BAB II
LANDASAN TEORI

2.1 Twitter

Twitter adalah sebuah situs web yang dimiliki dan dioperasikan oleh Twitter Inc., yang menawarkan jaringan sosial berupa mikroblog sehingga memungkinkan penggunanya untuk mengirim dan membaca pesan *Tweets* (Twitter, 2013). Mikroblog adalah salah satu jenis alat komunikasi online dimana pengguna dapat memperbarui status tentang mereka yang sedang memikirkan dan melakukan sesuatu, apa pendapat mereka tentang suatu objek atau fenomena tertentu. *Tweets* adalah teks tulisan hingga 140 karakter yang ditampilkan pada halaman profil pengguna. *Tweets* bisa dilihat secara publik, namun pengirim dapat membatasi pengiriman pesan ke daftar teman-teman mereka saja. Pengguna dapat melihat *Tweets* pengguna lain yang dikenal dengan sebutan pengikut (*follower)*.

2.1.1 Twitter API

Application Programming Interface (API) merupakan fungsi-fungsi untuk menggantikan bahasa yang digunakan dalam *system calls* dengan bahasa yang lebih terstruktur dan mudah dimengerti oleh programmer. Fungsi yang dibuat dengan menggunakan API tersebut kemudian akan memanggil *system calls* sesuai dengan sistem operasinya. Tidak tertutup kemungkinan nama dari *system calls* sama dengan nama di API (Arifidin, 2016).

Pada awalnya perusahaan *Summize* yang menyediakan fasilitas mencari data di Twitter. Kemudian perusahaan *Summize* ini diakuisisi dan diganti merek menjadi *Twitter Search* sehingga *Search API* terpisah sebagai entitas sendiri. API Twitter terdiri dari 3 (tiga) bagian yaitu (Arifidin, 2016) :

a. *Search API*

Search API dirancang untuk memudahkan user dalam mengelola *query search* di konten Twitter. User dapat menggunakannya untuk mencari *tweet* berdasarkan *keyword* khusus atau mencari *tweet* lebih spesifik berdasarkan username Twitter. *Search API* juga menyediakan akses pada data *Trending Topic*.

b. *Representational State Transfer (REST) API*

REST API memperbolehkan *developer* untuk mengakses inti dari Twitter seperti *timeline, status update* dan informasi user. *REST API* digunakan dalam membangun sebuah aplikasi Twitter yang kompleks yang memerlukan inti dari Twitter.

c. *Streaming API*

Streaming API digunakan *developer* untuk kebutuhan yang lebih intensif seperti melakukan penelitian dan analisis data. *Streaming API* dapat menghasilkan aplikasi yang dapat mengetahui statistik *status update*, *follower* dan lain sebagainya.

2.1.2 Struktur Data Twitter

Pesan Twitter memiliki banyak ragam struktur data. (Go, dkk. 2009) menjabarkan karekteristik Twitter sebagai berikut :

1. Pada pesan Twitter, setiap *tweet* hanya berisa panjang maksimal 140 karakter. (Go, dkk. 2009) mencoba menghitung nilai rata-rata panjang setiap *tweet* dimana diketahui rata-rata *tweet* adalah 14 kata atau 78 karakter.
2. Data Twitter dapat bersumber dari beberapa tempat. Dengan Twitter API data dengan mudah didapat.
3. Pengguna Twitter dapat dengan mudah menggunakan media apapun untuk menulis dan mengirimkan pesan mereka, termasuk penggunaan media ponsel. Kemunculan kesalahan penulisan ataupun penggunaan bahasa slang jauh lebih tinggi.

4. Terdapat ragam topik didalamnya. Setiap pengguna dapat menuliskan topik apapun pada Twitter.

Di samping itu (Davidov, dkk. 2010) menyimpulkan bahwa sebuah *tweet* biasanya mengandung alamat URL, alamat pengguna Twitter yang disebut *username* (@+username), atau konten tag yang disebut *hashtag* (#), dan *emoticon*. *Emoticon* adalah ekspresi wajah yang diwakili dengan karakter tertentu hal ini untuk menggambarkan suasana hati atau emosi pengguna. Pengguna biasanya menggunakan *hashtag* (#) untuk menandai atau menentukan topik tertentu (Agarwal, dkk. 2011).

Penggunaan *hashtag* dan *emoticon* dianggap juga dapat tidak mewakili dari sentimen dalam sebuah *tweet* (Go, dkk. 2009). Hal ini bila pada satu kalimat *tweet* mengandung dua emosi.

2.2 Text Mining

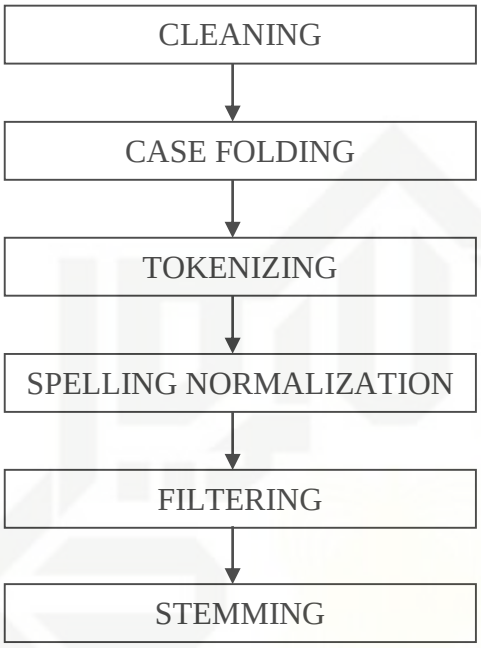
Text mining merupakan teknik yang digunakan untuk menangani masalah klasifikasi, *clustering*, *information extraction* dan *information retrieval* (Berry & Kogan, 2010). *Text mining* berfungsi untuk menemukan kembali informasi yang tersirat dan berasal dari sumber data *text* yang berbeda-beda. Pada dasarnya proses kerja *text mining* mengadopsi dari *data mining*, namun perbedaannya adalah pola yang digunakan oleh *text mining* diambil dari sekumpulan bahasa alami yang tidak terstruktur. Sedangkan *data mining* pola yang diambil dari *database* yang terstruktur.

Secara umum, *text mining* memiliki dua tahap, yaitu *text preprocessing* dan *feature selection* (Feldman & Sanger, 2007). Berikut penjelasan dari tahapan-tahapan tersebut :

2.2.1 *Text Preprocessing*

Text processing merupakan tahapan pertama terhadap *text* untuk mempersiapkan *text* menjadi data yang akan diolah (Feldman & Sanger, 2007) yang

dikutip oleh (Indranandita, Amelia, & dkk, 2008). Tahapan yang dilakukan dari *text preprocessing* dapat dilihat pada gambar 2.1



Gambar 2.1 Proses *text preprocessing*

Tahapan *preprocessing* dimulai dari proses *cleaning*, yaitu proses untuk membersihkan *tweet* dari kata-kata yang tidak diperlukan untuk mengurangi *noise*. Kata atau karakter yang akan dihilangkan adalah karakter atau simbol, HTML, *hashtag* (#), *username* atau *mention* (@username), link url (http://situs.com), *emoticon*, dan RT (tanda re*tweet*).

Setelah proses *cleaning* dilakukan, tahapan selanjutnya ialah *case folding*. Dimana proses ini melakukan penyeragaman bentuk huruf dengan mengubah semua huruf menjadi huruf besar atau huruf kecil, kemudian hanya menggunakan huruf a sampai z. Pada proses *case folding* ini juga akan menghilangkan tanda baca dan angka.

Kemudian dilakukan proses *tokenizing*. Pada proses ini *tweet* atau kalimat akan dipecah menjadi sebuah kata dari sekumpulan data, dengan memisahkan kata tersebut dan menentukan struktur sintaksis setiap kata tersebut.

Selanjutnya dilakukan proses spelling *normalization*. Tahapan ini adalah tahapan yang mengidentifikasi penulisan kata berlebihan dan kata silang kemudian diganti dengan kata kamus KBBI (Rosdiansyah, 2014). Setiap kata yang dijumpai dan penggunaan hurufnya berlebihan dan tidak baku akan diubah. Algoritma normalisasi yang akan dilakukan pada penelitian ini sebagai berikut :

1. Cari kata yang di normalisasi dalam kamus. Setelah ditemukan kata tersebut di asumsikan adalah *root word*, dan algoritma berhenti.
2. Jika tidak ditemukan, hapus huruf berlebihan dimulai dari setiap huruf pada kata, kemudiaan periksa huruf awal kata tersebut, kemudian *recording*. Periksa huruf selanjutnya, jika huruf sama dengan huruf sebelumnya maka hapus huruf tersebut. Jika tidak, simpan huruf dan lakukan hal yang sama pada huruf selanjutnya. Dan melakukan *recording*.
3. Setelah diperiksa untuk setiap huruf periksa kata hasil proses sebelumnya pada kamus.
4. Jika ditemukan maka algoritma berhenti. Jika tidak ditemukan, algoritma mengembalikan kata yang asli sebelum dilakukannya penghapusan huruf berlebihan.
5. Kemudian dilanjutkan dengan periksa kata pada kamus.
6. Jika ditemukan, lakukan perubahan kata menjadi kata baku. Jika tidak ditemukan maka kata dikembalikan pada *root word*.

Dalam penelitian ini kamus normalisasi yang digunakan menggunakan kamus dari penelitian Rosdiansyah (2014).

Setelah proses *spelling normalization* dilakukan, kemudian dilanjutkan proses *filtering*. Proses *filtering* ini adalah proses untuk memperbaiki kata-kata yang dibutuhkan namun tidak sesuai. Misalnya “mantaappp” diubah menjadi “mantap” dengan kata lain, jika ditemui kata berbahasa Indonesia tidak baku maka diganti dengan sinonimnya berupa kata baku yang sesuai KBBI. Selain itu dilakukan

pembuangan kata yang tidak berpengaruh dalam klasifikasi sentimen suatu *tweet* yang disebut *stopword.*

Proses selanjutnya ialah *Stemming*. Proses ini mencari kata dasar dari setiap hasil dari *filtering*. Algoritma *stemming* mempunyai tingkat keakuratan yang lebih baik dari algoritma lainnya adalah algoritma Nazief & Andriani (Agusta, 2009 yang dikutip oleh Rosdiansyah, 2014) pada komentar yang akan diklasifikasi menggunakan bahasa Indonesia. Algoritma ini mengacu pada aturan KBBI yang mengelompokan imbuhan yang diperbolehkan atau imbuhan yang tidak diperbolehkan.

Berikut merupakan langkah-langkah yang dilakukan oleh algoritma Nazief & Andriani (Agusta, 2009 yang dikutip oleh Rosdiansyah, 2014).

1. Kata yang belum di *stemming* dicari pada KBBI. Apabila kata langsung ditemukan, berarti kata tersebut adalah kata dasar, kata dikembalikan dan algoritma dihentikan.
2. Hilangkan *inflectional suffixes* terlebih dahulu, jika ini berhasil dan *suffix* adalah pertikel (“lah” atau “kah”), langkah ini dilakukan lagi untuk menghilangkan *inflectional possesive pronoun suffixes* (“ku”, “mu” atau “nya”).
3. Partikel *Derivational suffix* (“i”,”-an”,”-kan”) kemudian dihilangkan, langkah dilanjutkan lagi untuk mengecek apakah masih ada *derivational suffix* yang tersisa, jika ada maka akan dihilangkan. Apabila tidak ada lagi maka lakuan langkah selanjutnya.
4. *Derivational prefix* (“di-”, “ke-”, “se-”, “te-”, “me-”, “be-”, “pe-”) dihilangkan, kemudian langkah dilanjutkan lagi untuk mengecek apakah masih ada *derivational prefix* yang tersisa, jika ada maka akan dihilangkan. Apabila tidak ada lagi maka lakukan langkah selanjutnya.
5. Setelah tidak ada lagi imbuhan yang tersisa, maka algoritma ini dihentikan kemudian kata dasar tersebut di cari pada KBBI, jika

kata ditemukan berarti algoritma ini berhasil tapi jika kata dasar tidak ditemukan maka dilakukan *recoding.*

6. Jika semua langkah telah dilakukan tetapi kata dasar tidak ditemukan pada kamus, maka algoritma ini mengembalikan kata yang asli sebelum dilakukan *stemming*.

Tahapan proses terkahir ialah *Tagging*. Tahap ini mencari bentuk awal dari kata lampau atau kata *stemming*. Untuk dokumen berbahas Indonesia proses *tagging* tidak diterapkan, karena bahasa Indonesia tidak memiliki bentuk lampau.

2.2.2 *N-Gram*

Setelah selesai melakukan *text preprocessing*, tahap selanjutnya ialah melakukan pemilihan fitur. Menentukan fitur merupakan tugas yang paling penting dalam klasifikasi teks. Seleksi fitur adalah tugas memilih *term* (kata atau istilah) yang akan digunakan dalam *training* set. Dalam hal ini fitur diambil bukan kata per kata tetapi dari keseluruhan dokumen. Hubungan tiap kata dalam sebuah dokumen dianalisa terlebih dahulu untuk mendapatkan relasi antar kata, dan dari kata-kata yang membentuk relasi tersebut diambil untuk menjadi fitur klasifikasi, n-gram adalah potongan **n** karakter dalam suatu string tertentu (Mustika, 2015). Misalnya dalam kata “Kesempatan” akan didapatkan n-gram sebagai berikut.

Tabel 2-1 Contoh pemotongan N-gram berbasis karakter

Nama	n-gram karakter
Uni-gram	*K, E, S, E, M, P, A, T, A, N*
Bi-gram	*_K, KE, ES, SE, EM, MP, PA, AT, TA, AN, N_*
Tri-gram	*_KE, KES, ESE, SEM, EMP, MPA, PAT, ATA, TAN, AN_, N_ _*
Quad-gram	*_KES, KESE, ESEM, SEMP, EMPA, MPAT, PATA, ATAN, TAN_, AN_ _, N_ _ _*

Karakter blank "_" digunakan untuk merepresentasikan spasi di depan dan diakhir kata. Dan untuk *word-based n-gram* contohnya adalah sebagai berikut.

Kalimat : "*N-gram* adalah potongan n karakter dalam suatu string tertentu"

Tabel 2-2 Contoh pemotongan *N-gram* berbasis kata

Nama	*n-gram* karakter
Uni-gram	*n-gram*, adalah, potongan, n, karakter, dalam, suatu, sring, tertentu
Bi-gram	*n-gram* adalah, adalah potongan, potongan n, n karakter, karakter dalam, dalam suatu, suatu sring, string tertentu
Tri-gram	*n-gram* adalah potongan, adalah potongan n, potongan n karakter, n karakter dalam, karakter dalam sesuatu, dalam suatu string, suatu string tertentu
Dst…	

2.2.3 *Transformation*

Pada tahapan ini pemprosesan teks dilanjukan dengan proses transformasi teks menjadi data numerik sebagai reprentasi dari setiap dokumen. Pada text transformation ini kita hanya menentukan (TF) saja, yaitu jumlah frekuensi kemunculan kata dalam dokumen tersebut (Hadna & Paulus Insap Santosa, 2016).

2.2.4 Penggalian Informasi Pada *Text Mining*

Tahap akhir penggalian informasi pada *text mining* yaitu ekstraksi ilmu pengetahuan (*knowledge discovery*), dimana terdapat beberapa jenis kategori utama yang bisa dilakukan sebagai berikut (Miner, dkk, 2012 dikutip oleh Chyntia, 2015).

1. Klasifikasi / prediksi,

Klasifikasi adalah bentuk analisis data yang mengekstrak model untuk menggambarkan kelas data (Jiawei, Kamber, & Pei, 2012 dikutip oleh Chyntia,

2015). Pengklasifikasian meliputi model yang dibangun dan prediksi kategori label kelas. Terdapat dua tahap proses klasifikasi, tahap pertama ialah tahap pembelajaran (*learning step*). Klasifikasi tahap pertama ini dibangun berdasarkan label yang telah diketahui. Dan tahapan kedua ialah tahapan klasifikasi (*classification step*) yang dimana model digunakan untuk memprediksi label kelas dari data yang diberikan (Miner, dkk, 2012 dikutip oleh Chyntia, 2015).

2. Pengelompokan (*Clustering*)

Pada model *clustering* pengelompokan data dilakukan dengan menggunakan algoritma yang sudah ditentukan dan data akan diproses oleh algoritma untuk dikelompokkan menurut karakteristik alaminya. Algoritma akan berjalan dengan sendirinya untuk mengelompokkan data. Data yang lebih dekat (mirip) dengan data lain akan berkelompok dalam satu *cluster*, sedangkan data yang lebih jauh (berbeda) dari data yang lain akan berpisah dalam kelompok yang berbeda.

Untuk masalah pengelompokkan data berdasarkan kemiripan / ketidakmiripan antar data tanpa ada label kelas yang diketahui sebelumnya disebut dengan pembelajaran tidak terbimbing atau *unsupervised learning.* Dalam konteks yang lain, pembelajaran tidak terbimbing disebut juga pengelompokan atau *clustering*. Menurut struktur, *clustering* terbagi menjadi dua, yaitu *hierarki* dan *partisi*. Dalam pengelompokan berbasis hierarki, satu data tunggal bisa dianggap sebuah *cluster*, dua atau lebih *cluster* kecil dapat bergabung menjadi sebuah *cluster* besar, begitu seterusnya hingga semua data dapat bergabung menjadi sebuah *cluster*. Di sisi lain, pengelompokan berbasis partisi membagi set data ke dalam sejumlah *cluster* yang tidak bertumpang-tindih antara satu *cluster* dengan *cluster* yang lain, artinya setiap data hanya menjadi anggota satu *cluster* saja.

3. Asosiasi

Asosiasi merupakan proses pencarian hubungan antar elamen data. Dalam dunia industri retail, analisis asosiasi biasanya disebut *market Basket Analysis* (Miner, dkk, 2012 dikutip oleh Chyntia, 2015). Asosiasi tersebut dihitung berdasarkan ukuran *Support* (presentase dokumen yang memuat seluruh konsep

suatu produk A dan B) dan *confidence* (presentase dokumen yang memuat seluruh konsep produk B yang berada dalam subset yang sama dengan dokumen yang memuat seluruh konsep produk A).

4. Analisis Tren

Tujuan dari analisis tren yaitu untuk mencari perubahan suatu objek atau kejadian oleh waktu (Miner, dkk, 2012 dikutip oleh Chyntia, 2015). Salah satu aplikasi analisis tren yaitu kegiatan identifikasi evolusi topik penelitian pada artikel akademis.

2.3 Naïve Bayes Classifier

Algoritma *naive bayes classifier* merupakan algoritma yang digunakan untuk mencari nilai probabilitas tertinggi untuk mengklasifikasi data uji pada kategori yang paling tepat (Feldman & Sanger, 2007). Dalam penelitian ini yang menjadi data uji adalah dokumen *tweets*. Ada dua tahap pada klasifikaasi dokumen. Tahap pertama adalah pelatihan terhadap dokumen yang sudah diketahui kategorinya. Sedangkan tahap kedua adalah proses klasifikasi dokumen yang belum diketahui kategorinya.

Sebuah keuntungan dari *naïve bayes classifier* adalah bahwa ia memerlukan sejumlah kecil data pelatihan untuk mengestimasi parameter (rata-rata dan varian dari variabel) yang diperlukan untuk klasifikasi. Karena variabel diasumsikan independen, hanya varian dari variabel-variabel untuk setiap kelas yang perlu ditentukan dan bukan keseluruhan *covariance matrix*. Penghitungan nilai probabilitas tersebut menggunakan persamaan, (jurafsky, 2011) :

$$\boldsymbol{P(c) = \frac{N_c}{N}} \tag{2.2}$$

P(c) = Nilai prior setiap kelas.

Nc = Banyak dokumen dalam suatu kelas (n)

N = Jumlah keseluruhan dokumen.

dan

$$\mathbf{P}(\boldsymbol{w}|\boldsymbol{c}) = \frac{\boldsymbol{count}\ (\boldsymbol{w},\boldsymbol{c}+\mathbf{1})}{\boldsymbol{count}\ (\boldsymbol{c})+|\boldsymbol{V}|} \quad (2.3)$$

Count (w,c) = Frekuensi kata **w** pada kelas **c**

Count (c) = Total frekuensi kata pada masing-masing kelas **c**

I V I = Total kata unik pada keseluruhan kelas c

Selanjutnya,

$$\mathbf{P}\left(\mathbf{c}\,\middle|\,\mathbf{dn}\right) = \mathbf{P}(\boldsymbol{c}) * \boldsymbol{\Pi p}(\boldsymbol{w}|\boldsymbol{c}) \quad (2.4)$$

P(c|dn) = Pemilihan Kelas

Pc = *Priors*

$\Pi p(w|c)$ = Total *Conditional Probabilities*

2.4 K-fold *Cross Validation*

Teknik K-*fold cross validation* dapat digunakan jika memiliki jumlah data tang terbatas. Cara kerja K-*fold cross validation* adalah sebagai berikut:

1. Total data dibagi menjadi K bagian.

2. *Fold* ke-1 adalah ketika bagian ke-1 menjadi data uji (*testing data*) dan sisanya menjadi data latih (*training data*). Selanjutnya, hitung akurasi berdasarkan porsi data tersebut.

3. *Fold* ke-2 adalah ketika bagian ke-2 menjadi data uji (*testing data*) dan sisanya menjadi data latih (*training data*). Selanjutnya, hitung akurasi berdasarkan porsi data tersebut.

4. Demikian seterusnya hingga mencapai *fold* ke-K.

5. Hitung rata-rata akurasi dari N buah akurasi di atas. Rata-rata akurasi ini menjadi akurasi final.

Metode k-*fold cross validation* melakukan generalisasi dengan membagi data ke dalam k bagian berukuran sama. Selama proses berlangsung, salah satu dari partisi dipilih untuk *testing*, dan sisanya digunakan untuk *training*. Langkah ini diulangi k kali sehingga setiap partisi digunakan untuk *testing* tepat satu kali. Total

error ditentukan dengan menjumlahkan *error* untuk semua k proses tersebut. Metode k-*fold cross validation* menetapkan k = N, ukuran dari data set. Pendekatan ini memiliki keuntungan dalam pengunaan data sebanyak mungkin untuk *training*. *Test set* bersifat *mutually exclusive* dan secara efektif mencakup keseluruhan data set. Kekurangan dari pendekatan ini adalah banyaknya komputasi untuk mengulangi prosedur sebanyak N kali. K-*fold cross validation* adalah salah satu teknik untuk mengevaluasi keakuratan model.

2.5 Confusion Matrix

Metode ini menggunakan tabel matriks seperti yang terlihat pada 2.3 berikut ini jika data set hanya terdiri dari dua kelas, kelas yang satu dianggap sebagai positif dan yang lainnya negatif. Menurut Rasepta, (2016) Performa dari suatu model kasifikasi dapat diukur dengan tingkat akurasinya berdasarkan *Confusion matrix*.

Tabel 2-3 *Confusion matrix*

Klasifikasi yg benar	Diklasifikasikan sebagai	
	Positives	Negatives
Positives	True positives	False positives
Negatives	False negatives	True negatives

True positives adalah jumlah *record* positif yang diklasifikasikan sebagai positif, *false positives* adalah jumlah *record* positif yang diklasifikasikan sebagai negatif, *false negatives* adalah jumlah *record* negatif yang diklasifikasikan sebagai positif, *true negatives* adalah jumlah *record* negatif yang diklasifikasikan sebagai negatif, kemudian masukkan data uji. Setelah data uji dimasukkan ke dalam *confusion matrix*, hitung nilai-nilai yang telah dimasukkan tersebut untuk dihitung jumlah *sensitivity* (*recall*), *specificity*, *precision* dan *accuracy*. *Sensitivity* digunakan untuk membandingkan jumlah TP terhadap jumlah record yang positif sedangkan *specificity* adalah perbandingan jumlah TN terhadap jumlah *record* yang negatif. Untuk menghitung digunakan persamaan di bawah ini:

$$Sensivity = \frac{TP}{(TP+FN)} \quad (2.5)$$

$$Specifity = \frac{TN}{(FP+TN)} \quad (2.6)$$

$$Precision = \frac{TP}{(TP+FP)} \quad (2.7)$$

$$Accuracy = \frac{TP+TN}{(P+N)} \quad (2.8)$$

Keterangan:

TP = jumlah *true positives*

TN = jumlah *true negatives*

FP = jumlah *false positives*

FN = jumlah *false negatives*

2.6 Penelitian Terkait

Pada table berikut dapat dilihat beberapa penelitian sebelumnya mengenai analisis sentimen :

Table 2-4 Penelitian Terkait

Peneliti	Judul	Metode	Ekstraksi fitur	Domain	Akurasi
Amelia Mustika (2015)	Penerapan Metode Support Vector Machine Dalam Klasifikasi Sentimen *Tweet* Public Figure	SVM	BI-Gram	Twitter	72%

Antonis Rahmad dan Yuan Lukito (2016)	Klasifikasi Sentimen Komentar Dri Fecebook Page menggnakan Naïve Bayes Classifier	NBC	TF-IDF	Facebook Page	83%
Nugroho & dkk (2016)	Analisis Sentimen Pada Jasa Ojek Online Menggunakan Metode Naïve Bayes Classifier	NBC		Twitter pada akun @GojekIndonesia dan @GrabID	80%
Iin Kusumawati (2017)	Analisa Sentimen Menggunakan *Lexicon Based* Untuk Melihat Persepsi Masyarakat Terhadap Kenaikan Harga Jual Rokok Pada Media Sosial *Twitter*	Lexicon Based		Twitter	81%
Dian Fitriana (2017)	Analisis Sentimen Publik Seputar Tren Wisata Pada *Twitter* Menggunakan *Naive Bayes Classifier* dengan Penambahan Fitur	NBC	N-Gram	Twitter	76,8%

	N-gram				
Deden dan Nina (2017)	Analisis Sentimen Pasar Otomotif Mobil : *Tweet Twitter* Menggunakan *Naïve bayes Classifier*	NBC		Twitter	93%