BAB 2

LANDASAN TEORI

2.1 *Text Mining*

Secara umum, *text mining* merupakan proses ekstraksi informasi dan pengetahuan dari teks yang tidak terstruktur (Hotho, Nürnberger, & Paaß, 2005). *Text mining*, atau *knowledge discovery from text* (KDT), menggunakan teknik *information retrieval, information extraction* serta *natural language processing*, dan menggabungkan hasilnya dengan algoritma *data mining*, statistik, serta *machine learning*.

Oleh karena bentuknya yang tidak terstruktur, tahap *preprocessing* merupakan tahap utama dalam mengidentifikasi dan mengekstraksi ciri (*feature*) pada dokumen. Melalui tahap ini, teks pada dokumen diubah agar memiliki format yang lebih terstruktur (Feldman & Sanger, 2007).

Text mining dapat diaplikasikan pada berbagai bidang, seperti *competitive intelligence, market analysis*, dan *customer relationship management* (Gupta & Lehal, 2009). Selain itu, *text mining* juga dapat diterapkan untuk meningkatkan hasil pencarian melalui *web*, klasifikasi sentimen, serta *bibliographic data mining* (Radovanovic & Ivanovic, 2008).

2.2 Analisis Sentimen

Analisis sentimen, atau *opinion mining*, merupakan bidang komputasi terhadap opini, sentimen, dan emosi yang diekspresikan pada teks (Liu, 2010). Sentimen diekspresikan melalui satu atau kombinasi kata-kata serta dapat diidentifikasi melalui opini dan emosi. Opini merupakan hal yang ditentukan oleh subjek, seperti suka, tidak suka, ingin, tidak ingin, benci, dan sebagainya, sedangkan emosi merupakan hal yang dirasakan oleh subjek, seperti senang, sedih, marah, kecewa, dan sebagainya (Hovy, 2015).

Analisis sentimen dapat diterapkan pada berbagai hal, antara lain komentar pada situs ulasan film (Yessenov & Misailovic, 2009), sentimen terhadap partai politik (Steven, 2014), berita *online* (Fong, Zhuang, & Li, 2013), dan prediksi hasil pemilihan presiden (Jahanbakhsh & Moon, 2014; Wang et al., 2012).

Klasifikasi sentimen umumnya diformulasikan menjadi 2 kelas, yaitu positif dan negatif. Kunci utama dalam klasifikasi sentimen terletak pada teknik pengumpulan *feature* yang efektif. Berikut contoh *feature* yang dimaksud (Liu, 2010).

1. Istilah dan frekuensinya

   Feature ini berupa satuan kata (*unigram*) dan *n-gram* dengan frekuensi kemunculannya. Posisi kata serta pembobotan menggunakan TF-IDF juga dapat dipertimbangkan.

2. *Part of Speech* (POS)

Kata yang berada pada POS berbeda dapat diperlakukan secara berbeda. Misalnya, penelitian A membuktikan bahwa kata sifat merupakan indikator penting opini, tetapi penelitian B menggunakan seluruh POS *tag* dan *n-gram* sebagai *feature*.

3. *Sentiment words and phrases*

   Umumnya berupa kata sifat dan kata keterangan, tetapi kata benda (misal: sampah, kotoran) dan kata kerja (misal: suka, benci) juga dapat digunakan untuk mengekspresikan sentimen.

4. *Rules of opinions*

   Selain kata dan frase bersentimen, komposisi bahasa atau ekspresi lain juga dapat digunakan untuk mengekspresikan sentimen.

5. *Sentiment shifters*

   Merupakan ekspresi yang digunakan untuk mengubah orientasi sentimen, misalnya kata negasi.

6. *Syntactic dependency*

   Feature yang bergantung pada kata yang dihasilkan dari uraian atau struktur kalimat juga diduga mengandung sentimen.

Analisis sentimen meliputi tahap-tahap berikut (Jurafsky, 2012).

1. *Tokenization*

   Tahap ini memecah kalimat menjadi *token* dan banyak berkaitan dengan HTML,

XML, *hashtag*, kapitalisasi, nomor telepon, tanggal, dan *emoticons*.

2. *Feature Extraction*

   Ekstraksi *feature* dapat dilakukan dengan beragam cara, disesuaikan dengan data. Tahap ini juga mencakup cara menangani kata negasi.

3. Klasifikasi

   Tahap klasifikasi dapat menggunakan berbagai *classifier*, antara lain *Naive Bayes, MaxEnt*, dan *Support Vector Machine* (SVM).

2.3 *Naive Bayes Classifier*

Dalam *machine learning*, *Naive Bayes Classifier* merupakan *classifier* probabilistik sederhana yang berdasarkan pada penerapan teori Bayes dengan asumsi independen yang kuat antar *feature*. *Naive Bayes Classifier* dapat dilatih dengan efisien secara *supervised*. Meskipun dirancang secara naif, *Naive Bayes Classifier* bekerja dengan cukup baik pada berbagai situasi dalam dunia nyata. Salah satu kelebihannya yaitu *Naive Bayes Classifier* hanya membutuhkan data *training* dalam jumlah kecil (*Naive Bayes Classifier*, n.d.).

Berikut cara kerja *Naive Bayes Classifier* (Han & Kamber, 2006).

1. Misalkan D merupakan data *training* yang terdiri dari sejumlah baris data yang sudah diberi *label*. Setiap baris memiliki atribut sebanyak n, contoh: $X = (x_1, x_2, x_3, ..., x_n)$ dengan atribut $A_1, A_2, ..., A_n$.

2. Terdapat sebanyak m kelas, yaitu $C_1, C_2, ..., C_m$. *Classifier* akan memprediksi data X termasuk pada kelas yang kemungkinannya paling tinggi dengan kondisi X, yaitu $P(C_i|\boldsymbol{X}) > P(C_j|\boldsymbol{X})$ di mana $1 \leq j \leq m, j \neq i$. Nilai $P(C_i|\boldsymbol{X})$ akan dimaksimalkan. Kelas C_i yang memiliki nilai $P(C_i|\boldsymbol{X})$ terbesar disebut *maximum posteriori hypothesis*.

$$P(C_i|\boldsymbol{X}) = \frac{P(\boldsymbol{X}|C_i)P(C_i)}{P(\boldsymbol{X})} \tag{1}$$

3. Nilai $P(C_i|\boldsymbol{X})$ konstan untuk semua kelas, hanya nilai $P(\boldsymbol{X}|C_i)P(C_i)$ yang perlu dimaksimalkan. Apabila probabilitas kelas sebelumnya tidak diketahui, maka diasumsikan bahwa kelas-kelas tersebut serupa, $P(C_1) = P(C_2) = ... = P(C_m)$, dan nilai $P(\boldsymbol{X}|C_i)$ yang akan dimaksimalkan. Probabilitas kelas sebelumnya dapat dihitung dengan $P(C_i) = \frac{|C_i,D|}{|D|}$ di mana $|C_i, D|$ merupakan jumlah baris data *training* kelas C_i pada D.

4. Apabila data memiliki banyak atribut, penghitungan $P(\boldsymbol{X}|C_i)$ akan menjadi rumit. Untuk mengurangi penghitungan, diasumsikan bahwa nilai atribut pada kelas tidak saling berhubungan.

$$P(\boldsymbol{X}|C_i) = \prod_{k}^{n} P(x_k|C_i) = P(x_1|C_i) \times P(x_2|C_i) \times ... \times P(x_n|C_i) \tag{2}$$

di mana $1 \leq k \leq n$. Variabel x_k mengacu pada nilai atribut A_k pada baris $\boldsymbol{X}$. Untuk masing-masing atribut, periksa apakah atribut tersebut mutlak atau

bernilai kontinu.

(a) Jika A_k mutlak, maka $P(x_k|C_i)$ merupakan jumlah baris data pada kelas C_i pada D yang memiliki nilai x_k untuk A_k, dibagi dengan $|C_{i,D}|$.

(b) Jika A_k bernilai kontinu, maka penghitungan menggunakan rumus

$$P(x_k|C_i) = g(x_k, _{C_i}, \sigma_{C_i}) \tag{3}$$

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{4}$$

μ_{C_i} merupakan nilai rata-rata dan σ_{C_i} merupakan standar deviasi atribut A_k pada baris data kelas C_i.

5. Untuk memprediksi kelas dari baris $\boldsymbol{X}$, $P(\boldsymbol{X}|C_i)P(C_i)$ dihitung untuk setiap kelas C_i. *Classifier* memprediksi baris $\boldsymbol{X}$ tergolong pada kelas C_i jika dan hanya jika $P(\boldsymbol{X}|C_i)P(C_i) > P(\boldsymbol{X}|C_j)P(C_j)$ di mana $1 \leq j \leq m, j \neq i$. Dengan kata lain, kelas yang diprediksi merupakan kelas C_i yang memiliki nilai $P(\boldsymbol{X}|C_i)P(\boldsymbol{C_i})$ terbesar.

2.4 *Cross Validation*

Cross validation merupakan salah satu teknik untuk memeriksa akurasi sebuah *classifier*, terutama jika jumlah data *training* dan *testing* terbatas. Pada *cross validation*, jumlah *fold* atau partisi data ditentukan, misalnya k, kemudian data dipisah

menjadi k bagian. Misal $k = 3$, maka data dibagi menjadi tiga bagian dengan ukuran sama besar. Masing-masing bagian digunakan secara bergantian untuk *training* dan *testing*. Pada *3-fold cross validation*, dua per tiga data digunakan untuk *training* dan sisanya untuk *testing*. Prosedur dilakukan tiga kali, di mana akurasi dihitung dari rata-rata akurasi *classifier* per *fold* (Witten, Frank, & Hall, 2011). Pada prediksi, tingkat kesalahan dihitung berdasarkan jumlah kesalahan pada k perulangan yang dibagi dengan jumlah data yang ada (Han & Kamber, 2006).

Cara standar untuk memprediksi tingkat kesalahan *classifier* yaitu dengan menggunakan *stratified 10-fold cross validation*. *Stratified* berarti setiap kelas direpresentasikan dengan tepat pada data *training* dan *testing*. Angka sepuluh dianggap merupakan jumlah *fold* yang tepat untuk memperoleh perkiraan *error* terbaik (Witten et al., 2011). *10-fold cross validation* direkomendasikan karena memiliki bias dan variansi yang cukup rendah (Han & Kamber, 2006).

2.5 *Confusion Matrix*

Confusion matrix merupakan alat yang digunakan untuk mengukur seberapa baik akurasi sebuah *classifier* dalam memprediksi data dari kelas yang berbeda. Jika terdapat m kelas, ukuran *confusion matrix* yaitu $m \times m$. Pada *classifier* biner, *confusion matrix* berukuran 2×2 (Gambar 2.1). Akurasi *classifier* dianggap baik jika nilai pada diagonal matriks $CM_{1,1}$ dan $CM_{m,m}$ mendekati 1, serta nilai lainnya mendekati nol (Han & Kamber, 2006).

		Predicted class	
		C_1	C_2
Actual class	C_1	true positives	false negatives
	C_2	false positives	true negatives

Gambar 2.1 *Confusion Matrix* pada *Classifier* Biner (Han & Kamber, 2006)

Hasil pengujian *classifier* dapat dikelompokkan menjadi *true positive, false positive, true negative*, dan *false negative*. *True positive* (TP) merupakan data positif yang diprediksi dengan benar, sedangkan *false positive* (FP) merupakan data negatif yang diprediksi sebagai positif. *True negative* (TN) merupakan data negatif yang diprediksi dengan benar, sedangkan *false negative* (FN) merupakan data positif yang diprediksi sebagai negatif. *True positive rate* dihitung dengan membagi TP dengan jumlah data positif. *False positive rate* dihitung dengan membagi FP dengan jumlah data negatif. Tingkat kesalahan (*error rate*) bernilai 1- tingkat kesuksesan (akurasi) prediksi, yang dihitung dengan rumus berikut (Metz, 1978; Witten et al., 2011).

Accuracy = sensitivity×fraction of positive data+specificity×fraction of negative data

$$= \frac{TP}{TP+FN} \times \frac{TP+FN}{TP+FP+TN+FN} + \frac{TN}{TN+FP} \times \frac{TN+FP}{TP+FP+TN+FN}$$

$$= \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

2.6 F_1-*score*

F_1-*score* (atau *F-measure*) merupakan salah satu ukuran performa sebuah *classifier* berdasarkan nilai *precision* dan *recall* (Witten et al., 2011). *Precision* merupakan ukuran akurasi *classifier* dalam memprediksi data positif, sedangkan *recall* merupakan ukuran seberapa banyak data positif yang diprediksi dengan tepat (Davis & Goadrich, 2006).

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

F_1-*score* menggambarkan *harmonic mean* antara nilai *precision* dan *recall* untuk sebuah *classifier* (Zaki & Jr., 2014). F_1-*score* memiliki rentang nilai 0 hingga 1. Jika nilai mendekati 1, *classifier* semakin bagus (*F1 Score*, n.d.). F_1-*score* dihitung dengan rumus berikut (Witten et al., 2011).

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{8}$$

2.7 *Matthews Correlation Coefficient*

Matthews Correlation Coefficient (MCC) merupakan ukuran kualitas sebuah *classifier* biner, yang diperkenalkan oleh Brian W. Matthews pada 1975. Nilai koe-

fisien korelasi *Matthews* dapat dihitung dengan rumus berikut (Matthews, 1975).

$$N = TN + TP + FN + FP \tag{9}$$

$$S = \frac{TP + FN}{N} \tag{10}$$

$$P = \frac{TP + FP}{N} \tag{11}$$

$$MCC = \frac{TP/N - S \times P}{\sqrt{PS(1-S)(1-P)}} \tag{12}$$

N merupakan total data, S merupakan persentase data positif dari keseluruhan data, sedangkan P merupakan persentase prediksi positif oleh *classifier*. Nilai koefisien korelasi *Matthews* juga dapat dihitung langsung berdasarkan hasil *confusion matrix* dengan rumus berikut (Raschka, 2014).

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{13}$$

Jika nilai koefisien korelasi mendekati 1, berarti prediksi *classifier* sempurna. Jika nilai -1, berarti hasil prediksi *classifier* tidak sesuai dengan seharusnya. Jika nilai nol, berarti *classifier* memprediksi dengan acak (Lund, Nielsen, Lundegaard, Kesmir, & Brunak, 2010).

2.8 *Majority Rule*

Majority rule merupakan salah satu dari beberapa aturan pemungutan suara. Pada aturan ini, hasil akhir antara *x* atau *y* dipilih dengan melihat perbandingan jumlah suara antara *x* dan *y*. *x* dinyatakan sebagai pemenang apabila memiliki jumlah lebih besar dari *y*. *Majority rule* bekerja cukup baik pada banyak *domain* dan termasuk aturan yang paling kuat (Dasgupta & Maskin, 2008). Aturan pemungutan suara ini sering digunakan pada lembaga pengambil keputusan, termasuk lembaga legislatif pada negara demokratis (*Majority Rule*, n.d.).

2.9 *Tweepy*

Tweepy merupakan salah satu *library Python* yang menggunakan *Twitter Streaming API* (Jahanbakhsh & Moon, 2014). *Streaming API* memungkinkan akses langsung terhadap aliran data *tweet* secara global (*The Streaming APIs*, n.d.). Dengan menggunakan *Streaming API*, *tweet* terbaru mengenai suatu *query* dapat diperoleh (*Documentation*, n.d.).

Penggunaan *Twitter API* membutuhkan otentikasi berupa *token* yang diperoleh dengan mendaftarkan aplikasi ke situs *Twitter Developer*. Ada empat *token* yang dibutuhkan, yaitu *consumer key (API key), consumer secret (API secret), access token*, dan *access token secret* (Efendi, Erwin, & Eng, 2015).

2.10 *Scikit-Learn*

Scikit-learn merupakan *library open source* untuk *machine learning* pada bahasa pemrograman *Python*. Pada mulanya, *scikit-learn* merupakan proyek bernama *scikits.learn* oleh David Cournapeau. Nama *scikit* berasal dari *Scipy Toolkit*, yaitu tambahan untuk *library SciPy* yang dibuat secara terpisah. *Scikit-learn* menyediakan algoritma untuk klasifikasi, regresi, dan *clustering* seperti *Support Vector Machine* (SVM), *random forests, k-means*, dan *gradient boosting* (*scikit-learn*, n.d.).

Scikit-learn dapat diakses oleh siapa saja dan dapat digunakan kembali untuk berbagai konteks. *Scikit-learn* dibangun berdasarkan *NumPy, SciPy*, dan *matplotlib*. Penggunaan *scikit-learn* sederhana dan efisien untuk data mining maupun analisis (*scikit-learn: Machine Learning in Python*, n.d.).

2.11 *Natural Language Toolkit* (NLTK)

Natural Language Toolkit (NLTK) dibuat pada tahun 2001 sebagai bagian dari mata kuliah komputasi linguistik pada *Department of Computer and Information Science* di *University of Pennsylvania*. Sejak saat itu, NLTK telah dikembangkan oleh sejumlah kontributor dan banyak digunakan di sejumlah universitas serta menjadi dasar dalam proyek penelitian. NLTK memiliki sejumlah modul penting, dapat dilihat pada Tabel 2.1 (Bird, Klen, & Loper, 2009). Penelitian ini akan menggunakan modul *nltk.classify* untuk melakukan klasifikasi dengan *Naive Bayes Classifier*.

Tabel 2.1 Modul NLTK

Language Processing Task	*NLTK Modules*	*Functionality*
Accessing corpora	*nltk.corpus*	*Standardized interfaces to corpora and lexicons*
String processing	*nltk.tokenize, nltk.stem*	*Tokenizers, sentence tokenizers, stemmers*
Collocation discovery	*nltk.collocations*	*t-test, chi-squared, point-wise mutual information*
Part-of-speech-tagging	*nltk.tag*	*n-gram, backoff, Brill, HMM, TnT*
Classification	*nltk.classify, nltk.cluster*	*Decision tree, maximum entropy, naive Bayes, EM, k-means*
Chunking	*nltk.chunk*	*Regular expressions, n-gram, named entity*
Parsing	*nltk.parse*	*Chart, feature-based, unification, probabilistic, dependency*
Semantic interpretation	*nltk.sem, nltk.inference*	*Lambda calculus, first-order logic, model-checking*
Evaluation metrics	*nltk.metrics*	*Frequency distributions, smoothed probability distributions*
Applications	*nltk.app, nltk.chat*	*Graphical concordance, parsers, WordNet browser, chatbots*
Linguistic fieldwork	*nltk.toolbox*	*Manipulate data in SIL Toolbox format*

2.12 Penelitian Terdahulu

2.12.1 *From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series*

O'Connor et al. (2010) melakukan penelitian terhadap sentimen yang terkandung dalam teks mengenai opini politik selama tahun 2008-2009. Penelitian ini

bertujuan untuk mengetahui apakah sentimen yang terkandung pada opini publik sesuai dengan hasil survei dan *polling* yang dilakukan.

Data yang dikumpulkan dalam penelitian ini berupa *tweet*, survei opini publik dan *polling*. Satu juta *tweet* dikumpulkan selama tahun 2008-2009 dengan menggunakan *Twitter API*, di mana sebagian besar pengguna berasal dari Amerika Serikat. Survei opini publik yang dikumpulkan berasal dari beberapa sumber, yaitu *Consumer Confidence Index* dari *Consumer Board*, *Index of Consumer Sentiment* (ICS) dari *Reuters/University of Michigan Surveys of Consumers*, *Economic Confidence Index* oleh *Gallup Organization*, dan *polling* dari *Pollster.com*. *Consumer confidence* merupakan tingkat keyakinan masyarakat mengenai kondisi perekonomian.

Secara garis besar, penelitian ini dapat dibagi menjadi dua bagian utama, yaitu pengumpulan pesan dan estimasi opini. Dalam pengumpulan pesan (berupa *tweet*), ditentukan sejumlah kata kunci untuk masing-masing topik, misalnya *Obama* dan *McCain* untuk topik mengenai pemilihan umum dan *economy, job, job*s untuk topik mengenai *consumer confidence*. Jumlah pesan yang terkumpul selama hari kerja cenderung lebih banyak daripada akhir pekan.

Nilai sentimen pada pesan diperoleh dengan menghitung jumlah pesan positif dan negatif. Kata-kata positif dan negatif didefinisikan oleh *subjectivity lexicon* dari *OpinionFinder*, yang berisi sekitar 1.600 kata positif dan 1.200 kata negatif. Sebuah pesan dikategorikan positif jika mengandung kata positif dan dikategorikan negatif jika mengandung kata negatif. Pada penerapannya, terdapat sentimen pesan yang

salah terdeteksi. Selain itu, banyak pesan pada *Twitter* yang ditulis dengan bahasa informal.

Dalam menghitung rasio sentimen, O'Connor et al. (2010) menggunakan *moving average* untuk memperoleh hasil yang lebih *smooth*. *Smoothing* menyebabkan rasio sentimen lebih lambat merespon perubahan dan jika berlebihan dapat menyebabkan perubahan sentimen menjadi tidak terlihat.

Dengan *15-day-smoothing*, rasio sentimen berkorelasi dengan survei *Gallup* dengan nilai $r = 73,1\%$. Berdasarkan analisis dengan *one variable linear least-squares model* dengan parameter *L*, *Gallup* terbukti merupakan indikator yang lebih baik daripada ICS. Namun, *Gallup* berkorelasi buruk dengan rasio sentimen kata *job* dan *economy*, yaitu hanya mencapai 10% dan 7% (dengan $k = 15, L = 0$).

Model tersebut kemudian dihitung dengan menggunakan *rolling forecast* untuk melihat prediksi hasil survei berdasarkan sentimen pada teks. Pada 2008 dan awal 2009, sentimen pada teks memprediksi *consumer confidence* dengan buruk. Namun, mulai pertengahan 2009, prediksi berdasarkan sentimen pada teks dapat memprediksi dengan lebih baik. Hal ini dapat disebabkan oleh perubahan jumlah pengguna *Twitter* serta cara pengguna *Twitter* dalam memanfaatkan media sosial.

Analisis sentimen terhadap kinerja Obama sebagai presiden menurun secara stabil sejak awal masa jabatannya, dengan nilai $r = 72,5\%$ pada $k = 15$. Namun, rasio sentimen pada *tweet* tidak berkorelasi dengan hasil *polling* pemilihan umum. Hal ini disebabkan karena *polling* pemilihan umum memungkinkan masya-

rakat memilih ”Obama”, ”McCain”, belum memutuskan, tidak berencana memilih, atau kandidat independen. Frekuensi jumlah pesan tidak memiliki hubungan langsung dengan opini publik.

Penelitian ini membuktikan bahwa *polling* dapat dilengkapi dengan data dari media sosial. Namun, penelitian ini membutuhkan *lexicon* yang lebih sesuai dengan konteks. Penggunaan *retweet* dan kalimat *headline* berita dipertimbangkan dapat mempengaruhi sentimen.

2.12.2 *Predicting US Primary Elections with Twitter*

Penelitian ini bertujuan untuk memprediksi opini publik dan hasil pemilihan kandidat Presiden Amerika Serikat 2012 dari Partai Republik. Data dikumpulkan dengan menggunakan *Twitter Streaming API* dan *Twitter API endpoint*, masing-masing sebanyak sekitar 300 juta *tweet* dan sekitar sepuluh juta *tweet*. Pengumpulan data dilakukan sejak September 2011 hingga Februari 2012. Data disimpan pada *database* MongoDB.

Pada mulanya, prediksi dilakukan dengan cara tradisional, yaitu hanya menggunakan volume *tweet* yang ada dan analisis sentimen dengan metode serupa dengan O’Connor et al. (2010). Hasil analisis sentimen dan volume *tweet* menunjukkan hasil yang serupa. Eksperimen tersebut membuktikan bahwa prediksi tidak dapat dilakukan dengan tepat jika hanya menggunakan volume *tweet* atau *tweet* bermakna positif.

Lokasi pengguna *Twitter* dianggap penting untuk memperoleh sentimen di setiap negara bagian. Namun, hanya 1,5% *tweet* yang mengandung informasi lokasi sehingga perlu ada mekanisme untuk mengetahui lokasi pengguna *Twitter*. Mekanisme ini disebut *Location Recovery Algorithm*, dan bekerja dengan menggunakan informasi lokasi pada *field Location* yang tertera pada profil pengguna.

Algoritma tersebut hanya mengidentifikasi lokasi pengguna berdasarkan teks dan *gazetteer* sehingga prediksi berdasarkan lokasi yang paling sesuai tidak menyelesaikan masalah. Oleh karena itu, Shi et al. (2012) merancang algoritma yang menghasilkan *confidence score* terhadap tingkat keyakinan sistem terhadap kebenaran hasil prediksi. Penelitian ini juga mengembangkan model regresi linear dengan adanya *lasso (Least Absolute Shrinkage and Selection Operator)*. Koefisien variabel yang tidak relevan akan dianggap nol.

$$L = \sum_{i} (y_i - \sum_{p} \beta_p x_{ip})^2 + \lambda \sum_{p} \| \beta_p \| \tag{14}$$

Pada persamaan tersebut, x_{ip} menunjukkan *feature* ke-p pada data ke-i, y_i menunjukkan nilai target pada data, dan β_p menunjukkan koefisien regresi pada *feature* ke-p. λ merupakan parameter penyesuaian kompleksitas yang mengontrol jumlah penyusutan. Semakin besar nilai λ, semakin besar penyusutan. Oleh karena itu, λ disesuaikan dengan perkiraan *error* pada prediksi. $\Sigma_p \beta_p$ mengarahkan ke persebaran yang sempit pada *feature space*, berarti koefisien regresi pada *feature* yang tidak relevan atau berulang akan disusutkan menjadi nol. Model regresi ini efektif

ketika ada banyak *feature* yang tidak relevan dan hanya ada sedikit data *training*. Penelitian ini hanya memiliki sekitar 100 data *training*, sehingga penggunaan *Lasso regression* cukup tepat.

Dengan mengidentifikasi lokasi geografis pengguna, sentimen *tweet* per negara bagian dapat diperoleh. Namun, *RealClearPolitics.com* (RCP) tidak menyediakan nilai per negara bagian, hanya ada tiga negara bagian. Sembilan puluh persen data digunakan untuk *training* dan sisanya untuk *testing*. Pada beberapa negara bagian, hasil prediksi tidak banyak berubah dari minggu ke minggu. Namun, hasil pemilihan di South Carolina berubah drastis, menggambarkan adanya perubahan opini publik, yang mungkin disebabkan isu pengembalian pajak oleh Mitt Romney.

Analisis sentimen juga dilakukan secara nasional. Sembilan puluh persen data digunakan untuk *training* dan sisanya digunakan untuk *testing*. Oleh karena perbedaan pendukung, model dibuat secara terpisah untuk masing-masing kandidat. Prediksi untuk Mitt Romney dan Newt Gingrich cukup mendekati hasil pada RCP, sedangkan dua kandidat lainnya tidak sesuai. Hal ini dapat disebabkan oleh data *training* yang cukup datar sehingga model *Lasso regression* tidak dapat menentukan *feature* yang tepat.

Penelitian ini membuktikan bahwa *polling* dapat dilengkapi atau digantikan dengan analisis data dari media sosial. Penggunaan volume *tweet* dan analisis sentimen dianggap tidak cukup dan diperlukan algoritma yang lebih mutakhir agar hasil prediksi lebih berhasil.

2.12.3 *The Predictive Power of Social Media: On the Predictability of U.S. Presidential Elections using Twitter*

Penelitian ini dilatarbelakangi oleh potensi media sosial, khususnya *Twitter*, dalam menyimpan informasi yang dapat dianalisis. Pemilihan Presiden Amerika Serikat 2012 dipilih sebagai kejadian yang diprediksi karena hasilnya terukur dan memiliki dampak besar pada bidang sosial-ekonomi, serta memiliki sumber data yang besar pada *Twitter*. Tujuan penelitian ini yaitu untuk mengetahui apakah analisis sentimen data pada *Twitter* dapat digunakan untuk memprediksi Pemilihan Presiden Amerika Serikat 2012, dan mengetahui apakah topik yang dibicarakan di *Twitter* sesuai dengan topik diskusi secara *offline*.

Pengumpulan data dilakukan sejak 29 September hingga 16 November 2012. Distribusi data per kata kunci (misal: *Obama*) dihitung per hari. Untuk setiap *tweet*, polaritas $P(l|tw)$ dihitung, di mana l merupakan polaritas *tweet* tw dan dapat bernilai salah satu dari $L = \{0, -1, +1\}$. *Tweet* diberikan *label* l yang memiliki kemungkinan polaritas terbesar. Analisis sentimen dilakukan dengan menghitung kemungkinan polaritas dari sebagian *tweet* melalui *random sampling* per harinya. Kemudian, analisis sentimen dilakukan berdasarkan lokasi geografis dengan menggunakan *tweet* yang memiliki informasi lokasi. Probabilitas distribusi topik dihitung dengan $P(w_i|\text{topic index} = k)$ di mana $w_i \in V$ merupakan kata dari kosa kata *tweet* dan k merupakan indeks topik.

Tweet dikumpulkan dengan menggunakan *Twitter Streaming API* pada *li-*

brary Tweepy. Hanya *tweet* yang mengandung kata kunci politik yang diambil, seperti *barack obama, mitt romney, us election*, dan *joe biden*. Jahanbakhsh & Moon (2014) berhasil mengumpulkan sekitar 39 juta *tweet* dan menyimpannya ke *database* MySQL. Sebanyak 140.000 *tweet* yang memiliki informasi geografis disimpan pada struktur data *k-d tree*.

Penelitian ini mengembangkan mesin *machine learning and language processing* (ML-NLP) yang memungkinkan analisis teks pada *tweet* berjumlah besar. Mesin ini terdiri dari 4 komponen utama, yaitu (1) komponen statistik, (2) analisis teks, (3) *Naive Bayes classifier* untuk analisis sentimen, dan (4) algoritma *Latent Dirichlet Allocation* (LDA) untuk *topic modeling*.

Analisis statistik terdiri dari distribusi frekuensi *tweet* per hari, distribusi *mention tweet* per hari, dan distribusi *hashtag*. *Tweet* terbanyak ada pada 3, 11, 16, 22 Oktober serta 6 November 2012. Obama memimpin pada jumlah *mention*, sedangkan Romney hanya sempat memimpin selama satu hingga dua hari pasca debat presiden. Melalui distribusi *hashtag*, terlihat bahwa *#obama* merupakan *hashtag* terpopuler di *tweet* yang terkumpul.

Analisis sentimen dilakukan dengan menggunakan *Naive Bayes Classifier* karena dapat memproduksi hasil berkualitas tinggi dan penggunaannya sederhana. Setiap isi *tweet* dianggap sebagai *bag of words*, dan memiliki *label* 0 (netral), -1 (negatif), atau 1 (positif). *Label* dihitung dengan rumus $label_{NB} = \arg\max_{l_j \in L} P(label =$

$l_j \mid w_i$). Probabilitas $P(label = l_j \mid w_i)$ dihitung dengan

$$P(label = l_j \mid w_i) = \frac{count(w_i, l_j) + 1}{count(l_j) + \mid V \mid} \quad (15)$$

di mana $count(w_i, l_j)$ merupakan jumlah munculnya kata w_i dengan *label* l_j pada data *training*, $count(l_j)$ merupakan jumlah munculnya *label* l_j pada data *training*, dan $\mid V \mid$ merupakan jumlah kata pada data *training*.

Data *training* yang digunakan sebanyak 989 *tweet* yang dipilih secara acak dan diberi *label* secara manual. *Tweet* yang digunakan untuk *training* hanyalah *tweet* yang mengandung salah satu nama kandidat. Sebelum dilatih, data melalui tahap *preprocessing* terlebih dahulu, meliputi pembersihan dari URL, *mention*, *re-tweets* (RT), *hashtags*, angka, dan *stop words*; tokenisasi; penambahan kata *not_* apabila terdapat kata negasi.

Analisis sentimen dilakukan dengan mengambil 10.000 *tweet* secara acak. Untuk meminimalisir *error*, analisis dilakukan pada *tweet* yang mengandung satu nama kandidat. Jumlah *tweet* negatif lebih banyak daripada *tweet* positif. Hasil analisis dibandingkan dengan *polling* dari *Huffington Post*, di mana sebagian besar hasil *polling* sesuai dengan analisis sentimen pada *Twitter*, dengan beberapa *latency*. Pada analisis berdasarkan lokasi geografis, 76% negara bagian berhasil diprediksi dengan tepat. Hal ini dapat disebabkan oleh lebih banyaknya penggunaan *Twitter* pada daerah dengan populasi tinggi.

Topic modeling dilakukan dengan algoritma LDA menggunakan metode *Gibbs sampling*. *Tweet* yang dianalisis diperoleh dari tiga debat presiden, untuk mengetahui topik yang didiskusikan selama debat berlangsung. Jumlah topik diatur menjadi lima ($K = 5$) dan *Gibbs sampling* dilakukan sebanyak 1.000 iterasi menggunakan mesin ML-NLP. Untuk masing-masing topik, 15 kata teratas diambil.

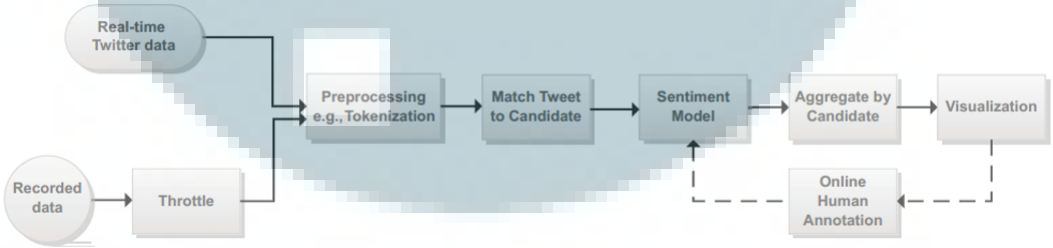
Melalui penelitian ini, terbukti bahwa teknik *machine learning* dapat digunakan untuk menganalisis data dari media sosial. Hasil analisis pada penelitian ini cukup kredibel, ditunjukkan dengan memimpinnya Obama di *Twitter*, sesuai dengan hasil *polling*. Penelitian ini juga mendemonstrasikan LDA sebagai algoritma *unsupervised* yang cukup tangguh. Namun, penelitian ini tidak menggunakan *tweet* yang mengandung lebih dari satu nama kandidat.

2.12.4 *A System for Real-Time Twitter Sentiment Analysis of 2012 U.S. Presidential Election Cycle*

Penelitian ini bertujuan untuk membuat sistem yang menampilkan analisis sentimen terhadap kandidat Presiden Amerika Serikat 2012 secara *real-time*. Maraknya penggunaan *Twitter* sebagai media masyarakat dalam menyampaikan pendapat, termasuk mengenai masalah politik, serta meningkatnya jumlah *tweet* ketika kampanye atau debat sedang berlangsung mendorong dilakukannya penelitian ini. Sistem ini dibangun menggunakan *IBM InfoSphere Streams platform* karena membutuhkan skalabilitas tinggi mengingat jumlah *tweet* cukup tinggi ketika ada

kejadian tertentu (misal: debat) dan rendah ketika tidak ada kejadian khusus.

Sentimen dianalisis dari *tweet* yang dikumpulkan sejak 12 Oktober 2012 dan mencapai lebih dari 36 juta *tweet*. *Tweet* dikumpulkan menggunakan *Gnip Power Track* karena keterbatasan jumlah data yang dapat diperoleh menggunakan *Twitter API*. *Tweet* yang dikumpulkan merupakan *tweet* yang berkaitan dengan Barack Obama (Presiden Amerika Serikat saat itu, sekaligus kandidat Partai Demokrat) serta sembilan orang kandidat dari Partai Republik. Dalam mengumpulkan *tweet* yang relevan, Wang et al. (2012) menentukan sekitar 200 aturan, misalnya kata kunci *@MittRomney, @MittNews*, *#romney* untuk memperoleh *tweet* mengenai Mitt Romney.



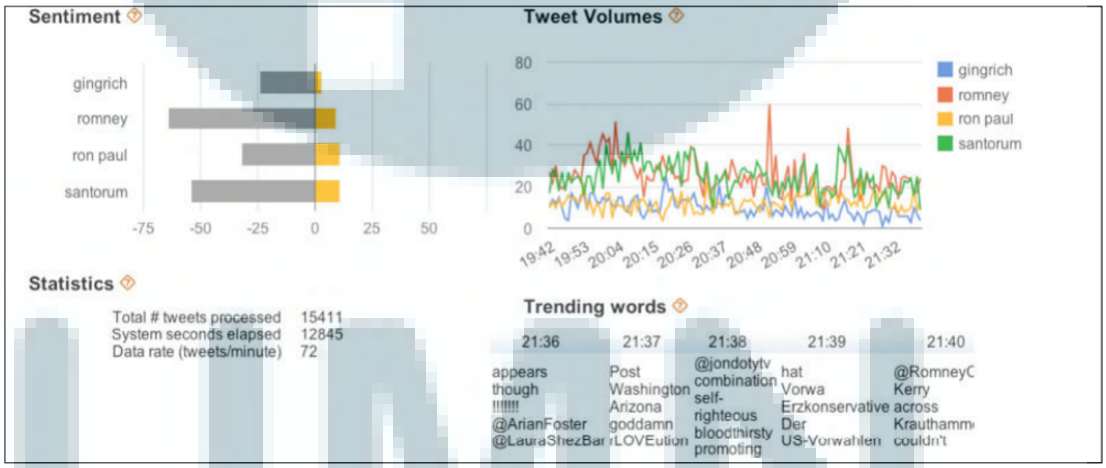
Gambar 2.2 Arsitektur Sistem (Wang et al., 2012)

Tweet yang sudah dikumpulkan melalui tahap *preprocessing*. Pada tahap ini, teks dalam *tweet* di-*tokenize* menggunakan *tokenizer* buatan Christopher Potts. *Tokenizer* tersebut memisahkan URL, *emoticons*, nomor telepon, *tag* HTML, *mention*, *hashtag*, angka pecahan maupun desimal, serta repetisi simbol maupun karakter *Unicode*.

Dalam menentukan sentimen, Wang et al. (2012) mengumpulkan sejumlah

orang (*Turkers*) melalui *Amazon Mechanical Turk* (AMT) untuk menandai *tweet* yang sudah terkumpul. *Tweet* ditampilkan pada sebuah antarmuka, di mana *Turkers* harus mengisi usia, jenis kelamin, dan pandangan politiknya. Data *training* berjumlah sekitar 17.000 *tweet* dengan komposisi 16% positif, 56% negatif, 18% netral, dan 10% tidak yakin. Ada sekitar 800 orang *Turkers* yang terlibat dalam penelitian ini.

Data *training* dilatih menggunakan *Naive Bayes Classifier* dengan *unigram features*. *Classifier* memiliki akurasi 59% dengan 4 kelas, yaitu positif, negatif, netral, dan tidak yakin. Oleh karena sistem dibuat secara *real-time*, jumlah *tweet* yang masuk diperbarui setiap menit. Sentimen yang masuk divisualisasikan setiap 5 menit.



Gambar 2.3 Tampilan *Dashboard* (Wang et al., 2012)

Sentimen divisualisasikan pada *dashboard* berbasis AJAX-HTML. *Dashboard* tersebut mengambil data dari *server* dan di-*refresh* secara otomatis setiap 30 detik. Selain sentimen, *dashboard* ini menampilkan jumlah *tweet* yang diproses serta

kata yang banyak digunakan (*trending words*). *Trending words* dihitung menggunakan TF-IDF, di mana *trending words* merupakan kata yang memiliki bobot TF-IDF tertinggi pada menit tersebut.

Sistem ini dapat mengatasi adanya kecenderungan negatif pada *tweet*. Selain itu, arsitektur dan metode sistem tergolong generik sehingga dapat diadopsi untuk sistem lain.

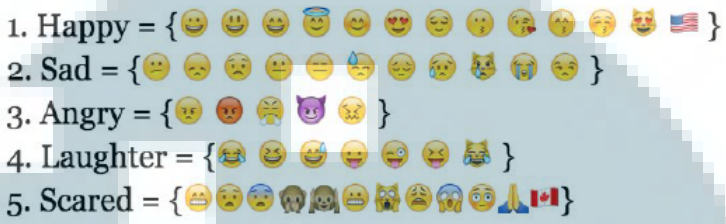
2.12.5 *Analyzing Twitter Sentiment of the 2016 Presidential Candidates*

Chin et al. (2015) melakukan analisis sentimen pada *tweet* mengenai beberapa kandidat utama Pemilihan Presiden Amerika Serikat 2016, yaitu Donald Trump, Hillary Clinton, Ben Carson, dan Bernie Sanders. Sentimen mengenai kandidat-kandidat tersebut berfluktuasi seiring dengan adanya debat, wawancara, respon terhadap kejadian global, maupun isu lainnya. Oleh karena itu, penelitian ini bertujuan untuk memprediksi sentimen publik terhadap masing-masing kandidat.

Tweet dikumpulkan menggunakan *Twitter API* dengan sejumlah kata kunci, antara lain *political candidates*, *politics*, dan nama lengkap kandidat. Dari sekitar 30.000 *tweet* yang terkumpul, diperoleh hampir 3.000 *tweet* yang mengandung *emoji* yang sudah ditentukan. Sebanyak 2.000 *tweet* digunakan sebagai data *training* dan sisanya untuk *testing*. *Label tweet* ditentukan dengan mengambil *emoji* pertama pada *tweet*.

Klasifikasi sentimen dibagi menjadi lima kelas: (1) senang (*happy*), (2) se-

dih (*sad*), (3) marah (*angry*), (4) tertawa (*laughter*), dan (5) takut (*scared*). Sebuah *tweet* akan dikategorikan dalam kelas tersebut apabila mengandung *emoji* yang sudah didaftar sebelumnya (Gambar 2.4).



Gambar 2.4 Daftar *Emoji* pada Masing-masing Kelas (Chin et al., 2015)

Tahap *preprocessing* dilakukan dengan mengubah teks menjadi huruf kecil; menyingkirkan *hashtag*, URL, *username*, dan spasi berlebih; menyingkirkan kata-kata umum yang tidak mengandung sentimen; dan mengganti huruf sama yang muncul lebih dari dua kali menjadi dua huruf tersebut (contoh: *soooo funnnyyy* menjadi *soo funnyy*).

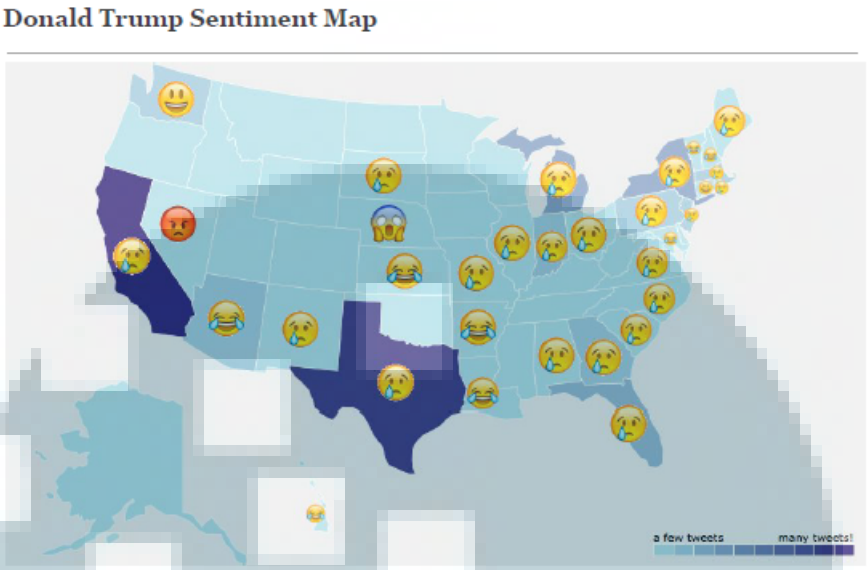
Klasifikasi dilakukan dengan *multi-class classification* pada tiga algoritma: (1) *Support Vector Machine* (SVM), (2) *k-nearest neighbors* (KNN), dan (3) *Naive Bayes*. Hasil klasifikasi menunjukkan akurasi tertinggi pada model SVM (49,22%). *Naive Bayes* memiliki akurasi sedikit lebih rendah (49,02%), sedangkan KNN hanya dapat mencapai akurasi tertinggi sebesar 45,99% dengan nilai $k = 6$. Secara keseluruhan, *Naive Bayes* merupakan algoritma yang paling efisien dalam hal waktu, sedangkan KNN bekerja paling lambat dan kurang akurat.

Chin et al. (2015) mencoba menganalisis penyebab *error* pada model dengan melatih model SVM baru dan membandingkan akurasinya dengan model SVM

awal. Ketika dilatih dengan data yang tidak mengandung tanda baca, akurasi model hanya meningkat sedikit. Pada percobaan berikutnya, model dilatih dengan data *training* yang mengandung URL dan *username*. Model tersebut memiliki akurasi yang lebih tinggi. Pada percobaan terakhir, data *training* tidak dibersihkan dari kata-kata umum yang tidak bersentimen. Model yang dilatih dengan data tersebut hanya memiliki akurasi sedikit lebih tinggi dari model awal.

Untuk menganalisis sentimen publik berdasarkan lokasi, Chin et al. (2015) memanfaatkan *metadata tweet*, yaitu *latitude*, *longitude*, serta informasi yang dimasukkan pengguna (nama universitas, kode area, dsb.). Sentimen *tweet* dipetakan menggunakan *heat map*. Apabila terdapat lebih dari satu *emoji* pada suatu negara bagian, hanya satu *emoji* dominan yang akan digunakan dalam peta. Namun demikian, skala antara populasi negara bagian dengan jumlah *tweet* tidak disesuaikan sehingga negara bagian dengan populasi sedikit cenderung memiliki lebih sedikit jumlah *tweet*.

Apabila *retweet* tidak dihitung, *laughter* merupakan *emoji* yang paling dominan untuk setiap kandidat. Pemetaan sentimen pada Donald Trump menunjukkan sentimen yang tepat waktu, dengan banyaknya *emoji sad* pada momen penembakan di *San Bernardino* (Gambar 2.5). Hal ini dapat terjadi karena adanya pernyataan atau tindakan Trump yang tidak tepat dalam merespon peristiwa tersebut. Selain itu, kesalahan Ben Carson dalam mengucapkan ”Hamas” menjadi ”hummus” pada forum di D.C. menyebabkan banyaknya *emoji laughter* pada dirinya.



Gambar 2.5 Contoh Pemetaan Sentimen (Chin et al., 2015)

Meskipun berhasil melakukan pemetaan sentimen pada kandidat, prediksi hanya menggunakan *tweet* yang mengandung *emoji*. *Tweet* tersebut jumlahnya sedikit jika dibandingkan dengan keseluruhan *tweet* yang membahas Pemilihan Presiden Amerika Serikat 2016 dan umumnya *emoji* digunakan oleh kaum muda. Akibatnya, analisis yang dilakukan cenderung *bias*. Selain itu, pemetaan sentimen dilakukan berdasarkan *metadata tweet* dan tidak semua pengguna *Twitter* menuliskan data lengkap sehingga sejumlah *tweet* tidak dapat dipetakan.