagikan pada Twitter biasanya disebut dengan kicauan (Tweet) yang terdiri dari 140 karakter (Kwak, et al., 2010). Berdasarkan penelitian Semiocast, lembaga riset media sosial yang berpusat di Paris, Prancis, mengatakan bahwa jumlah pemilik akun Twitter di Indonesia merupakan yang terbesar kelima di

dunia, dan berada pada posisi ketiga negara yang paling aktif mengirim Tweet perhari (Asih, 2012). Tingginya pengguna Twitter menjadi peluang untuk masyarakat dalam melakukan jual beli, menyampaikan informasi, promosi, atau bahkan untuk mengutarakan perasaan dan opini, termasuk juga dalam mengutarakan opini film. Opini tentang film dapat digunakan sebagai sarana untuk memberikan penilaian terhadap film yang diproduksi untuk masyarakat. Hal tersebut dapat dimanfaatkan oleh produser film

untuk mengetahui bagaimana tanggapan masyarakat mengenai film yang telah diproduksi, begitu juga dengan masyarakat dapat meninjau film yang akan ditonton. Disinilah analisis sentimen memiliki peran yang sangat penting. Analisis sentimen merupakan daerah penelitian perhitungan untuk mengekstraksi polaritas pendapat antar kelas (positif dan negatif) dari dokumen teks (Perdana & Pinandito, 2017).

Penggunaan analisis sentimen dapat diterapkan pada opini film pada dokumen Twitter berbahasa Indonesia. Pada penulisan opini film terkadang terdapat penulisan Tweet yang sulit dibaca. Hal ini disebabkan oleh beberapa faktor seperti penulisan kata yang disingkat, penggunaan bahasa modern atau *slang*, salah dalam mengetik huruf dan tidak baku dalam penulisan opini (Agarwal, et al., 2014). Beberapa hal tersebut menyebabkan perlu dilakukan representasi ulang kata sebelum dilakukan penentuan sentimen opini. Kata tersebut nantinya akan diklasifikasikan agar dapat diketahui apakah akan masuk ke dalam sentimen negatif atau positif. Dalam hal ini maka dibutuhkan metode klasifikasi untuk menganalisis komentar atau opini.

Pada penelitian sebelumnya yang berkaitan dengan analisis sentimen sudah pernah dilakukan oleh Firmansyah (2016). Pada penelitian ini digunakan metode Naive Bayes untuk mengklasifikasikan sentimen dari dokumen *review* aplikasi *mobile*. Selain Naive Bayes pada penelitian ini juga digunakan metode Query Expansion untuk mengoptimasi komentar singkat. Hasil pengujian pada penelitian ini menunjukkan bahwa dengan menggunakan algoritme Naive Bayes didapatkan hasil akurasi yang cukup baik yaitu sebesar 95%. Dan dengan penambahan penggunaan metode Query Expansion menambah nilai akurasi dari Naive Bayes yaitu menjadi 98% (Firmansyah, 2016). Namun pada penelitian ini masih memiliki kekurangan yaitu masih belum dilakukannya proses normalisasi bahasa atau perbaikan kata tidak baku, sehingga menyebabkan menurunnya tingkat akurasi.

Penelitian selanjutnya yang berkaitan dengan analisis sentimen yaitu penelitian yang dilakukan oleh Nugraha (2014). Pada penelitian ini dilakukan pengklasifikasian terhadap sentiment positif dan negatif mengenai *review*

film berbahasa Inggris dengan menggunakan metode K-Nearest Neighbor (K-NN). Pada penelitian ini membandingkan metode K-NN dengan metode Naive Bayes. Hasil dari akurasi yang didapat menunjukkan bahwa metode Naive Bayes lebih baik dibandingkan dengan metode K-NN dimana metode Naive Bayes memiliki nilai akurasi tertinggi sebesar 81% dibandingkan dengan menggunakan metode K-NN yang bernilai akurasi sebesar 71%. Dan pada penelitian ini juga masih belum dilakukannya perbaikan kata tidak baku, sehingga tingkat akurasi tertinggi yang dicapai untuk metode Naive Bayes hanya sebesar 81%.

Berdasarkan uraian di atas, peneliti mengusulkan penelitian dengan menggunakan Naive Bayes untuk klasifikasi opini film dan untuk menangani masalah kata yang tidak baku, akan dilakukan suatu proses perbaikan kata tidak baku dengan menggunakan kamus kata tidak baku dan dengan dilakukan proses normalisasi Levenshtein Distance.

2. TINJAUAN PUSTAKA

2.1 Pre-processing

Pada *text mining* diperlukan beberapa tahapan untuk mengolah teks menjadi lebih terstruktur. Salah satu tahapan pada *text mining* adalah *pre-processing*. Tahap ini adalah tahapan yang mana data disiapkan agar menjadi data yang siap untuk dianalisis (Hadna, et al., 2016). Tahapan *pre-processing* pada penelitian ini meliputi proses *tokenizing*, *cleansing*, *case folding*, perbaikan kata tidak baku, *filtering*, dan *stemming*.

2.2 Perbaikan kata tidak baku

Perbaikan kata tidak baku atau normalisasi bahasa adalah proses yang digunakan untuk mengubah kata-kata yang tidak baku menjadi kata baku sesuai dengan Kamus Besar Bahasa Indonesia (KBBI) (Buntoro, et al., 2014).

Pada perbaikan kata tidak baku terdapat bermacam-macam kata yang dianggap kata tidak baku, seperti penulisan dengan bahasa modern atau *slang* (ex : gue = saya), penulisan dengan kaidah atau pedoman bahasa (ex : kualitas = kualitas), penulisan dengan kata singkat (ex : adlh = adalah), penulisan dengan angka (ex : n41k = naik), dan penulisan dengan kata salah eja (ex : jelsk = jelek).

Pada penelitian ini digunakan perbaikan kata tidak baku untuk menangani permasalahan kata tidak baku hanya pada bahasa modern atau *slang*, penyingkatan kata, dan salah eja saja. Perbaikan kata tidak baku dilakukan dengan proses normalisasi yang dilakukan dengan melakukan pengecekan *token-token* pada kalimat dengan menggunakan kamus_katabaku, jika terdapat kata yang tidak baku maka akan diubah menjadi kata yang baku sesuai dengan kamus. Sehingga didapatkan hasil akhir berupa kalimat dengan menggunakan kata baku. Tabel contoh kamus_katabaku dapat dilihat pada Tabel 1.

Tabel 1. Contoh kamus_katabaku

Kata Tidak Baku	Kata Baku
abis	habis
cew	cewek
gokil	gila
kece	keren
kshn	kasihan
gue	saya
nyesel	sesal

2.2.1 Normalisasi Levenshtein Distance

Levenshtein Distance atau yang biasa disebut dengan *edit distance* adalah suatu metode yang dapat digunakan untuk mengatasi terjadinya kesalahan ejaan. Kesalahan ejaan terjadi apabila kata yang diketik oleh pengguna tidak terdapat pada daftar kamus Bahasa Indonesia. Fungsi metode Levenshtein Distance yaitu untuk menghitung jarak kedekatan dari dua buah *string* melalui penambahan karakter, pengubahan karakter, dan penghapusan karakter hingga kedua string tersebut cocok (Freman, et al., 2006).

Pada algoritme Levenshtein distance terdapat 3 macam operasi utama yang dilakukan yaitu (Adriyani, et al., 2012):

1. Operasi Penambahan Karakter
Operasi penambahan karakter yaitu operasi yang digunakan untuk menambahkan karakter ke dalam *string*. Contoh pada penulisan *string* 'kern' maka diubah menjadi *string* 'keren' dengan menambahkan karakter 'e'.

2. Operasi Pengubahan Karakter

Operasi pengubahan karakter yaitu operasi yang digunakan untuk mengubah karakter dengan cara menukar sebuah karakter dengan karakter lain. Contoh pada penulisan *string* 'hidsp' diubah menjadi *string* 'hidup' dengan mengubah karakter 's' menjadi karakter 'u'.

3. Operasi Penghapusan Karakter

Operasi penghapusan karakter yaitu operasi yang digunakan untuk menghapus suatu karakter pada *string*. Contoh pada penulisan *string* 'hebatt' diubah menjadi *string* 'hebat' dengan menghilangkan karakter 't'.

Pada algoritme ini tahapan proses dimulai dari pojok kiri atas dari sebuah array dua dimensi yang telah diisi karakter *string* input dan *string* target. Selain itu juga terdapat nilai *cost* didalamnya. Nilai *cost* yang berada pada pojok kanan bawah merupakan nilai *edit distance* yang menggambarkan jumlah perbedaan kedua *string*. Contoh perhitungan *edit distance* dapat dilihat pada Gambar 1.

		F	I	L	M
	0	1	2	3	4
F	1	0	1	2	3
L	2	1	1	1	2
M	3	2	2	2	1

Gambar 1. Matriks Perhitungan Edit Distance

Gambar 1 menunjukkan perhitungan *edit distance* menggunakan 2 *string* yang berbeda. Nilai *edit distance* yang dihasilkan adalah bernilai 1 ditandai dengan arsiran pada pojok kanan bawah. Pengecekan dimulai dari iterasi awal dari kedua *string* kemudian dilakukan operasi pengubahan, penambahan dan penghapusan karakter. Pada proses ini hanya terdapat 1 proses yaitu penyisipan karakter 'I' pada *string* 'FLM'. Sehingga menjadi *string* 'FILM'.

2.3 Naive Bayes Classifier

Naive Bayes Classifier merupakan salah satu metode yang populer untuk keperluan *data mining* karena penggunaannya yang mudah (Hall, 2006) dan dalam pemrosesan memiliki waktu yang cepat, mudah diimplementasikan dengan strukturnya yang cukup sederhana dan

untuk tingkat efektivitasnya memiliki efektivitas yang tinggi (Taheri & Mammadov, 2013). Klasifikasi Naive Bayes juga memperlihatkan tingginya akurasi dan cepat ketika digunakan untuk dataset dengan jumlah besar (Aggarwal, 2015).

Secara umum proses dari klasifikasi Naive Bayes dapat dilihat pada Persamaan 1.

$$P(c_j|w_i) = \frac{P(c_j) P(w_i|c_j)}{P(w_i)} \quad (1)$$

Keterangan:

$P(c_j|w_i)$: Peluang kategori j ketika terdapat kemunculan kata i

$P(w_i|c_j)$: Peluang sebuah kata i masuk ke dalam kategori j

$P(c_j)$: Peluang kemunculan sebuah kategori j

$P(w_i)$: Peluang kemunculan sebuah kata

Pada proses perhitungan klasifikasi peluang kemunculan kata sebenarnya dapat dihilangkan, hal ini dikarenakan peluang tersebut tidak berpengaruh pada perbandingan hasil klasifikasi dari setiap kategori. Sehingga proses pada klasifikasi dapat disederhanakan dengan Persamaan 2.

$$P(c_j|w_i) = P(c_j) P(w_i|c_j) \quad (2)$$

Pada perhitungan klasifikasi terdapat *prior* yang digunakan untuk menghitung peluang kemunculan kategori pada semua dokumen, perhitungan prior dapat dilihat pada Persamaan 3.

$$P(c_j) = \frac{N_c}{N} \quad (3)$$

Keterangan:

N_c : Banyak dokumen berkategori c_j pada dokumen latih

N : Jumlah keseluruhan dokumen latih yang digunakan.

Perhitungan *posterior* merupakan perhitungan yang dilakukan dengan mengalikan *prior* dengan total *conditional probability*. Rumus perhitungan dapat dilihat pada Persamaan 4.

$$P(c_j|w_i) = P(c_j) \times P(w_1|c_j) \times \dots \times P(w_n|c_j) \quad (4)$$

2.3.1 Multinomial Naive Bayes

Multinomial Naive Bayes adalah *conditional probability* yang dilakukan tanpa memperhitungkan urutan kata dan informasi yang ada dalam kalimat atau dokumen secara umum. Dan pada algoritme ini juga memperhitungkan jumlah kata yang muncul dalam dokumen (Destuardi dan Surya, 2009). Algoritme *conditional probability* Multinomial Naive Bayes dalam melakukan perhitungan peluang sebuah kata i masuk ke dalam kategori j dapat dilakukan dengan menggunakan Persamaan 5.

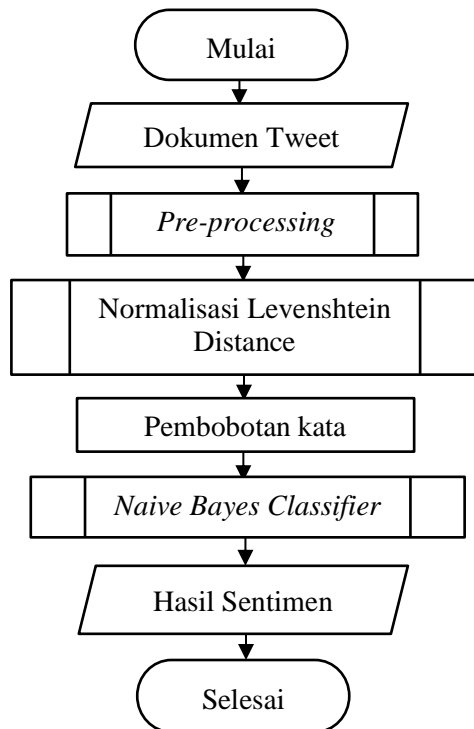
$$P(w_i|c_j) = \frac{\text{count}(w_i, c_j) + 1}{(\sum_{w \in V} \text{count}(w, c_j)) + |V|} \quad (5)$$

Keterangan :

Bahwa $\text{count}(w_i, c_j)$ merupakan jumlah dari suatu kata *query* yang muncul dalam suatu kelas atau kategori. Dan penambahan angka 1 tersebut digunakan untuk menghindari nilai nol. Selanjutnya $\sum_{w \in V} \text{count}(w, c_j)$ merupakan jumlah dari seluruh kata yang ada pada kelas atau kategori c_j . Dan $|V|$ merupakan jumlah dari seluruh kata unik yang ada pada seluruh kategori.

3. METODOLOGI

Pada analisis sentimen tentang opini film pada dokumen Twitter berbahasa Indonesia dilakukan proses *pre-processing* terlebih dahulu untuk selanjutnya dilakukan perbaikan kata tidak baku dan proses klasifikasi menggunakan Naive Bayes. Diagram alir sistem secara umum digambarkan pada Gambar 2.



Gambar 2. Diagram alir sistem

Pada Gambar 2 dijelaskan sistem yang akan dikembangkan dalam penelitian ini yaitu bertujuan untuk menyelesaikan permasalahan analisis sentimen tentang opini film pada dokumen Twitter berbahasa Indonesia menggunakan Naive Bayes dengan perbaikan kata tidak baku. Dokumen yang akan diklasifikasi diawali dengan proses *pre-processing*, dimana dalam proses *pre-processing* terdapat proses tambahan yaitu perbaikan kata tidak baku yang dilakukan setelah tahap *case folding*. Setelah dilakukan *pre-processing* maka langkah selanjutnya akan dilanjutkan dengan normalisasi Levenshtein Distance yaitu untuk memperbaiki kata yang masih tidak baku. Kemudian setelah normalisasi Levenshtein Distance selesai dilakukan maka dilakukan pengklasifikasian menggunakan Naive Bayes untuk menentukan kelas sentimen. Hasil klasifikasi pada Naive Bayes tergantung pada probabilitas frekuensi kemunculan kata pada data latih.

4. HASIL DAN PEMBAHASAN

Pada pengujian ini terdapat 3 macam pengujian yang dilakukan. Pengujian pertama dilakukan dengan tujuan untuk mengetahui pengaruh proses penggunaan *pre-processing*.

Kemudian untuk pengujian kedua dilakukan untuk mengetahui pengaruh penggunaan proses perbaikan kata tidak baku. Dan terakhir adalah pengujian untuk mengetahui pengaruh dari dilakukannya proses *pre-processing* dengan perbaikan kata tidak baku dan normalisasi Levenshtein Distance.

Pada pengujian ini data yang digunakan merupakan data asli dari pengguna Twitter mengenai Tweet opini film berbahasa Indonesia. Data latih yang diambil sebanyak 140 data opini, yang terdiri dari 70 data opini positif dan 70 data opini negatif. Sedangkan untuk data uji digunakan 60 data, yang terdiri dari 30 data opini positif dan 30 data opini negatif.

3.1 Pengujian pengaruh penggunaan *pre-processing*

Pengujian ini menjelaskan tentang pengujian untuk mengetahui pengaruh proses *pre-processing*. Pada pengujian ini dilakukan tiga tahapan. Yaitu Tahap 1 pengujian tanpa *pre-processing* dan tanpa perbaikan kata, Tahap 2 pengujian tanpa *pre-processing* dengan perbaikan kata, dan Tahap 3 pengujian dengan *pre-processing* dan tanpa perbaikan kata. Hasil pengujian pengaruh penggunaan *pre-processing* dapat dilihat pada Tabel 2.

Tabel 2. Hasil pengujian pengaruh penggunaan *pre-processing*

Uji	<i>accuracy</i>	<i>precision</i>	<i>recall</i>	<i>f-m</i>
Tahap 1	70%	80%	53.33%	64%
Tahap 2	75%	71.43%	83.33%	76.92%
Tahap 3	86.67%	92.31%	80%	85.72%

Berdasarkan hasil pengujian pada pengujian pengaruh penggunaan *pre-processing*, diperoleh hasil dari Tahap 1 dengan nilai *accuracy*, *precision*, *recall*, dan *f-measure* sebesar 70%, 80%, 53.33%, dan 64%. Hasil tersebut dikatakan rendah karena proses *pre-processing* tidak digunakan pada data uji sehingga kata yang digunakan untuk pengklasifikasian masih memiliki kata-kata asli seperti murah, termurah, murahan. Padahal kata tersebut memiliki arti kata yang sama yaitu murah. Akibat tidak dilakukannya proses *pre-processing* membuat semakin banyaknya kata yang tidak berguna yaitu kata yang ada dalam

data uji tidak banyak muncul pada data latih. Sehingga membuat proses klasifikasi menjadi tidak sempurna dan menghasilkan nilai akurasi yang rendah.

Selanjutnya pada Tahap 2 sama seperti Tahap 1 yaitu data uji tanpa dilakukan *pre-processing* namun pada tahap ini dilakukan perbaikan kata tidak baku dengan kamus_katabaku. Dihasilkan *accuracy*, *precision*, *recall*, dan *f-measure* sebesar 75%, 71.43%, 83.33%, dan 76.92%. Hasil ini menunjukkan hasil yang lebih baik dibandingkan dengan tanpa *pre-processing* dan tanpa perbaikan. Karena kata yang ada pada data uji dilakukan perbaikan kata dengan kamus_katabaku. Yaitu misal pada kata “bgt” diperbaiki menjadi “banget”. Sehingga kata yang ada dalam data uji akan lebih banyak muncul pada data latih dibandingkan Tahap 1. Namun masih memiliki kekurangan karena kata tidak penting atau tidak memiliki arti masih tercantum akibat tidak dilakukannya *filtering*. Dan juga tanpa dilakukannya proses mengubah kata berimbuhan menjadi kata dasar (*stemming*).

Selanjutnya Tahap 3 adalah tahap dilakukannya proses *pre-processing* dan tanpa perbaikan. Hasil *accuracy*, *precision*, *recall*, dan *f-measure* yang didapat adalah sebesar 86.67%, 92.31%, 80%, dan 85.71%. Hasil ini jauh lebih baik dibandingkan dengan tahap sebelumnya yang tanpa dilakukan *pre-processing*. Dari hasil tersebut menunjukkan bahwa dengan *pre-processing*, kata yang tidak memiliki arti akan dihilangkan, dan juga kata berimbuhan yang memiliki kata dasar yang sama akan dianggap sebagai suatu kata yang sama. Sehingga hasil dari pengujian ini dapat dibuktikan bahwa *pre-processing* berpengaruh besar pada hasil klasifikasi.

3.2 Pengujian pengaruh penggunaan perbaikan kata tidak baku

Pengujian ini menjelaskan tentang pengujian untuk mengetahui pengaruh proses perbaikan kata tidak baku menggunakan kamus kata tidak baku. Data latih dan data uji sama-sama dilakukan proses *pre-processing* namun dilakukan variasi dalam proses perbaikan kata tidak baku. Pada pengujian ini dilakukan dua tahapan. Yaitu Tahap 1 pengujian dengan *pre-processing* dan tanpa perbaikan kata, dan Tahap

2 pengujian dengan *pre-processing* dan dengan perbaikan kata. Hasil pengujian pengaruh penggunaan perbaikan kata tidak baku dapat dilihat pada Tabel 3.

Tabel 3. Hasil pengujian pengaruh penggunaan perbaikan kata tidak baku

Uji	<i>accuracy</i>	<i>precision</i>	<i>recall</i>	<i>f-m</i>
Tahap 1	86.67%	92.31%	80%	85.72%
Tahap 2	91.67%	85.71%	100%	92.31%

Berdasarkan pengujian penggunaan perbaikan kata tidak baku diperoleh hasil dari Tahap 1 dengan hasil *accuracy*, *precision*, *recall*, dan *f-measure* sebesar 86.67%, 92.31%, 80%, dan 85.72%. Dan hasil pada Tahap 2 sebesar 91.67%, 85.71%, 100%, dan 92.31%. Dari perbandingan nilai tersebut dapat dikatakan bahwa klasifikasi yang dilakukan proses *pre-processing* dan perbaikan kata tidak baku mendapatkan hasil yang jauh lebih baik. Karena data uji dijadikan kata yang lebih terstruktur dahulu sebelum dilakukan proses klasifikasi.

3.3 Pengujian pengaruh penggunaan perbaikan kata tidak baku dan normalisasi Levenshtein Distance

Pengujian ini menjelaskan tentang pengujian untuk mengetahui pengaruh proses perbaikan kata tidak baku menggunakan kamus tidak baku dan normalisasi Levenshtein Distance. Data latih dan data uji dilakukan variasi proses *pre-processing*. Dan dilakukan proses perbaikan kata tidak baku dan normalisasi Levenshtein Distance. Pada pengujian ini dilakukan dua tahapan. Yaitu Tahap 1 pengujian tanpa *pre-processing* dengan perbaikan kata dan Levenshtein Distance, dan Tahap 2 pengujian *pre-processing* dengan perbaikan kata dan Levenshtein Distance. Hasil pengujian pengaruh penggunaan perbaikan kata tidak baku dan normalisasi Levenshtein Distance dapat dilihat pada Tabel 4.

Tabel 4. Hasil pengujian pengaruh penggunaan perbaikan kata tidak baku dan normalisasi Levenshtein Distance

Uji	<i>accuracy</i>	<i>precision</i>	<i>recall</i>	<i>f-m</i>
Tahap 1	85%	86.21%	83.33%	84.75%
Tahap 2	98.33%	96.77%	100%	98.36%

Berdasarkan pengujian pada pengaruh penggunaan perbaikan kata tidak baku dan normalisasi Levenshtein Distance, diperoleh hasil pada Tahap 1 dengan *accuracy*, *precision*, *recall*, dan *f-measure* masing-masing dengan nilai 85%, 86.21%, 83.33%, dan 84.75%. Hasil ini menghasilkan akurasi yang baik walaupun tidak dilakukan proses *pre-processing*, karena pada saat perbaikan dengan menggunakan normalisasi Levenshtein Distance setiap kata akan dicocokkan dengan kamus *wordlist*. Sehingga kata yang tidak terstruktur tergantikan oleh kata yang ada pada *wordlist* yang didapat dari perhitungan *editdistance* dengan mengambil nilai *edit* terpendek untuk mengganti kata.

Kemudian Tahap 2 adalah dengan dilakukannya proses *pre-processing* dengan perbaikan kata dan Levenshtein Distance. Hasil dari pengujian ini didapatkan nilai *accuracy*, *precision*, *recall*, dan *f-measure* masing-masing dengan nilai 98.33%, 96.77%, 100%, dan 98.36%. Hasil ini menghasilkan akurasi terbaik dari seluruh pengujian diatas. Karena pada pengujian ini dilakukan seluruh proses mulai dari proses *pre-processing* yang menjadikan kata tidak terstruktur menjadi terstruktur dan ditambah dengan perbaikan kata tidak baku berdasarkan kamus *katabaku* yang mengubah kata singkatan atau kata *slang* menjadi kata baku. Selain itu juga dilakukan proses perbaikan kata dengan Levenshtein Distance. Yaitu memperbaiki kesalahan kata seperti salah eja yang tidak bisa diperbaiki pada kamus *katabaku*. Sehingga kata yang dihasilkan data uji memiliki banyak kemunculan kata sama dengan data latih yang berpengaruh pada proses klasifikasi. Dengan banyaknya kata yang muncul pada data latih maka menjadikan nilai akurasi menjadi lebih baik.

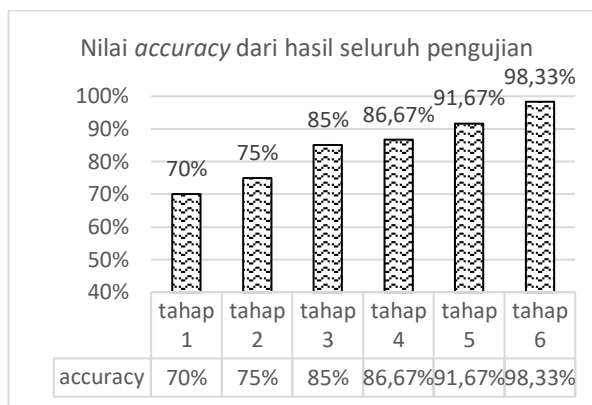
3.4 Hasil dari seluruh pengujian

Pada pengujian ini dilakukan untuk mengetahui hasil dari seluruh proses pengujian yang telah dilakukan. Yaitu dengan menampilkan grafik dari seluruh hasil. Hasil dari pengujian ini terdiri dari nilai *accuracy*, *precision*, *recall*, dan *f-measure*.

Pada pengujian ini ditampilkan 6 Tahapan dari seluruh proses pengujian yang telah dilakukan. Berikut urutan tahapan dari seluruh pengujian:

- Tahap 1 :
Tanpa *Pre-processing* dan tanpa perbaikan kata tidak baku
- Tahap 2 :
Tanpa *Pre-processing* dengan perbaikan kata tidak baku
- Tahap 3 :
Tanpa *Pre-processing* dengan perbaikan kata tidak baku dan Levenshtein Distance
- Tahap 4 :
Pre-processing tanpa perbaikan kata tidak baku
- Tahap 5 :
Pre-processing dengan perbaikan kata tidak baku
- Tahap 6 :
Pre-processing dengan perbaikan kata tidak baku dan Levenshtein Distance

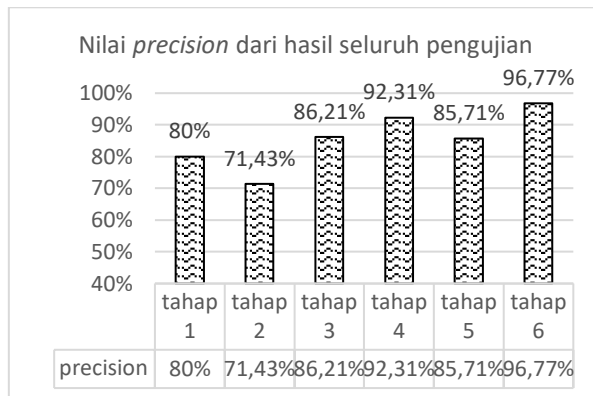
1. Hasil *accuracy*



Gambar 3. Nilai *accuracy* dari seluruh pengujian

Pada Gambar 3 menunjukkan hasil dari nilai *accuracy* yaitu nilai tingkat kedekatan antara nilai prediksi dengan nilai aktual. Pada grafik tersebut dapat dilihat bahwa pengujian *pre-processing* dengan perbaikan kata tidak baku dan Levenshtein Distance menghasilkan akurasi tertinggi. Karena pada pengujian tersebut dilakukan berbagai tahapan mulai dari *pre-processing*, perbaikan kata tidak baku menggunakan kamus *katabaku*, dan perbaikan kata tidak baku dengan normalisasi Levenshtein Distance. Sedangkan pada pengujian yang lain menghasilkan akurasi yang lebih rendah karena dipengaruhi oleh beberapa faktor. Pada grafik *accuracy* semakin banyak tahapan proses yang dilakukan maka tingkat akurasi semakin naik.

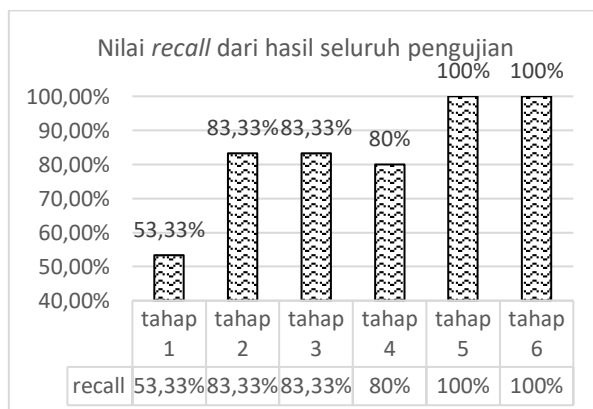
2. Hasil *precision*



Gambar 4. Nilai *precision* dari seluruh pengujian

Pada Gambar 4 menunjukkan hasil dari nilai *precision* yaitu tingkat ketepatan prediksi sistem, dengan menghitung prediksi benar dari total data yang diprediksi sistem termasuk prediksi salah. Hasil dari *precision* menunjukkan bahwa semakin banyak tahapan proses yang dilakukan nilai *precision* yang diperoleh cenderung naik.

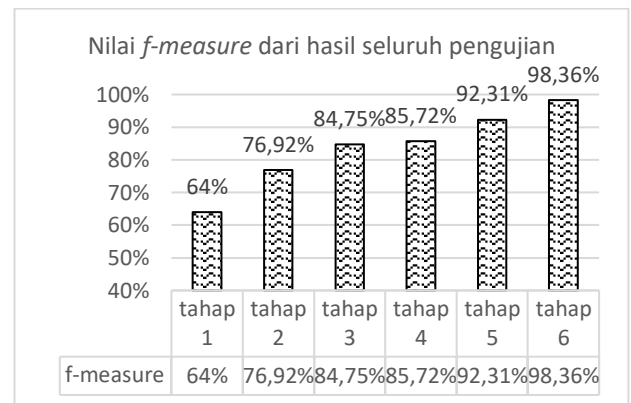
3. Hasil *recall*



Gambar 5. Nilai *recall* dari seluruh pengujian

Pada Gambar 5 menunjukkan hasil dari nilai *recall* yaitu nilai tingkat keberhasilan mengenali suatu kelas yang harus dikenali. Hasil dari *recall* menunjukkan bahwa semakin banyak tahapan proses yang dilakukan maka nilai *recall* yang diperoleh cenderung naik. Yaitu pada tahap dilakukannya proses *pre-processing* dengan perbaikan kata tidak baku dan Levenshtein Distance menghasilkan nilai *recall* yang tinggi yaitu mencapai 100%.

4. Hasil *f-measure*



Gambar 6. Nilai *f-measure* dari seluruh pengujian

Pada Gambar 6 menunjukkan hasil dari nilai *f-measure* yaitu penggabungan *precision* dan *recall*. Atau nilai yang mewakili keseluruhan kinerja sistem. Hasil pada nilai *f-measure* menunjukkan nilai tertinggi berada pada pengujian dengan *pre-processing* dan perbaikan kata tidak baku dan Levenshtein Distance. Kemudian untuk nilai terendah berada pada pengujian tanpa *pre-processing* dan tanpa perbaikan kata tidak baku. Hasil dari *f-measure* menunjukkan bahwa semakin banyak tahapan proses yang dilakukan maka nilai *f-measure* semakin naik.

5. KESIMPULAN

Berdasarkan hasil pengujian dan analisis yang telah selesai dilakukan, dapat disimpulkan sebagai berikut :

1. Metode klasifikasi Naive Bayes dengan perbaikan kata tidak baku dapat diterapkan pada proses analisis sentimen tentang opini film pada dokumen Twitter berbahasa Indonesia. Data latih dan data uji dilakukan proses *pre-processing* terlebih dahulu, yang mana pada proses *pre-processing* terdapat tambahan perbaikan kata tidak baku menggunakan kamus_katabaku yang dilakukan setelah *case folding*. Dan perbaikan kata tidak baku menggunakan normalisasi Levenshtein Distance yang dilakukan setelah proses *pre-processing*.
2. Pada penggunaan proses *pre-processing* dan perbaikan kata tidak baku dengan penambahan normalisasi Levenshtein Distance terhadap hasil klasifikasi memberikan pengaruh akurasi yang lebih

baik, yaitu sebesar 98.33% dibandingkan dengan *pre-processing* yang hanya dilakukan dengan perbaikan kata tidak baku menggunakan kamus_katabaku saja. Yaitu dengan nilai akurasi sebesar 91.67%.

3. Berdasarkan pengujian yang telah dilakukan, diperoleh hasil akurasi terbaik dari analisis sentimen tentang opini film pada dokumen Twitter berbahasa Indonesia menggunakan Naive Bayes dengan perbaikan kata tidak baku adalah sebesar 98.33% Dan untuk *precision*, *recall*, dan *f-measure* adalah 96.77%, 100%, dan 98.36%.

6. DAFTAR PUSTAKA

- Adriyani, N. M., Santiyasa, I. W. & Muliantara, A., 2012. *Implementasi Algoritma Levenshtein Distance dan Metode Impiris untuk Menampilkan Saran Perbaikan Kesalahan Pengetikan Doukem Berbahasa Indonesia*. Fakultas Ilmu Komputer, Universitas Udayana.
- Agarwal, A. et al., 2014. *Sentiment Analysis of Twitter Data*. Department of Computer Science Columbia University. Available at: <http://www.cs.columbia.edu/~julia/papers/Agarwaletal11.pdf>.
- Aggarwal, C., 2015. *Data Classification: Algorithms and Applications*. Berilustrasi penyunt. New York: CRC Press.
- Asih, R., 2012. Indonesia Pengguna Twitter Terbesar Kelima Dunia. [Online]. <http://www.tempo.co/read/news/2012/02/02/072381323/> [Diakses 21 Februari 2017].
- Buntoro, G. A., Adji, T. B., & Purnamasari, A. E. 2014. *Sentiment Analysis Twitter dengan Kombinasi Lexicon Based dan Double Propagation*. CTIEE. Halaman 39-43.
- Destuardi dan Surya, S. 2009. *Klasifikasi Emosi Untuk Teks Bahasa Indonesia Menggunakan Metode Naive Bayes*. Teknik, Institut Teknologi Sepuluh Nopember, Surabaya.
- Firmansyah, R. F. N., 2016. *Sentiment Analysis Pada Review Aplikasi Mobile Menggunakan Metode Naive Bayes dan Query Expansion*. Malang: Fakultas Ilmu Komputer Universitas Brawijaya.
- Freman AT, Condon SL, Ackerman CM. 2006. Cross linguistic name matching in English and Arabic: a “one to many mapping” extension of the levenshtein edit distance algorithm. *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL*; 2006 Jun; New York, Amerika Serikat. New York (US): Association for Computational Linguistic. hlm 471-478.
- Hadna, N. et al., 2016. *Studi literatur tentang perbandingan metode proses analisis sentimen di twitter*. Fakultas Teknik, Universitas Gadjah Mada.
- Hall, M., 2006. *A Decision Tree-Based Attribute Weighting Filter for Naive Bayes. Knowledge-Based Systems*, pp.120–126. Available at: <http://www.cs.waikato.ac.nz/pubs/wp/2006/uow-cs-wp-2006-05.pdf>.
- Kwak, H., Lee, C., Park, H., dan Moon S. 2010. *What is Twitter, a Social Network or a News Media?*. Department of Computer Science, KAIST, Korea.
- Nugraha, M., 2014. *Sentiment Analysis Pada Review Film Dengan Menggunakan Metode K-Nearest Neighbor*. Bandung : Universitas Widyatama Bandung.
- Perdana, R. S., & Pinandito, A. 2017. *Combining Likes-Retweet Analysis and Naive Bayes Classifier Within Twitter for Sentiment Analysis*. Malang : Fakultas Ilmu Komputer Universitas Brawijaya.
- Taheri, S. & Mammadov, M., 2013. *Learning The Naive Bayes Classifier With Optimization Models*. In International Journal of Applied Mathematics and Computer Science. pp. 787–795.