IJCCS, Vol.7, No.1, January 2013, pp. 13~22
ISSN: 1978-1520

Klasifikasi Posting Twitter Kemacetan Lalu Lintas Kota Bandung Menggunakan *Naive Bayesian Classification*

Sandi Fajar Rodiyansyah*[1], Edi Winarko[2]
[1] Program Ilmu Komputer-Universitas Pendidikan Indonesia
[2]Jurusan Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta
e-mail: [1]**galuh29@yahoo.com**, [2]ewinarko@ugm.ac.id

Abstrak

Setiap hari server Twitter menerima data tweet dengan jumlah yang sangat besar, dengan demikian, kita dapat melakukan data mining yang digunakan untuk tujuan tertentu. Salah satunya adalah untuk visualisasi kemacetan lalu lintas di sebuah kota.

Naive bayes classifier adalah pendekatan yang mengacu pada teorema Bayes, dengan mengkombinasikan pengetahuan sebelumnya dengan pengetahuan baru. Sehingga merupakan salah satu algoritma klasifikasi yang sederhana namun memiliki akurasi tinggi. Untuk itu, dalam penelitian ini akan membuktikan kemampuan naive bayes classifier untuk mengklasifikasikan tweet yang berisi informasi dari kemacetan lalu lintas di Bandung.

Dari hasil uji coba, aplikasi menunjukan bahwa nilai akurasi terkecil 78% dihasilkan pada pengujian dengan sampel sebanyak 100 dan menghasilkan nilai akurasi tinggi 91,60% pada pengujian dengan sampel sebanyak 13106. Hasil pengujian dengan perangkat lunak Rapid Miner 5.1 diperoleh nilai akurasi terkecil 72% dengan sampel sebanyak 100 dan nilai akurasi tertinggi 93,58% dengan sampel 13106 untuk metode naive bayesian classification. Sedangkan untuk metode support vector machine diperoleh nilai akurasi terkecil 92% dengan sampel sebanyak 100 dan nilai akurasi tertinggi 99,11% dengan sampel sebanyak 13106.

Kata kunci— *Twitter, tweet, klasifikasi, naive bayesian classification, support vector machine*

Abstract

Every day the Twitter server receives data tweet with a very large number, thus, we can perform data mining to be used for specific purpose. One of which is for the visualization of traffic jam in a city.

Naive bayes classifier is an approach that refers to the bayes theorem, is a combination of prior knowledge with new knowledge. So that is one of the classification algorithm is simple but has a high accuracy. With this, in this research will prove the ability naive bayes classifier to classify the tweet that contains information of traffic jam in Bandung.

The testing result, the program shows that the smallest value of the accuracy is 78% on testing by using a sample 100 record and generate high accuracy is 91,60% on the testing by using a sample 13106 record. The testing results with Rapid Miner 5.1 software obtained the smallest value of the accuracy is 72% by using a sample 100 records and the high accuracy is 93.58% by using a sample 13.106 records for naive bayesian classification. And for the method of support vector machine obtained the smallest value is 92% accuracy by using a sample 100 records and the high accuracy of 99.11% by using a sample 13.106 records.

Keywords—*Twitter, tweet, classification, naive bayesian classification, support vector machine*

Received November 1st,2012; Revised December 1st, 2012; Accepted December 15th, 2012

1. PENDAHULUAN

Saat ini dunia internet sedang berada pada pada fase *user generated content*, yang berarti seluruh konten yang berada di internet adalah buatan pengguna secara umum. Salah satu aplikasi internet yang mendukung *user generated content* adalah *microblogging*.

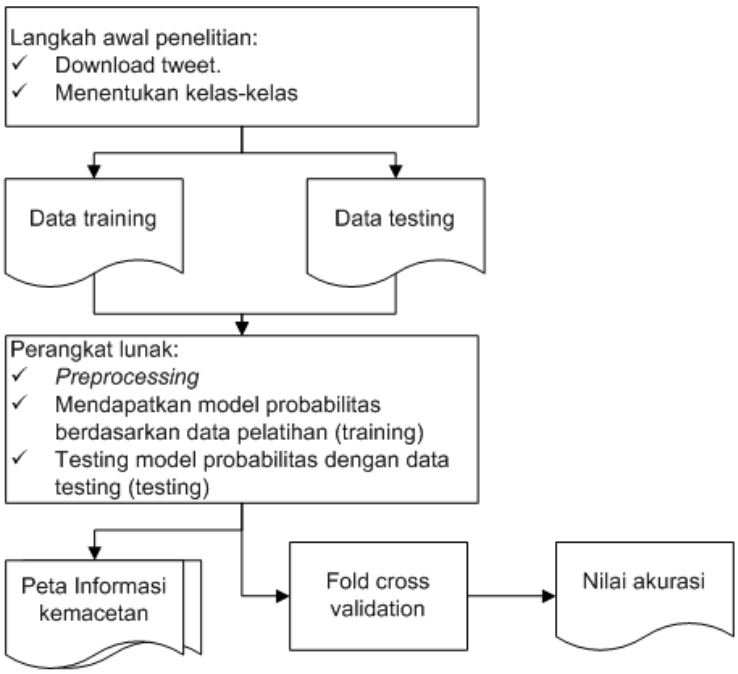
Saat ini, *microblogging* menjadi populer sebagai alat komunikasi antara pengguna internet. Pada pertengahan tahun 2010 Twitter memiliki pengguna lebih dari 106 juta pengguna diseluruh dunia dan terus meningkat setiap harinya sebanyak 300.000 pengguna dan Twitter setiap harinya mendapatkan lebih dari 3 juta *request*. Dari angka tersebut Indonesia menjadi negara yang menduduki peringkat 8 dalam mengakses situs Twitter. Twitter menerima *tweet* dari pengguna sebanyak 55 juta pesan setiap harinya [1]. Dengan demikian, kita dapat memanfaatkan data *tweet* tersebut untuk kepentingan tertentu misalkan untuk memvisualisasikan kemacetan lalu lintas berdasarkan *tweet* yang dikirim pengguna.

Data mining merupakan sebuah proses dari *knowledge discovery* (penemuan pengetahuan) dari data yang sangat besar [2]. Sementara itu, text mining merupakan bidang data mining yang bertujuan untuk mengumpulkan informasi yang berguna dari data teks dalam bahasa alami atau proses analisis data teks kemudian mengekstrak informasi yang berguna untuk tujuan tertentu [3].

Dalam penelitian ini akan dikembangkan aplikasi dengan menggunakan teknik klasifikasi *naive bayesian classifiers*. Pendekatan ini merupakan pendekatan mengacu pada teorema Bayes yang merupakan prinsip peluang statistika untuk mengkombinasikan pengetahuan sebelumnya dengan pengetahuan baru. Prinsip ini kemudian digunakan untuk memecahkan masalah klasifikasi [4].

Penggunaan algoritma ini dinilai sesuai karena naive bayesian classifier merupakan salah satu algoritma klasifikasi yang sederhana namun memiliki kemampuan dan akurasi tinggi [5]. Aplikasi yang dikembangkan dalam penelitian ini akan melakukan pengklasifikasian data tweet yang mengandung tentang informasi lalu lintas di kota Bandung. Setelah data terklasifikasikan, kemudian dilakukan visualisasi di peta kota Bandung dengan melalui API Google Map.

2. METODE PENELITIAN



Gambar 1 Alur Penelitian

Berdasarkan pada Gambar 1 penelitian ini diawali dengan melakukan download tweet secara real time tweet yang mengandung hastag #lalinbdg dan #bantra. Pengumpulan data ini dilakukan dengan memanfaatkan API Search Twitter. Data tweet yang dikumpulkan antara lain username, isi tweet, URI tweet, tanggal dan waktu tweet dan URI profil pengguna.

Tahap berikutnya adalah melakukan pe-label-an tweet sesuai dengan kelas yang ditentukan sebelumnya yaitu kelas 'macet' untuk tweet yang menginformasikan kemacetan lalu lintas, 'lancar' untuk tweet yang menginformasikan kelancaran lalu lintas dan 'uknown' untuk tweet yang tidak menginformasikan keadaan lalu lintas.

Setelah seluruh data sudah memiliki kelas, kemudian dilakukan pemisahan mejadi dua bagian yaitu data training dan data testing. Selanjutnya, kedua bagian data tersebut kemudian dilakukan text preprocessing dari data tersebut. Adapun tahapan preprocessing antara lain :

1. Penghapusan kata tertentu yang tidak dipakai dalam proses klasifikasi.
2. Konversi menjadi huruf kecil.
3. Menghapus url (http://bit.ly/mHibqV).
4. Melakukan perbaikan data apabila ada kata yang diperlukan tetapi data tidak sesuai, misalkan ada kata "maceeeettt" kemudian di ubah menjadi "macet".
5. Menghapus mention (@xxx).
6. Menghapus karakter selain a-z.
7. Mengganti sinonim (yg=yang, jln=jalan).
8. Menghilangkan stopword.
9. Menghapus kata dengan satu karakter.

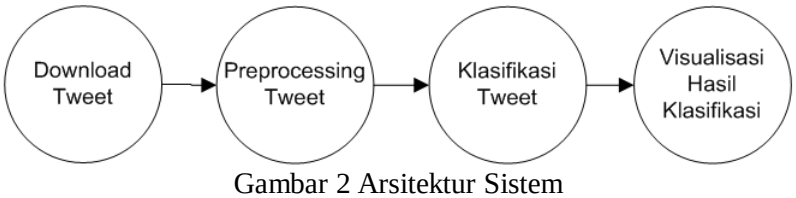
Setelah data training dan testing bersih. Kemudian dilakukan implementasi algoritma naive bayes classifier pada proses training untuk membangun model probabilitas dari data training. Dan selanjutnya dilakukan pengujian model klasifikasi yang dihasilkan pada proses training dengan menggunakan data tweet baru (data testing). Pengujian model klasifikasi ini dilakukan pada sistem yang dikembangkan dan pengujian akurasi model klasifikasi dari hasil data bersih yang dihasilkan dengan menggunakan perangkat lunak lain yang digunakan sebagai data pembanding. Pengujian ini dilakukan dengan menggunakan teknik 10 fold cross validation. Dan untuk menghitung nilai akurasinya dilakukan dengan menggunakan persamaan (1).

$$Akurasi = \frac{Jumlah\ Klasifikasi\ Benar}{Jumlah\ Data\ Uji} \times 100\% \quad (1)$$

Kemudian tahap akhir adalah dilakukan visualisasi dari hasil klasifikasi yang sudah dilakukan oleh sistem ke dalam peta jalan kota Bandung melalui Google Map dengan menggunakan API Google Map.

2.1 Arsitektur Sistem

Secara umum sistem ini terdiri dari empat bagian diantaranya adalah download (pengambilan data), preprocessing tweet, klasifikasi tweet dan visualisasi hasil klasifikasi. Adapun desain arsitektur sistem dapat terlihat pada Gambar 2.



Gambar 2 Arsitektur Sistem

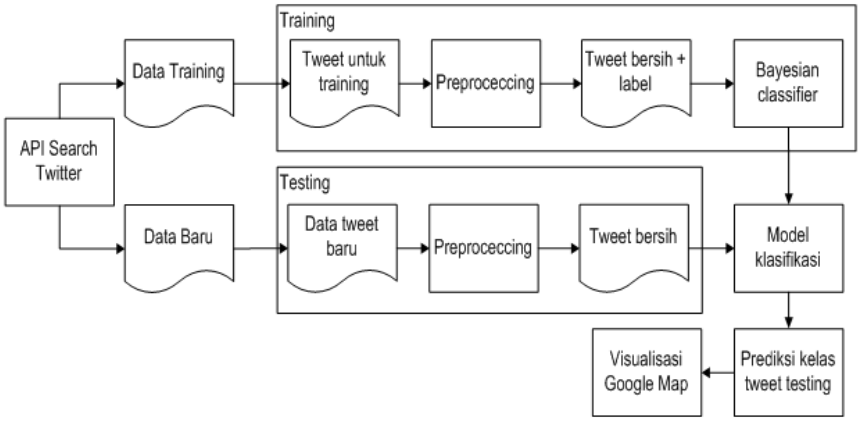
2.2 Data dan Kelas Data

Tweet yang dikumpulkan adalah tweet yang mengandung hastag #lalinbdg dan hastag #bantra. Pengambilan data tweet yang akan digunakan dalam penelitian ini dimulai sejak tanggal 24 Mei 2011 sampai tanggal 1 September 2011 yang menghasilkan data tweet sebanyak 15401 record.

Data diklasifikasi menjadi tiga kelas/label yaitu kelas ‘macet’ untuk tweet yang menginformasikan keadaan lalu lintas yang macet di suatu jalan atau tempat, kelas ‘lancar’ untuk tweet yang menginformasikan keadaan lalu lintas yang lancar di suatu jalan atau tempat dan kelas ‘unknown’ untuk tweet yang yang tidak mengandung ciri untuk kelas ‘macet’ maupun kelas ‘lancar’.

2.3 Perancangan Sistem

Secara garis besar sistem ini bertujuan untuk menentukan kelas pada tweet menggunakan teknik data mining yaitu naive bayesian classification untuk mencari pola prediksi tempat-tempat yang cenderung macet, yang kemudian divisualisasikan dengan menggunakan Google Map. Gambaran umum sistem dapat dilihat pada Gambar 3.



Gambar 3 Gambaran Umum Sistem

2.4 Naive Bayesian Classification

Teorema Bayes merupakan teorema yang mengacu pada konsep probabilitas bersyarat [4]. Secara umum teorema Bayes dapat dinotasikan pada persamaan (2).

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)} \quad (2)$$

Salah satu metode klasifikasi yang dapat digunakan adalah metode *naive bayes* yang sering disebut sebagai *naive bayes classification* (NBC). Kelebihan NBC adalah sederhana tetapi memiliki akurasi yang tinggi [5].

Pada *naive bayes classification* setiap *tweet* direpresentasikan dalam pasangan atribut $< a_1, a_2, a_3 \ldots, a_n >$ dimana a_1 adalah kata pertama a_2 adalah kata kedua dan seterusnya, sedangkan V adalah himpunan kelas [6]. Pada saat klasifikasi, metode ini akan menghasilkan kategori/kelas yang paling tinggi probabilitasnya (V_{MAP}) dengan memasukan atribut $< a_1, a_2, a_3, \ldots, a_n >$. Adapun rumus V_{MAP} dapat dilihat pada persamaan (3).

$$V_{MAP} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j | a_1, a_2, a_3, \ldots, a_n) \quad (3)$$

Dengan menggunakan teorema Bayes, maka persamaan (3) dapat ditulis menjadi,

$$V_{MAP} = \underset{v_j \in V}{\operatorname{argmax}} \frac{P(a_1, a_2, a_3, .., a_n | v_j) \times P(v_j)}{P(a_1, a_2, a_3, .., a_n)} \quad (4)$$

$P(a_1, a_2, a_3, .., a_n)$ nilainya konstan untuk semua v_j sehingga persamaan (4) dapat dinyatakan juga menjadi persamaan (5).

$$V_{MAP} = \underset{v_j \in V}{\operatorname{argmax}} P(a_1, a_2, a_3, .., a_n | v_j) \times P(v_j) \quad (5)$$

Naive bayes classifier menyederhanakan hal ini dengan mengasumsikan bahwa didalam setiap kategori, setiap atribut bebas bersyarat satu sama lain [4]. Dengan kata lain,

$$P(a_1, a_2, a_3, .., a_n|v_j) = \prod_i P(a_i|v_j) \quad (6)$$

Kemudian apabila persamaan (5) disubstitusikan ke persamaan (6), maka akan menghasilkan

$$V_{MAP} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j) \times \prod_i P(a_i|v_j) \quad (7)$$

P(v_j) dan probabilitas kata a_i untuk setiap kategori P($a_i|v_j$) dihitung pada saat *training*. Dimana,

$$P(v_j) = \frac{docs_j}{training} \quad (8)$$

$$P(a_i|v_j) = \frac{n_i + 1}{n + kosakata} \quad (9)$$

Dimana $docs_j$ adalah jumlah dokumen pada kategori j dan *training* adalah jumlah dokumen yang digunakan dalam proses *training*. Sedangkan n_i adalah jumlah kemunculan kata a_i pada kategori v_j, n adalah jumlah kosakata yang muncul pada kategori v_j dan kosakata adalah jumlah kata unik pada semua data *training*. Ringkasan algoritma untuk *naive bayes classifier* dapat dilihat di Gambar 4.

TrainMultinominalNB(C,D)
1. V ← ExcractVocabulary(D)
2. N ← CountDocs(D)
3. For each c ϵ C
4. Do N_c ← CountDocsInClass (D,c)
5. Prior[c] ← N_c /N
6. $Text_c$ ← ConcatenateTextOfAllDocsInClass(D,c)
7. For each t ϵ V
8. Do T_{ct} ← CountTokensOfTerm($Text_c$, t)
9. For each t ϵ V
10. Do $condprob[t][c] \leftarrow \frac{T_{ct}+1}{\sum_{t'}(T_{ct'}+1)}$
11. Return V, prior, condprob

ApplyMultinominalNB(C,V,prior, condprob, d)
1. W ← ExtractTokensFromDoc(V,d)
2. For each c ϵ C
3. Do score[c] ← log prior[c]
4. For each t ϵ W
5. Do score[c] += log condprob[t][c]

Return argmax $_{c \epsilon C}$ score[c]

Gambar 4 Algoritma naive bayes classifier [6]

3. 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*, menguji akurasi model probabilitas dengan data *testing* serta melakukan percobaan dengan menggunakan data yang sama dengan aplikasi lain yang bersifat *free* dan membandingkan hasil akurasinya dengan aplikasi yang dibuat.

3.1 Pengujian Running Time Preprocessing

Pada pengujian ini dilakukan perhitungan running time preprocessing dari berbagai porsi jumlah sampel yang telah ditentukan sebelumnya. Tabel 1 tertuang hasil pengujian running time preprocessing.

Tabel 1. *Running time preprocessing*

No	Sampel	Hasil Sampel	Rata-rata *preprocessing*	Rata-rata TFIDF
1	100	100	0,470 *tweet*/detik	0,000 *tweet*/detik
2	1000	1000	0,549 *tweet*/detik	0,047 *tweet*/detik
3	5000	5000	0,773 *tweet*/detik	0,266 *tweet*/detik
4	10000	10000	1,035 *tweet*/detik	0,516 *tweet*/detik
5	15401	13106	1,114 *tweet*/detik	0,550 *tweet*/detik

Dari Tabel 1 dapat dilihat bahwa semakin besar jumlah sampel yang digunakan, semakin besar juga rata-rata *tweet* yang dapat diolah pada proses *preprocessing* dan tf-idf dalam satu detik. Pada *preprocessing* data dengan jumlah sampel 100 sampai 10000 data bersih yang dapat terpakai pada adalah sebanyak jumlah sampel yang diinginkan, tetapi pada *preprocessing* dengan jumlah sampel 15401 data bersih yang dapat terpakai adalah sebanyak 13106. Hal ini karena apabila data diambil semua (15401 *record*) dari jumlah data tersebut dapat mengandung *tweet* pertanyaan yang pada *preprocessing tweet* pertanyaan tersebut tidak dipakai (dihapus). Dengan demikian, data tidak dapat terambil semua.

3.2 Pengujian Akurasi Model Probabilitas

Pada pengujian ini dilakukan pengujian akurasi model probabilitas terhadap dari data bersih yang terbentuk dengan berbagai porsi jumlah. Seperti pada *preprocessing*, porsi jumlah data pada pengujian akurasi model probabilitas terdiri dari 100, 1000, 5000, 10000 dan 13106 sampel data yang digunakan yang dihasilkan dari *preprocessing*. Berikut merupakan rincian hasil pengujian dengan berbagai porsi jumlah data.

Tabel 2. Merupakan hasil pengujian akurasi model dengan menggunakan teknik 10 *fold cross validation.*

Tabel 2. Rata-rata nilai akurasi model

sampel	Rata-rata akurasi (%)
100	78,00
1000	83,42
5000	88,60
10000	90,58
13106	91,60

Dari Tabel 2 dapat diketahui bahwa semakin besar jumlah sampel yang digunakan semakin besar pula rata-rata nilai akurasinya dan nilai akurasi tertinggi terdapat pada pengujian dengan menggunakan sampel data sebanyak 13106.

Tabel 3. Rata-rata nilai akurasi kelas

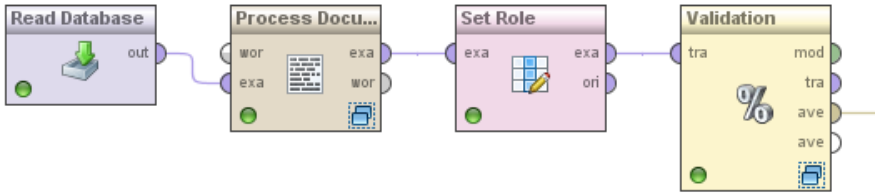
Jumlah sampel	Akurasi Macet (%)	Akurasi Lancar (%)	Akurasi Unknown (%)
100	100,00	46,67	24,17
1000	99,40	53,18	48,44
5000	99,72	76,41	65,77
10000	99,62	83,67	70,81
13106	99,59	85,73	74,01

Dengan memperhatikan Tabel 3 dapat dilihat bahwa untuk semua pengujiian dengan berbagai porsi jumlah sampel, nilai akurasi yang paling besar untuk semua kelas terjadi pada kelas macet. Dengan nilai yang paling besar terdapat pada pengujian dengan sampel 100 yang mencapai 100%. Pada kelas lancar dan unknown nilai akurasi cenderung meningkat seiring ditambahnya jumlah data sampel yang digunakan.

3.3 Pengujian dengan Perangkat Lunak Lain

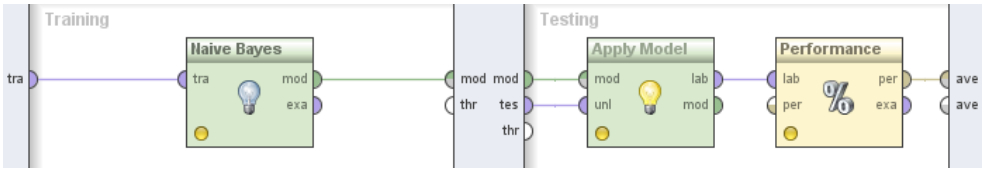
Sementara itu, pada pengujian dengan menggunakan perangkat lunak lain dalam penelitian ini adalah dengan menggunakan Rapid Miner 5.1. Metode yang digunakan untuk mengklasifikasikan data penelitian ini adalah dengan menggunakan metode *naive bayes classifier* dan metode *suport vector machine*.

Pada pengujian ini, data yang digunakan adalah data bersih yang telah melalui *preprocessing* oleh sistem yang dikembangkan yang tersimpan di Tabel data_bersih. Kemudian proses *read database* mengambil data tersebut. Proses selanjutnya adalah mengkonversi data menjadi dokumen dengan *process document from data*. Proses selanjutnya adalah menentukan field mana yang akan dijadikan sebagai kelas dengan menggunakan proses *set role*. Selanjutnya, tahap akhir adalah tahap pengujian dengan menggunakan proses *validation*. Untuk lebih jelas, keseluruhan proses pengujian dengan perangkat lunak Rapid Miner 5.1 dapat dilihat di Gambar 5.



Gambar 5. Proses klasifikasi dengan Rapid Miner 5.1

3.3.1 Metode naive bayes classifier



Gambar 6. Proses klasifikasi dengan naive bayes classifier Rapid Miner 5.1

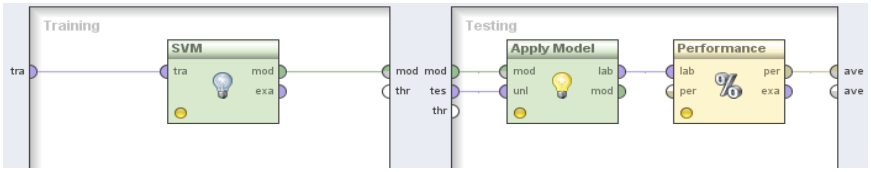
Gambar 6 menunjukan proses klasifikasi dengan Rapid Miner 5.1 dengan metode *naive bayes classifier* yang digunakan untuk mengklasifikasikan data penelitian ini sehingga diperoleh akurasi pada perangkat lunak ini.

Tabel 4. Akurasi klasifikasi dengan *naive bayes classifier* Rapid Miner 5.1

Sampel	Akurasi Rapid Miner 5.1 (%)			
	Macet	Lancar	Unknown	Model
100	76,62	61,54	50,00	72,00
1000	84,83	87,03	64,91	84,10
5000	92,53	67,45	90,43	89,90
10000	94,75	76,08	92,74	92,66
13106	95,90	74,52	94,33	93,58

Dengan memperhatikan Tabel 4 dapat dilihat bahwa nilai akurasi model terkecil terdapat pada hasil pengujian dengan menggunakan sampel sebanyak 100 dan nilai akurasi terbesar terdapat pada hasil pengujian dengan menggunakan sampel sebanyak 13106. Sementara itu, terjadi peningkatan akurasi kelas seiring sampel ditambah.

3.3.2 Metode support vector machine



Gambar 7. Proses klasifikasi dengan *support vector machine* Rapid Miner 5.1

Gambar 7 menunjukan proses klasifikasi dengan Rapid Miner 5.1 dengan metode *support vector machine* yang digunakan untuk mengklasifikasikan data penelitian ini sehingga diperoleh akurasi pada perangkat lunak ini.

Tabel 5. Akurasi klasifikasi dengan support vector machine Rapid Miner 5.1

Sampel	Akurasi Rapid Miner 5.1 (%)			
	Macet	Lancar	Unknown	Model
100	93,94	87,50	88,46	92,00
1000	100,00	97,14	95,85	98,70
5000	99,97	99,14	95,98	98,76
10000	99,97	99,58	96,63	99,00
13106	99,98	99,47	97,02	99,11

Dengan memperhatikan Tabel 5 dapat dilihat bahwa nilai akurasi model terkecil terdapat pada hasil pengujian dengan menggunakan sampel sebanyak 100 dan nilai akurasi terbesar terdapat pada hasil pengujian dengan menggunakan sampel sebanyak 13106. Terjadi peningkatan juga pada seluruh akurasi kelas seiring dengan ditambahnya jumlah sampel yang digunakan. Terjadi nilai akurasi kelas yang mencapai 100% pada kelas macet dengan pengujian yang menggunakan sampel sebanyak 1000 *record*.

3.4 Visualisasi Google Map

Dari data *tweet* yang telah terklasifikasikan, kemudian data tersebut digunakan untuk divisualisasikan di peta kota Bandung dengan memanfaatkan API Google Map. Data yang ditampilkan di peta adalah data jumlah *tweet* yang mengandung informasi kemacetan di suatu jalan. Gambar 8 merupakan salah satu tampilan visualisasi kemacetan kota Bandung pada tanggal 1 Juni 2011 berdasarkan aplikasi ini.

Gambar 8 Visualisasi hasil klasifikasi pada peta tanggal 1 Juni 2011

3.5 Implementasi untuk Kota Lain

Seperti sudah dijelaskan pada bagian sebelumnya pada penelitian ini dilakukan pengolahan data *tweet* yang menginformasikan keadaan lalu lintas kota Bandung. Hal ini terlihat dari data *tweet* yang dijadikan data *training* dan data *testing* menggunakan *tweet* yang mengandung *hastag* #lalinbdg dan *hastag* #bantra yang biasa digunakan oleh pengguna Twitter yang berasal dari kota Bandung atau yang sedang berada di Bandung.

Oleh karena itu, apabila sistem ini digunakan untuk kota lain, ada beberapa hal yang harus diperhatikan, antara lain :

1. Pengguna Twitter di kota tersebut harus memiliki kesamaan dalam menggunakan *hastag* dalam menginformasikan keadaan lalu lintas di sekitarnya. Hal ini dilakukan untuk proses pengambilan data dari server Twitter melalui API Search Twitter dengan *query hastag*.
2. Pada proses download, *query* yang dikirimkan ke API Search Twitter harus disesuaikan dengan *hastag* yang dipakai pada *tweet* yang menginformasikan keadaan lalu lintas di kota tersebut.
3. Untuk membangun model probabilitas perlu dilakukan *preprocessing* data dan proses *training* untuk membangun model probabilitas yang akan digunakan untuk menentukan kelas pada data yang baru.
4. Perlu pengaturan ulang pada data jalan untuk menyesuaikan nama jalan pada kota tersebut dan koordinat jalan tersebut pada Google Map sehingga nama dan koordinat jalan sesuai dengan kota tersebut.
5. Pada bagian yang digunakan untuk mengakses API Google Map, perlu dilakukan pengaturan ulang untuk mengakses peta Google Map agar koordinat diarahkan ke kota tersebut.

4. KESIMPULAN

1. Proses kerja sistem diawali dengan mengambil data (*download*) *tweet* dari server Twitter dengan memanfaatkan API twitter. Data *tweet* yang diambil disimpan di database. Data *tweet* kemudian diolah dengan menggunakan *preprocessing*. Data bersih yang dihasilkan oleh *preprocessing* kemudian diolah dengan menggunakan *naive bayes classifier* sehingga membentuk model probabilitas klasifikasi. Model probabiliyas klasifikasi ini kemudian digunakan untuk menentukan kelas pada *tweet* yang baru yang belum diketahui kelasnya.
2. Metode *naive bayes classification* mengasumsikan bahwa didalam setiap kelas, setiap kata unik bebas bersyarat satu sama lain. Asumsi ini digunakan agar proses *training* untuk membuat model klasifikasi dapat dijalankan.
3. Dari hasil pengujian akurasi sistem yang dikembangkan, menghasilkan nilai akurasi terkecil sebesar 78% pada proses pengujian dengan menggunakan sampel sebanyak 100 dan menghasilkan nilai akurasi tertinggi sebesar 91,60% pada proses pengujian dengan menggunakan sampel sebanyak 13106.
4. Dalam hal akurasi kelas dari sistem yang dikembangkan, dapat disimpulkan bahwa dari lima kali proses pengujian yang dilakukan terlihat bahwa semuanya menghasilkan nilai akurasi lebih dari 99% pada kelas macet. Sedangkan pada nilai akurasi kelas lancar dan kelas unknown menghasilkan nilai akurasi yang cenderung meningkat seiring ditambahnya jumlah sampel.
5. Hasil pengujian akurasi dengan menggunakan perangkat lunak Rapid Miner 5.1 dengan metode *naive bayes classifier* menghasilkan nilai akurasi terkecil sebesar 72% pada proses pengujian dengan menggunakan sampel sebanyak 100 dan menghasilkan nilai akurasi tertinggi sebesar 93,58% pada proses pengujian dengan menggunakan sampel sebanyak 13106.

6. Hasil pengujian akurasi dengan menggunakan perangkat lunak Rapid Miner 5.1 dengan metode *support vector machine* menghasilkan nilai akurasi terkecil sebesar 92% pada proses pengujian dengan menggunakan sampel sebanyak 100 dan menghasilkan nilai akurasi tertinggi sebesar 99,11% pada proses pengujian dengan menggunakan sampel sebanyak 13106.
7. Nilai akurasi sistem yang dikembangkan relatif hampir sama dengan nilai akurasi yang dihasilkan oleh perangkat lunak Rapid Miner 5.1 dengan metode *naive bayes classifier*.
8. Pada klasifikasi dengan menggunakan perangkat lunak Rapid Miner 5.1 maka dapat disimpulkan bahwa nilai akurasi metode *support vector machine* lebih baik dari pada nilai akurasi metode *naive bayes classifier.*

5. SARAN

Pengembangan perangkat lunak ini masih memiliki keterbatasan yang dapat dijadikan acuan untuk pengembangan dimasa yang akan datang, sehingga dapat disarankan beberapa hal sebagai berikut :

1. Perlu dilakukan analisis hubungan antar kata pada proses klasifikasi.
2. Untuk mendeteksi nama jalan, diperlukan proses untuk menstandarkan nama jalan, hal ini diperlukan untuk visualisasi.
3. Dari segi teknis programming, perlu dicoba melakukan proses training dan testing pada saat klasifikasi untuk menyimpan model klasifikasi yang dihasilkan di memori (pada penelitian ini disimpan di database). Hal ini dilakukan untuk mengoptimalkan proses klasifikasi.
4. Perlu dilakukan pengujian apakah hasil klasifikasi ini sesuai dengan keadaan di jalan sebenarnya.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada orang tua yang telah memberikan dukungan yang tidak terhingga dan juga kepada rekan-rekan mahasiswa S2 Ilmu Komputer UGM yang telah membantu dalam proses penelitian ini.

DAFTAR PUSTAKA

[1] Hepburn, A, 2010, Infographic: Twitter Statistics, Facts & Figures, http://www.digitalbuzzblog.com/infographic-twitter-statistics-facts-figures/ Di akses tanggal 9 Mei 2011.

[2] Han, J., & Kamber, M., 2006, *Data Mining: Concepts and Techniques 2e,* Morgan Kaufmann Publishers, San Francisco.

[3] Witten, I. H., 2005, *Text mining : Practical Handbook of Internet Computing,* Chapman & Hall/CRC Press, Florida.

[4] Tan, P. N., Steinbach, M., & Kumar, V., 2006, *Introduction to Data Mining,* Pearson Education, Boston.

[5] Rish, I., 2006, An empirical study of The Naive Bayes Classifier, *International Joint Conference on Artificial Intelligence,* California.

[6] Manning, C. D., Raghavan, P., & Schütze, H., 2008, *Introduction to Information Retrieval,* Cambridge University Press, Cambridge.