

# Implementasi Algoritma *Naive Bayes* Terhadap Analisis Sentimen Opini Film Pada Twitter

Fajar Ratnawati

Program Studi Rekayasa Perangkat Lunak, Politeknik Negeri Bengkalis, Riau

Email : [fajar@polbeng.ac.id](mailto:fajar@polbeng.ac.id)

**Abstrack** - The movie is one of the most interesting topics to talk about. When someone writes opinions of a movie, all the elements in the movie will be written down. Opinion movie to be taken on this research is from twitter. The number of opinions written on Twitter requires classification according to the sentiments that are owned for easy to get the tendency of opinion to the movie whether tending to positive or negative opinion. The algorithm to be used in this research is Naive Bayes Algorithm. Based on the experimental results, the sentimental analysis that can be done by the system with accuracy is 90% with the details of 92% precision value, 90% recall and f-measure 90%.

**Keywords:** sentiment analysis, naive bayes, movie opinion, twitter

**Intisari** - Film merupakan salah satu topik yang sangat menarik untuk dibicarakan. Ketika seseorang menulis opini suatu film, maka semua unsur yang ada di dalam film tersebut akan dituliskan. Data opini film pada penelitian ini diambil dari komentar film yang ditulis di *twitter*. Banyaknya opini yang dituliskan di *twitter* membutuhkan pengklasifikasian sesuai sentimen yang dimiliki agar mudah untuk mendapatkan kecenderungan opini tersebut terhadap film apakah cenderung beropini positif atau negatif. Algoritma yang akan digunakan pada penelitian ini adalah Algoritma *Naive Bayes*. Berdasarkan hasil eksperimen, analisis sentimen yang dapat dilakukan oleh sistem dengan akurasi yang didapat adalah 90 % dengan rincian nilai *precision* 92%, *recall* 90% dan *f-measure* 90% .

Kata Kunci : analisis sentimen, *naive bayes*, opini film, *twitter*

## I. PENDAHULUAN

Perkembangan dunia teknologi informasi dan komunikasi yang pesat tidak terlepas dari penyedia layanan web yang menyediakan informasi yang beragam. Informasi yang menyebabkan penambahan data yang kebanyakan berupa data teks dapat dijadikan sumber yang sangat potensial untuk digali lebih dalam. Salah satu contohnya adalah data text yang diambil dari *twitter*.

*Twitter* adalah layanan jejaring sosial dan microblog daring yang memungkinkan penggunaannya untuk mengirim dan membaca pesan berbasis teks hingga 140 karakter, yang dikenal dengan sebutan kicauan (*tweet*). *Twitter* didirikan pada bulan Maret 2006 oleh Jack Dorsey, dan situs jejaring sosialnya diluncurkan pada bulan Juli. Sejak diluncurkan, *twitter* telah menjadi salah satu dari sepuluh situs yang

paling sering dikunjungi di internet, dan dijuluki dengan pesan singkat dari internet . Di *twitter*, pengguna tak terdaftar hanya bisa membaca kicauan, sedangkan pengguna terdaftar bisa menulis kicauan melalui antarmuka situs web, pesan singkat (SMS) atau melalui berbagai aplikasi untuk perangkat seluler [1].

Opini adalah pendapat, ide atau pikiran untuk menjelaskan kecenderungan atau preferensi tertentu terhadap perspektif dan ideologi akan tetapi bersifat tidak objektif karena belum mendapatkan pemastian atau pengujian, dapat pula merupakan sebuah pernyataan tentang sesuatu yang berlaku pada masa depan dan kebenaran atau kesalahannya serta tidak dapat langsung ditentukan misalnya menurut pembuktian melalui induksi [2].

Film merupakan salah satu topik yang sangat menarik untuk dibicarakan. Ketika

seseorang menulis opini suatu film, maka semua unsur yang ada di dalam film tersebut akan dituliskan. Banyak pengguna *twitter* yang memberikan tanggapan tentang film yang pernah mereka tonton dalam bentuk opini berupa pengalaman baik maupun buruk. Hal tersebut menjadikan film memiliki berbagai macam topik yang informasinya bisa digali kembali.

Data dari *twitter* ini memiliki karakteristik yang tidak terstruktur dan banyak memuat *noise* sehingga dibutuhkan *teks mining* yang memiliki peran penting dalam bidang data mining.

*Teks mining* merupakan proses ekstraksi pola (informasi dan pengetahuan yang berguna) dari sejumlah data tak terstruktur yang nantinya akan diperoleh pola-pola data, tren dan ekstraksi pengetahuan yang potensial dari data teks [3]. Masukan untuk penambangan teks adalah data yang tidak (atau kurang) terstruktur, seperti dokumen, word, PDF, kutipan teks dll sedangkan masukan untuk penambangan data adalah data yang terstruktur [4]. Salah satu tujuan penggunaan *teks mining* adalah analisis sentimen.

Analisis sentimen adalah riset komputasional dari opini, sentimen dan emosi yang diekspresikan secara tekstual [5]. Untuk melakukan analisis sentimen ada beberapa algoritme yang dapat digunakan salah satunya adalah algoritme *Naive Bayes*.

*Naive Bayes classifier* merupakan sebuah metode klasifikasi yang berakar pada teorema Bayes. Metode pengklasifikasian dengan menggunakan metode probabilitas dan statistik yaitu memprediksi peluang berdasarkan pengalaman di masa sebelumnya (Teorema Bayes) dengan ciri utamanya adalah asumsi yang sangat kuat (naif) akan ketergantungan dari masing-masing kondisi/kejadian [6].

*Natural Language Processing* (NLP) adalah penerapan ilmu komputer, khususnya linguistik komputasional (*computational linguistics*), untuk mengkaji

interaksi antara komputer dengan bahasa (alami) manusia. NLP berupaya memecahkan masalah untuk memahami bahasa alami manusia, dengan segala aturan gramatikal dan semantiknya, dan mengubah bahasa tersebut menjadi representasi formal yang dapat diproses oleh komputer [7].

Penelitian dalam bidang analisis sentimen dewasa ini banyak dilakukan karena topik ini sangat menarik untuk dibahas salah satunya adalah penelitian Ratnawati, dkk (2017) tentang pengklasifikasian opini film yang data komentarnya diambil dari komentar yang dituliskan di *twitter* berdasarkan sentimen positif, sentimen negatif dan sentimen netral pada level kalimat. Analisis sentimen dilakukan dengan menggunakan metode *deep learning* dengan akurasi yang didapat 80,99%.

Penelitian yang dilakukan Falahah, dkk (2015) tentang pengklasifikasian opini publik pada *twitter* terkait layanan pemerintah terhadap masyarakat berdasarkan sentimen positif, sentimen negatif dan sentimen netral menggunakan pendekatan *Naive Bayes*.

Sistem yang dibuat pada penelitian ini untuk melakukan analisis sentimen yaitu sentimen positif dan negatif menggunakan algoritma *Naive Bayes* dengan data opini diambil dari *twitter*.

## II. SIGNIFIKASI STUDI

### A. Studi Literatur

#### 1. Analisis Sitem

Penelitian ini mengimplementasikan metode *Naive Bayes Classifier* kedalam suatu sistem untuk mengklasifikasikan data kedalam sentimen positif dan sentimen negatif berdasarkan data ulasan komentar film yang telah dikumpulkan.

Sistem yang akan dibangun terdiri dari empat proses utama, yaitu pengumpulan data, *preprocessing*, pelabelan dan klasifikasi data. Data yang akan digunakan adalah data *tweet* komentar film berbahasa Indonesia. Data *tweet* ini diperoleh dengan membuat program scraping menggunakan *library scrapy* yang disediakan oleh *python*.

Penelitian ini diawali dengan melakukan pengumpulan data *tweet* dengan cara *scraping tweet* yang mengandung hastag #judul film yang diinginkan menggunakan *library scrapy* yang disediakan oleh *Python*. Pengumpulan data ini dilakukan dengan memanfaatkan *API Search Twitter*. Data *tweet* yang dikumpulkan antara lain *id, username, index, data\_id, url, tanggal dan waktu tweet, isi tweet, original\_tweet*. Dalam proses *scraping*, secara otomatis akan mengambil data *tweet* yang mengandung kata judul film dengan jangka waktu data yang diambil bisa ditentukan. Data *tweet* yang terkumpul secara otomatis akan tersimpan dalam *database* *mongodb*. Dari *mongodb* data akan diekspor ke *file.txt* dan nantinya akan melewati tahap pelabelan dan selanjutnya *preprocessing*. *Database* yang digunakan diberi nama sesuai judul film yang diambil dimana didalam *database* tersebut terdapat 1 *collection* (tabel) yang akan disimpan data hasil *scraping*-nya. Pada setiap tabel terdapat 8 *fields* (kolom) yaitu *id, user\_name, index, data\_id, url, time\_tweet, text\_tweet, original\_tweet*. Pada *fields id* *MongoDB* secara otomatis membuat *primary key* di *fields\_id* dan akan terisi secara otomatis. Untuk *fields username* berisi nama pengguna *twitter*. *Index* berisi urutan *tweet*. *Data\_id* berisi data pengguna *twitter*. *Url* berisi alamat file internet. *Time\_tweet* adalah waktu *tweet* ditulis. *Text\_tweet* adalah isi dari *tweet* dan *original\_tweet* adalah keseluruhan isi *tweet*. Tahap berikutnya adalah pelabelan *tweet* secara manual sesuai dengan sentimen yang sudah ditentukan sebelumnya yaitu sentimen positif untuk opini film yang bersifat positif dan sentimen negatif untuk opini film yang bersifat negatif. Setelah seluruh data memiliki kelas masing-masing kemudian dilakukan pemisahan data menjadi dua bagian yaitu data *training* dan data *testing*. Data latih digunakan untuk mengklasifikasikan *tweet* pada kelas sentimennya dan jumlah datanya lebih banyak. Dalam penelitian ini untuk data latih sebanyak 80% dari total data

keseluruhan. Data uji digunakan untuk mengukur sejauh mana *classifier* berhasil melakukan klasifikasi dengan benar. Jumlah data yang digunakan dalam penelitian ini adalah 20% dari data keseluruhan. Selanjutnya kedua bagian tersebut dikenakan proses *preprocessing*. Pemrosesan teks merupakan proses menggali, mengolah, mengatur informasi dengan cara menganalisis hubungannya, aturan-aturan yang ada di data tekstual semi terstruktur atau tidak terstruktur. Untuk lebih efektif dalam pemrosesan dilakukan langkah transformasi data ke dalam suatu format yang memudahkan untuk kebutuhan pemakai. *Preprocessing* merupakan salah satu langkah yang penting dalam analisis sentimen. Adapun tahapan *preprocessing*-nya antara lain:

*Case folding*

Pada tahap ini, semua huruf akan diubah menjadi *lowercase* atau huruf kecil. Berikut adalah langkah-langkah *case folding*:

1. Memeriksa ukuran setiap karakter dari awal sampai akhir karakter.
2. Jika ditemukan karakter yang menggunakan huruf kapital (*uppercase*), maka huruf tersebut akan diubah menjadi huruf kecil (*lower case*)

Gambaran proses tahap *case folding* dapat dilihat pada tabel I.

TABEL I. CONTOH CASE FOLDING

Data Input	Data Output
Film ini Sarat dengan petualangan seru nan mendebarkan ala Alien	film ini sarat dengan petualangan seru nan mendebarkan ala alien

2. Normalisasi Fitur

Ada beberapa komponen khas yang biasa ada di *tweet* yakni *username*, URL (*Uniform Resource Locator*), dan karakter khusus seperti *hashtag* (#), *retweet*(RT), tanda baca (.,?! dll), karakter lainnya (\$,%\*, dll). Karena *username*, URL, karakter khusus, tanda baca dan karakter lainnya tidak memiliki pengaruh apapun terhadap nilai sentimen, maka komponen-komponen khas diatas akan dibuang. Komponen *username* diidentifikasi dengan

kemunculan karakter '@'. Selain *username*, karakter '@' biasa juga digunakan untuk pemanggilan tempat seperti @CinemaXXI. Namun nama tempat tersebut tidak memiliki pengaruh pada analisis sehingga nama tempat juga harus dihapus. Pada komponen URL dikenali melalui ekspresi reguler (seperti http, www). Berikut langkah-langkah pada tahap normalisasi fitur:

- a. Data yang digunakan hasil dari *case folding*.
- b. Hasil dari *case folding* akan diperiksa apakah terdapat *username*, URL, karakter khusus, tanda baca dan karakter lainnya.
- c. Jika terdapat *username*, URL, karakter khusus, tanda baca dan karakter lainnya maka akan dihilangkan.

Gambaran proses tahap normalisasi fitur dapat dilihat pada Tabel II.

TABEL II. CONTOH NORMALISASI FITUR

Data Input	Data Output
Film ini Sarat dengan petualangan seru nan mendebarkan ala Alien @CGV Blitz#5thave	film ini sarat dengan petualangan seru nan mendebarkan ala alien

Gambaran proses *preprocessing* dapat dilihat pada Gambar 1.



Gambar 1. Proses *Preprocessing*

Setelah tahapan *preprocessing* selesai kemudian dilakukan implementasi algoritme *Naive Bayes Classifier*. Pada proses *training* dilakukan untuk membangun model probabilitas dari data *training*. Dan selanjutnya dilakukan pengujian model klasifikasi yang dihasilkan dari proses *training* dengan menggunakan data baru atau yang disebut data *testing*. Pengujian dilakukan dengan *5 fold cross validation*.

**B. Metodologi Penelitian**

**1. Evaluasi Peforma Kualifikasi**

Dalam pengukuran performa klasifikasi terdapat beberapa cara, namun cara yang paling sering digunakan adalah dengan

menghitung akurasi, *precision*, *recall* dan *f-measure*. [8]. Akurasi merupakan persentase dari total sentimen yang benar dikenali. Perhitungan akurasi dilakukan dengan cara membagi jumlah data sentimen yang benar dengan total data dan data uji. Untuk menghitung nilai akurasinya dilakukan dengan menggunakan Persamaan (1).

$$akurasi = \frac{\text{jumlah sentimen benar}}{\text{jumlah data tes}} \times 100\% \quad (1)$$

Untuk pengukuran performa klasifikasi cara yang digunakan selain menghitung akurasi adalah menghitung *precision*, *recall* dan *f-measure*. *Precision* merupakan perbandingan jumlah data relevan yang ditemukan terhadap jumlah data yang ditemukan. Perhitungan *precision* dilakukan dengan cara membagi jumlah data benar yang bernilai positif dibagi dengan jumlah data benar yang bernilai positif dan data salah yang bernilai positif. Nilai dari data salah bernilai positif diambil dari jumlah nilai selain *true positive* kolom yang sesuai tiap kelasnya. Untuk menghitung nilai *precision* dapat dilakukan dengan menggunakan Persamaan (2).

$$precision = \frac{\text{true positive}}{\text{true positive} + \text{false positive}} \quad (2)$$

*Recall* merupakan perbandingan jumlah materi relevan yang ditemukan terhadap jumlah materi yang relevan. Perhitungan *recall* dilakukan dengan cara membagi data benar bernilai positif dengan hasil penjumlahan dari data benar yang bernilai positif dan data salah yang bernilai negatif. Nilai dari data salah yang bernilai negatif diambil dari jumlah nilai selain *true positive* baris yang sesuai tiap kelasnya. Perhitungan *recall* dapat menggunakan Persamaan (3).

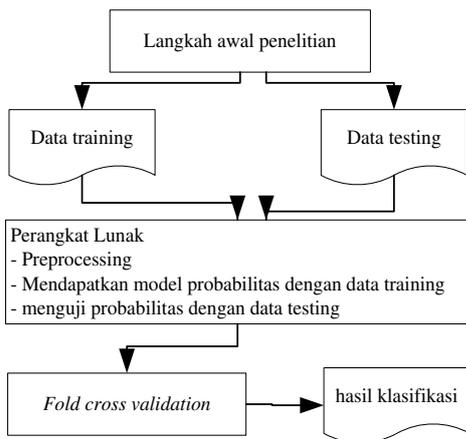
$$recall = \frac{\text{true positive}}{\text{true positive} + \text{false negative}} \quad (3)$$

*F-measure* merupakan parameter tunggal ukuran keberhasilan *retrieval* yang menggabungkan *recall* dan *precision*. Nilai *F-measure* didapat dari perhitungan

hasil perkalian *precision* dan *recall* dibagi dengan hasil penjumlahan *precision* dan *recall* kemudian dikalikan dua dan perhitungan *f-measure* menggunakan Persamaan (4).

$$f - \text{measure} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \quad (4)$$

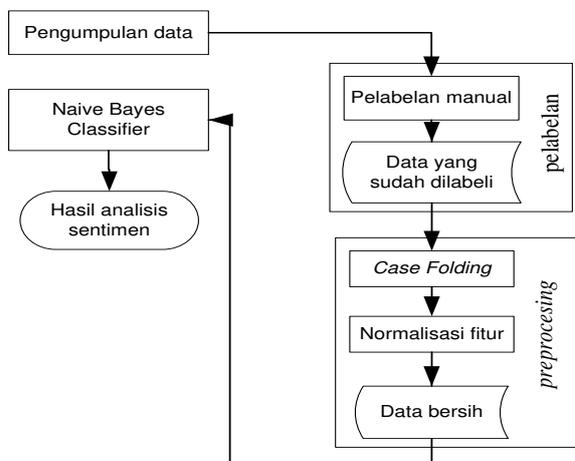
Untuk lebih jelasnya alur penelitian dapat dilihat pada Gambar 2.



Gambar 2. Alur Penelitian

## 2. Arsitektur Sistem

Secara umum sistem ini terdiri dari empat bagian diantaranya adalah pengambilan data, *preprocessing*, pelabelan dan klasifikasi data. Adapun desain arsitektur sistem dapat dilihat pada gambar 3.



Gambar 3. Arsitektur Sistem

## 3. Data dan Kelas data

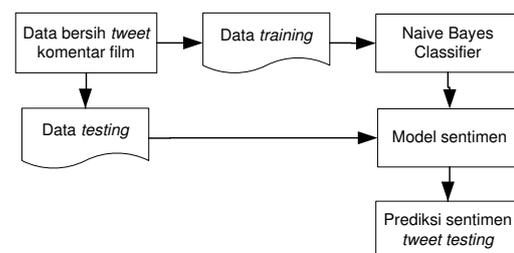
Data *input* merupakan data yang digunakan sebagai masukan pada sistem yang dibangun. Data tersebut kemudian akan diolah dan digunakan lebih lanjut untuk mendapatkan output yang diinginkan. Data *input* yang digunakan adalah data *tweet* opini film berbahasa Indonesia. Data *tweet* komentar film diperoleh dengan melakukan *scraping* pada halaman *search.twitter.com* sesuai dengan judul film yang dimasukkan.

*Tweet* yang dikumpulkan adalah *tweet* yang mengandung hashtag #judul film. Pengambilan data *tweet* yang akan digunakan dalam penelitian ini dimulai sejak tanggal 1 Januari 2015 sampai 31 Desember 2015. Data yang dipakai sebanyak 500 kalimat.

Data dibagi menjadi dua sentimen yaitu sentimen positif untuk komentar atau opini film yang bersifat positif dan sentimen negatif untuk komentar atau opini film yang bersifat negatif.

## 4. Pemodelan Data

Pemodelan data dibuat dengan tujuan untuk menguji keakuratan prediksi sistem berdasarkan data model yang sudah dibuat. Proses *training* untuk membangun model probabilitas pada penelitian ini dibutuhkan data *training*. Selanjutnya dilakukan pengujian model klasifikasi sentimen yang dihasilkan pada proses *training* dengan menggunakan data *tweet* baru (data *testing*). Gambaran umum sistem dapat dilihat pada Gambar 4.



Gambar 4. Pemodelan data

Pada Gambar 4 dapat dilihat data *tweet* komentar film yang dibutuhkan ada dua

bagian yaitu data untuk proses *training* atau pelatihan dan data untuk proses *testing* atau pengujian. Pada data *training*, data yang digunakan harus sudah terlebih dahulu dilabeli dan juga harus melalui proses *preprocessing* untuk dijadikan *input* dari penggunaan metode *Naive Bayes Classifier* dan disimpan sebagai model data dengan ekstensi *.pickle* (dot *pickle*). Pada proses *training* sistem akan membaca data *tweet* yang dimasukkan selanjutnya sistem hanya akan mengambil kata yang terdiri dari 3 atau lebih karakter. Jika kata yang dimasukkan kurang dari 2 karakter maka akan dihapus secara otomatis oleh sistem. Selanjutnya sistem akan melakukan proses ekstraksi fitur kata dan selanjutnya menyimpan model data.

Pada bagian proses *testing* atau pengujian digunakan data baru yang tidak terdapat pada proses *training*. Berbeda dengan data *training*, data *testing* tidak diperlukan pelabelan tetapi harus melalui proses *preprocessing*. Ketika data *testing* dimasukkan kedalam sistem, proses yang akan dilalui adalah penentuan fitur, mengekstrak fitur dan mengklasifikasikannya sesuai dengan model yang sudah dibuat. Sistem akan memberikan output dari hasil klasifikasi sistem yaitu negatif atau positif sesuai dengan model data.

##### 5. *Naive Bayes Classifier*

Pada pengujian *Naive Bayes classifier* ini menggunakan library NLTK. Tujuan pengujian ini adalah untuk dapat secara otomatis mengklasifikasikan *tweet* sebagai sentimen *tweet* yang positif atau negatif. Dalam metode *Naive Bayes Classifier* dilakukan proses pengklasifikasian teks berdasarkan data latih yang sebelumnya sudah disimpan. Pada implementasinya terdapat tiga tahap yaitu membuat daftar tuple tunggal, membuat daftar fitur kata, dan membuat *classifier*.

*Classifier* perlu dilatih dan untuk melakukannya diperlukan daftar *tweet* yang diklasifikasikan secara manual. Pada implementasinya digunakan sekitar 200

*tweet* positif dan 200 *tweet* negatif untuk melatih *classifier* dan disimpan di file *.txt* dengan nama file positif.txt dan negatif.txt. Kemudian diatur juga untuk data tes sehingga dapat dinilai ketepatan dari *classifier* terlatih.

Pada implementasinya diambil kedua data yaitu data positif dan data negatif dan membuat daftar *tuple* tunggal dengan masing-masing berisi dua elemen. Elemen pertama adalah array yang berisi kata-kata dan elemen kedua berisi jenis sentimen. Kata yang kurang dari tiga huruf dihilangkan dan menggunakan huruf kecil semua. Daftar fitur kata perlu diekstrak dari *tweet*. Fitur ini berisi daftar setiap kata-kata unik yang diurutkan berdasarkan frekuensi kemunculan. Menggunakan fungsi untuk menampilkan daftar fitur kata, fungsi ekstrak kata dari *tweet* dan fungsi untuk menghitung frekuensi kemunculan.

Untuk membuat *classifier*, kita perlu menentukan fitur apa saja yang sesuai atau relevan. Untuk melakukan itu pertama yang diperlukan adalah ekstraksi fitur. Yang nantinya akan kita gunakan kembali sebagai kamus untuk menunjukkan kata-kata apa yang terkandung dalam *input*. Yang dimaksud disini adalah *input tweet*.

*Naive Bayes classifier* menggunakan *prior probability* (yaitu nilai probabilitas yang diyakini benar sebelum melakukan eksperimen) dari setiap label yang merupakan frekuensi masing-masing label pada *training set* dan kontribusi dari masing-masing fitur.

Berdasarkan dari ciri alami dari sebuah model probabilitas, klasifikasi *Naive Bayes* bisa dibuat lebih efisien dalam bentuk pembelajaran *supervised* atau terawasi. Dalam beberapa bentuk praktiknya, parameter untuk perhitungan model *Naive Bayes* menggunakan metode *maximum likelihood* atau kemiripan tertinggi

Untuk ranah klasifikasinya yang dihitung adalah  $P(H|X)$ , yaitu peluang bahwa hipotesa benar (valid) untuk data *sample X* yang diamati dapat diterapkan pada persamaan 5.

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)} \quad (5)$$

Keterangan:

- X = Data sample dengan klas (label) yang tidak diketahui
- H = Hipotesa bahwa X adalah data dengan klas (label) C
- P(H|X) = Peluang bahwa hipotesa benar (valid) untuk data sampel X yang diamati
- P(X|H) = Peluang data sample X, bila diasumsikan bahwa hipotesa benar (valid).
- P(H) = Peluang dari hipotesa H
- P(X) = Peluang data sample yang diamati

### III. HASIL DAN PEMBAHASAN

Setelah semua rancangan selesai disusun dan sistem juga telah dibangun, maka tahap berikutnya adalah menggunakan sistem tersebut untuk menghasilkan model probabilitas dari data training, serta menguji akurasi model probabilitas dengan data testing.

Pada pengujian ini digunakan sekitar 200 *tweet* positif dan 200 *tweet* negatif untuk melatih *classifier* dan disimpan dalam file.txt sehingga total data untuk *classifier* adalah 400. Metode ini diimplementasikan menggunakan *python* dan library NLTK. Dengan tujuan untuk dapat secara otomatis mengklasifikasikan *tweet* sebagai sentimen *tweet* yang positif dan negative.

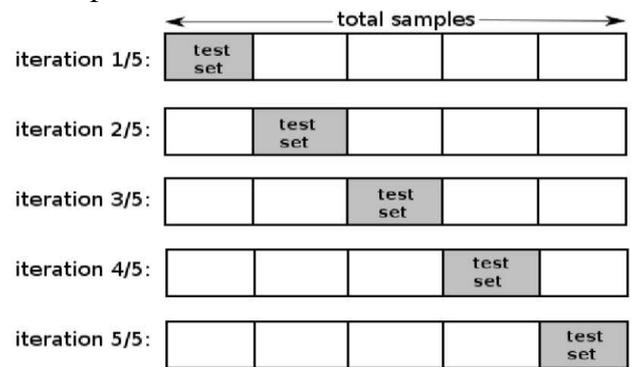
#### 1. K-Fold Cross Validation

*K-fold* adalah salah satu metode *Cross Validation* yang populer dengan melipat data sebanyak k dan mengulangi (iterasi) eksperimennya sebanyak k juga.

Pada pengujian ini data yang akan digunakan adalah 500 *tweet* termasuk data uji yang nantinya akan dibagi menjadi 5 bagian atau k=5 sehingga data yang diperoleh adalah 500 data dibagi menjadi 5 lipatan dengan masing-masing isinya adalah 100 data. Selain itu akan ditentukan mana yang termasuk data *training* dan mana yang termasuk data *testing* dengan perbandingan 80-20, sehingga didapatkan 400 data sebagai data *training* dan 100 sisanya

sebagai data testing. Berdasarkan 5 lipatan yang sudah ditentukan maka akan ada 4 lipatan dikalikan 100 = 400 data training dan sisanya 1 lipatan dikalikan 100 = 100 data *testing*.

Pengujian menggunakan data yang sudah dipartisi akan diulang sebanyak 5 kali (k=5) dengan posisi data tes berbeda disetiap iterasinya. Misalkan iterasi pertama data tes pada posisi awal, iterasi kedua data tes di posisi kedua begitu seterusnya. Hingga iterasi kelima Gambaran iterasi k-fold dapat dilihat pada Gambar 5.



Gambar 5. Iterasi k-fold

Untuk pengujian fold ke-1 sampai fold ke-5 jumlah data tes yang dimasukkan adalah 100 *tweet* dengan posisi data tes seperti digambarkan pada Gambar 5 dan hasilnya dapat dilihat pada Gambar 6 sampai Gambar 10.

```
Total accuracy: 90.000000% (90/100).
precision  recall  f1-score  support
positive   0.83    1.00    0.91     50
negative   1.00    0.80    0.89     50
avg / total 0.92    0.90    0.90    100
waktu:           : 1.14 detik
```

Gambar 6. Hasil pengujian fold ke-1

```
Total accuracy: 72.000000% (72/100).
precision  recall  f1-score  support
positive   0.82    0.56    0.67     50
negative   0.67    0.88    0.76     50
avg / total 0.75    0.72    0.71    100
waktu:           : 1.00 detik
```

Gambar 7. Hasil pengujian Fold ke-2

Total accuracy: 82.000000% (82/100).

	precision	recall	f1-score	support
positive	0.88	0.74	0.80	50
negative	0.78	0.90	0.83	50
avg / total	0.83	0.82	0.82	100

waktu: : 1.05 detik

Gambar 8. Hasil pengujian Fold ke-3

Total accuracy: 80.000000% (80/100).

	precision	recall	f1-score	support
positive	0.86	0.72	0.78	50
negative	0.76	0.88	0.81	50
avg / total	0.81	0.80	0.80	100

waktu: : 1.25 detik

Gambar 9. Hasil pengujian fold ke-4

Total accuracy: 88.000000% (88/100).

	precision	recall	f1-score	support
positive	0.87	0.90	0.88	50
negative	0.90	0.86	0.88	50
avg / total	0.88	0.88	0.88	100

waktu: : 1.20 detik

Gambar 10. Hasil pengujian fold ke-5

Untuk hasil pengujian keseluruhan *fold* dapat dilihat pada Tabel III.

TABEL III. HASIL PENGUJIAN *NAIVE BAYES*

Fold	Waktu (detik)	Akurasi (%)	Precision (%)	Recall (%)	f-measure (%)
I	1,49	90	92	90	90
II	1,00	72	75	72	71
III	1,05	82	83	82	82
IV	1,25	80	81	80	80
V	1,20	88	88	88	88
<b>RR</b>	<b>1,20</b>	<b>82,4</b>	<b>83,8</b>	<b>82,4</b>	<b>82,2</b>

Dari Tabel 3 dapat dilihat bahwa waktu tercepat yang dibutuhkan sistem untuk memproses data adalah pada saat *fold* 2 dengan waktu 0,88 detik diikuti oleh *fold* ke-3 dengan waktu 0,97 detik, *fold* ke-4 dan ke-5 dengan waktu 1,03 detik dan waktu terlama yang dibutuhkan sistem untuk memproses data adalah pada *fold* ke-1 yaitu 1,49 detik dengan rata-rata waktu yang dibutuhkan dari kelima *fold* adalah 1,08 detik.

Akurasi yang didapat dari pengujian data tes yang tertinggi pada saat pengujian *fold* ke-1 yaitu 90%, diikuti *fold* ke-5 dengan akurasi sebesar 88%, *fold* ke-3 dengan akurasi sebesar 82%, *fold* ke-4 dengan akurasi 80% dan akurasi terendah pada saat pengujian *fold* ke-2 yaitu 72% dengan rata-rata akurasi pengujian kelima *fold* adalah 82,4%.

*Precision* tertinggi pada saat pengujian *fold* ke-1 yaitu 92%, diikuti *fold* ke-5 sebesar 88%, *fold* ke-3 sebesar 83%, *fold* ke-4 sebesar 81% dan *precision* terendah pada saat pengujian *fold* ke-2 yaitu 75% dengan rata-rata *precision* kelima *fold* adalah 83,8%

*Recall* tertinggi pada saat pengujian *fold* ke-1 sebesar 90%, diikuti hasil *recall* pada pengujian *fold* ke-5 sebesar 88%, *fold* ke-3 sebesar 82%, *fold* ke-4 sebesar 80% dan *recall* terendah diperoleh pada saat pengujian *fold* ke-2 sebesar 72% dengan rata hasil *recall*nya 82,4%.

Hasil pengujian *f-measure* tertinggi dihasilkan pada saat pengujian *fold* ke-1 sebesar 90%, diikuti pengujian *fold* ke-5 sebesar 88%, *fold* ke-3 sebesar 82%, *fold* ke-4 sebesar 80% dan hasil pengujian *f-measure* terendah pada saat pengujian *fold* ke-2 sebesar 71% dengan rata-rata hasil pengujian *f-measure* 82,2%.

### B. Evaluasi

Metode penelitian yang dimulai dari pengumpulan data *tweet*, *preprocessing* sampai melakukan perhitungan nilai probabilitas menggunakan *Naive Bayes Classifier* masih terdapat berbagai permasalahan yang membuat sistem dalam penelitian ini bekerja tidak maksimal, yaitu:

#### 1. Kurangnya data *training*

Data *training* yang digunakan pada penelitian ini terbilang kurang karena hanya 400 data yang digunakan, sehingga pada saat data uji yang dimasukkan tidak dikenali pada data *training*, maka hasil prediksi dan label sebenarnya tidak sesuai. Semakin banyak data *training* yang digunakan maka semakin tinggi akurasi yang dihasilkan dan ketepatan sistem dalam

mengenali data uji akan lebih baik karena sistem mampu mengenali banyak kalimat dan kosakata yang bervariasi yang dijadikan sebagai pembelajaran oleh sistem.

## 2. Terdapat Dataset yang tidak tepat

Terdapat kesalahan dalam proses klasifikasi sistem yang disebabkan oleh penggunaan dataset yang kurang tepat. Keadaan ini yang menyebabkan banyak ditemukan fitur kemunculan data yang bukan termasuk kategorinya pada data uji yang digunakan. Contohnya dataset yang digunakan dalam data positif terdapat kata "lucu", kemudian pada data negatif juga terdapat kata "lucu" yang berasal dari kata "tidak lucu". Kata yang sama pada dataset positif dan negatif inilah yang mengakibatkan kesalahan dalam proses klasifikasi data. Kalimat "filmnya sangat lucu" yang sebenarnya terklasifikasi menjadi kalimat positif akan dikenali sistem sebagai kalimat negatif karena pada saat sistem bekerja kalimat tersebut akan masuk dalam klasifikasi negatif, karena dalam kalimat tersebut terdapat kata "lucu" dan kata "sangat tidak lucu" ada dalam dataset negatif. Jadi dapat disimpulkan bahwa permasalahan penggunaan dataset yang didalamnya terdapat kata yang sama maka akan mempengaruhi proses klasifikasi pada sistem. Permasalahan tersebut juga dapat mempengaruhi tingkat akurasi sistem, sehingga mengakibatkan kinerja sistem tidak berjalan secara maksimal.

## 3. Kendala memahami klasifikasi sentimen

Kendala yang sering ditemui dalam memahami kalimat untuk diolah dalam proses klasifikasi sentimen yaitu ketika menjumpai kalimat awal yang memberikan persepsi bahwa kalimat tersebut merupakan kalimat positif kemudian diakhir kalimat memberikan persepsi bahwa kalimat tersebut termasuk kalimat negatif ataupun kalimat yang awalnya memberikan persepsi bahwa kalimat tersebut merupakan kalimat negatif kemudian diakhir kalimat memberikan persepsi bahwa kalimat

tersebut termasuk kalimat positif. Kendala tersebut mengakibatkan kinerja sistem tidak berjalan secara maksimal karena sistem mendeteksi klasifikasi sentimen pada kalimat yang berada diawal saja. Salah satu contohnya yaitu kalimat pada data training "saya bilang ini film keren walaupun tidak seratus persen menghibur". Dalam klasifikasi data manual kalimat tersebut masuk dalam kategori sentimen negatif tetapi pada saat klasifikasi menggunakan sistem, kalimat tersebut masuk dalam kategori sentimen positif karena pada awal kalimat terdapat kata yang mengandung sentimen positif

## IV. KESIMPULAN

Sesuai dengan tujuan penelitian ini yaitu mengimplementasikan algoritme *Naive Bayes Classifier* terhadap analisis sentimen data opini film berbahasa Indonesia pada *twitter* telah berhasil dilakukan. Dari hasil pengujian dan analisis yang telah dilakukan dapat disimpulkan beberapa hal yaitu pengklasifikasian data opini film berbahasa Indonesia berdasarkan sentimennya dapat dilakukan dengan algoritme *Naive Bayes Classifier* dengan pembagian datasetnya menggunakan *5-fold cross validation*. Semakin banyak data training yang digunakan maka akan mempengaruhi kinerja dari sistem. Hasil akurasi akan semakin tinggi dan itu menandakan sistem berhasil melakukan klasifikasi dengan baik. Akurasi tertinggi didapat pada fold kedua yaitu 90%, *precision* 92%, *Recall* 90% dan *f-measure* 90%.

## REFERENSI

- [1] D'Monte, Leslie, *Swine Flu's Tweet Tweet Causes Online Flutter*. New Delhi, 2009
- [2] Ratnawati, F, Winarko, E, Analisis Opini Film pada *Twitter* Menggunakan Algoritme Dynamic Convolutional Neural Network, *Thesis*, Universitas Gadjah Mada, Yogyakarta, Indonesia, Jan 2018

- [3] Turban, E.; et.al. *Decision Support and Business Intelligence Systems (edisi ke-9)*. New Jersey: Pearson Education, Inc. 2011
- [4] R. Feldman, and J. Sanger, "The Text Mining Handbook Advances Approaches in Analyzing Unstructured Data." Cambridge University Press, New York, 2007.
- [5] Liu. B, *Sentiment Analysis and Subjectivity*, 2010.
- [6] Rish, I., An Empirical Study of The Naive Bayes Classifier, In *Proceedings of IJCAI-01 workshop on Empirical Methods in AI*, New York, 2001.
- [7] J. Pustejovsky, and A. Stubbs, "Natural Language Annotation for Machine Learning." Cambridge University Press, 2012.
- [8] Witten. H.I and Frank. E, *Data Mining Practical Machine Learning Tools and Techniques* second edition, Elsevier, San Fransisco, 2005
- [9] M. Wegmuller, J. P. von der Weid, P. Oberson, and N. Gisin, "Highresolution fiber distributed measurements with coherent OFDR," in *Proc. ECOC'00*, 2000, paper 11.3.4, p. 109
- [10] Nurrohmat, M. A., Aplikasi Pemrediksi Masa Studi dan Predikat Kelulusan Mahasiswa Informatika Universitas Muhammadiyah Surakarta Menggunakan Metode Naïve Bayes. *Khazanah Informatika*, 2015
- [11] Patel, T., Undavia, J., & Patela, A., Sentiment Analysis of Parents Feedback for Educational Institutes. *International Journal of Innovative and Emerging Research in Engineering*, 2015
- [12] Falahah dan Nur, A. D., Pengembangan Aplikasi Sentiment Analysis Menggunakan Metode Naive Bayes. *Seminar Nasioonal Sistem Informasi Indonesia*, 2015