

BAB II

LANDASAN TEORI

Pada bab ini akan diuraikan tentang teori-teori yang melandasi penulisan Laporan Penelitian ini.

2.1 *Opinion Mining*

Analisa sentimen atau biasa disebut *opinion mining* merupakan salah satu cabang penelitian *Text Mining*. *Opinion mining* adalah riset komputasional dari opini, sentimen dan emosi yang diekspresikan secara tekstual. Jika diberikan suatu set dokumen teks yang berisi opini mengenai suatu objek, maka *opinion mining* bertujuan untuk mengekstrak atribut dan komponen dari objek yang telah dikomentasi pada setiap dokumen dan untuk menentukan apakah komentar tersebut bermakna positif atau negatif.^[2]

Sentiment Analysis dapat dibedakan berdasarkan sumber datanya, beberapa level yang sering digunakan dalam penelitian *Sentiment Analysis* adalah *Sentiment Analysis* pada level dokumen dan *Sentiment Analysis* pada level kalimat.^[4] Berdasarkan level sumber datanya *Sentiment Analysis* terbagi menjadi 2 kelompok besar yaitu :

1. *Coarse-grained Sentiment Analysis*
2. *Fined-grained Sentiment Analysis*

2.1.1 *Coarse-grained Sentiment Analysis*

Pada *Sentiment Analysis* jenis ini, *Sentiment Analysis* yang dilakukan adalah pada level dokumen. Secara garis besar fokus utama dari *Sentiment Analysis* jenis ini adalah menganggap seluruh isi dokumen sebagai sebuah sentiment positif atau sentiment negatif.^[4] Salah satu contoh yang menggambarkan *coarse-grained Sentiment Analysis* adalah sebagai berikut :

Tabel 2.1 Contoh *coarse-grained Sentiment Analysis*

i ABSTRAK

Sebagai salah satu perusahaan tambang penyedia bahan galian golongan C jenis pasir dan batu, PD Anugrah perlu mengadakan pengelolaan persediaan karena sering mengalami kekurangan pasir dan penumpukan batu di *stockpile*, sehingga sulit untuk memprediksi jumlah persediaan optimal, dimana persediaan tidak terlalu besar dan terlalu kecil. Selain itu, semakin besar jumlah yang disimpan maka semakin tinggi total biaya persediaan. Saat ini, analisis investasi tambang PD Anugrah menggunakan metode *Break Even Point*, tetapi pengelolaan datanya masih menggunakan buku tambang dan kalkulator. Oleh karena itu, aplikasi ini dikembangkan untuk pengelolaan data dan perhitungan secara terkomputerisasi. Aplikasi ini diharapkan dapat membantu untuk memprediksi jumlah persediaan optimal di *stockpile* menggunakan metode *Safety Stock* dan *Reorder Point* dengan memanfaatkan informasi dalam buku tambang, meminimalkan total biaya persediaan menurut perhitungan perusahaan dan menurut metode *Wagner Within Algorithm*, serta membantu dalam analisis investasi tambang menggunakan metode *Break Even Point* untuk 4 jenis bahan galian (pasir, batu belah, batu skrop, dan sirtu), metode *Net Present Value*, dan metode *Payback Period*. Aplikasi ini menyediakan informasi tambang dengan data sesuai keadaan sesungguhnya yaitu 2 tahun masa umur tambang.

Kata Kunci : Persediaan, *Wagner Within Algorithm*, Analisis Investasi, dan Pertambangan.

2.1.2 *Fined-grained Sentiment Analysis*

Fined-grained Sentiment Analysis adalah *Sentiment Analysis* pada level kalimat. Fokus utama *fined-grained Sentiment Analysis* adalah menentukan sentimen pada setiap kalimat.^[4] Salah satu contoh yang menggambarkan *Fined-grained Sentiment Analysis* adalah sebagai berikut :

Tabel 2. 2 Contoh *Fined-grained Sentiment Analysis*

username	Tweet
humar_bis	Parkir liar di Soekarno Hatta kapan ditindak? Bikin macet. Naik Damri Metro-Leuwi Panjang bs 1,5 jam, 9 km. @PemkotBandung

2.2 *Text Mining*

Text mining dapat didefinisikan secara luas sebagai proses pengetahuan intensif dimana pengguna berinteraksi dengan koleksi dokumen dari waktu ke

waktu dengan menggunakan seperangkat alat analisis. *Text mining* berusaha untuk mengekstrak informasi yang berguna dari sumber data melalui identifikasi dan eksplorasi pola yang menarik. *Text mining* cenderung mengarah pada bidang penelitian *data mining*. Oleh karena itu, tidak mengherankan bahwa *text mining* dan *data mining* akan berada pada tingkat arsitektur yang sama (Feldman, dkk. 2007)^[7]. Penambangan teks dapat dianggap sebagai proses dua tahap yang diawali dengan penerapan struktur terhadap sumber data teks dan dilanjutkan dengan ekstraksi informasi dan pengetahuan yang relevan dari data teks terstruktur ini dengan menggunakan teknik dan alat yang sama dengan penambangan data^[7].

2.2.1 Pengertian *Text Preprocessing*

Text Preprocessing merupakan tahapan dari proses awal terhadap teks untuk mempersiapkan teks menjadi data yang akan diolah lebih lanjut. Sebuah teks yang ada harus dipisahkan, hal ini dapat dilakukan dalam beberapa tingkatan yang berbeda. Suatu dokumen atau *tweet* dapat dipecah menjadi bab, sub-bab, paragraf, kalimat dan pada akhirnya menjadi potongan kata/token. Selain itu pada tahapan ini keberadaan digit angka, huruf kapital, atau karakter- karakter yang lainnya dihilangkan dan dirubah (Ronen Feldman, 2007)^[7].

2.2.2 *Feature Selection*

Tahap seleksi fitur (*feature selection*) bertujuan untuk mengurangi dimensi dari suatu kumpulan teks, atau dengan kata lain menghapus kata-kata yang dianggap tidak penting atau tidak menggambarkan isi dokumen sehingga proses pengklasifikasian lebih efektif dan akurat (Feldman & Sanger, 2007.)^[7]. Pada tahap ini tindakan yang dilakukan adalah menghilangkan *stopword* (*stopword removal*) dan *stemming* terhadap kata yang berimbunan.

Stopword adalah kosakata yang bukan merupakan ciri (kata unik) dari suatu dokumen. Misalnya “di”, “oleh”, “pada”, “sebuah”, “karena” dan lain sebagainya. Sebelum proses *stopword removal* dilakukan, harus dibuat daftar *stopword* (*stoplist*). Jika termasuk di dalam *stoplist* maka kata-kata tersebut akan dihapus dari deskripsi sehingga kata-kata yang tersisa di dalam deskripsi dianggap sebagai kata-kata yang mencirikan isi dari suatu dokumen atau *keywords*. Setelah

melalui proses *stopword removal* tindakan selanjutnya adalah yaitu proses *stemming* ^[7].

Stemming adalah proses pemetaan dan penguraian berbagai bentuk (*variants*) dari suatu kata menjadi bentuk kata dasarnya (*stem*). Tujuan dari proses *stemming* adalah menghilangkan imbuhan-imbuhan baik itu berupa prefiks, sufiks, maupun konfiks yang ada pada setiap kata. Jika imbuhan tersebut tidak dihilangkan maka setiap satu kata dasar akan disimpan dengan berbagai macam bentuk yang berbeda sesuai dengan imbuhan yang melekatinya sehingga hal tersebut akan menambah beban database. Hal ini sangat berbeda jika menghilangkan imbuhan-imbuhan yang melekat dari setiap kata dasar, maka satu kata dasar akan disimpan sekali walaupun mungkin kata dasar tersebut pada sumber data sudah berubah dari bentuk aslinya dan mendapatkan berbagai macam imbuhan. Karena bahasa Indonesia mempunyai aturan morfologi maka proses *stemming* harus berdasarkan aturan morfologi bahasa Indonesia. Berdasarkan penelitian sebelumnya, ada beberapa *algoritma stemming* yang bisa digunakan untuk *stemming* bahasa Indonesia diantaranya *algoritma confixstripping*, *algoritma Porter stemmer* bahasa Indonesia, *algoritma Arifin dan Sutiono*, dan *Algoritma Idris* (Tala 2003, Agusta 2009, Asian et al 2005, Adriani et al 2007) ^[7]

2.2.3 Kata Dasar^[5]

Kata dasar yaitu kata yang belum mengalami perubahan kata (morfologis) seperti penambahan imbuhan awalan (prefix), akhiran (suffiix), sisipan (infix), infleksi, reduplikasi,

1. Nomina

Dari segi semantisnya, nomina adalah kata yang mengacu pada manusia, binatang, benda, dan konsep atau pengertian. Misalnya dosen, tikus, kursi, bahasa. Dari segi sintaksisnya, nomina mempunyai ciri-ciri:

- a. Menduduki fungsi *subjek*, *objek*, atau pelengkap. Misalnya, ayah membelikan adik buku.
- b. Dapat diingkarkan dengan kata bukan seperti bukan buku, bukan rumah, dan tidak dapat diingkarkan dengan kata tidak karena tidak ada bentuk **tidak buku*, **tidak rumah*, dsb.
- c. Umumnya diikuti adjektiva, baik secara langsung maupun diantara kata yang. Misalnya gadis cantik, gadis yang cantik.

2. Adjektiva

Adjektiva adalah kata yang memberikan keterangan yang lebih khusus tentang sesuatu yang dinyatakan oleh nomina dalam kalimat. Adjektiva sering disebut juga kata keadaan. Ciri-ciri adjektiva:

- a. Adjektiva memberikan makna kualitas atau keanggotaan dalam suatu golongan. Misalnya pohon tinggi, rumah besar, dan baju merah.
- b. Adjektiva dapat berfungsi sebagai predikat dan adverbial (keterangan) kalimat yang dapat mengacu ke suatu keadaan. Misalnya: Ibu sedang sakit.
- c. Adjektiva memiliki kemungkinan untuk menyatakan tingkat kualitas dan tingkat bandingan acuan nomina yang diterangkannya dengan menambahkan kata sangat, agak, lebih, atau paling di depan adjektiva tersebut. Misalnya: sangat besar, agak senang, lebih kecil, paling merah.

3. Verba

Verba sering disebut juga kata kerja. Ciri-ciri verba:

- a. Memiliki fungsi utama sebagai predikat atau sebagai inti predikat dalam kalimat walaupun dapat juga mempunyai fungsi lain. Misalnya:
 - Kakek *Tidur*
 - Ibu tidak *Menulis* novel

- b. Mengandung makna inheren perbuatan (aksi), proses, atau keadaan yang bukan sifat atau kualitas.
- c. Tidak diberi prefiks ter- yang berarti ‘paling’. Misalnya verba mati dan suka tidak dapat menjadi **termati* atau **tersuka*.
- d. Pada umumnya tidak dapat bergabung dengan kata-kata yang menyatakan makna kesangatan seperti agak, sangat, dan sekali karena tidak ada bentuk **agak belajar*, **sangat tidur*, **duduk sekali* meskipun ada bentuk seperti sangat berbahaya, agak

2.2.4 Aturan Peluruhan Kata Dasar

Ada beberapa kata dasar yang apabila dilekati oleh awalan "me(N)-", "pe(N)-", "pe(R)-", "te(R)-", "be(R)-" akan mengalami peluruhan atau perubahan pada karakter awal dari kata dasar tersebut (Kridalaksana, 2009). Sebagai contoh kata "tanya", karakter awal dari kata "tanya" akan berubah apabila ditambahkan awalan "me-" dan menjadi "menanya". Begitu juga untuk beberapa kata dasar lainnya.^[10]

Untuk melakukan proses stemming pada kata-kata tersebut harus mengikuti aturan peluruhan yang telah ditetapkan oleh algoritma (Adriani et al, 2007)^[10]. Aturan- aturan tersebut dijelaskan pada gambar 2.4.

Aturan	Bentuk Awalan	Peluruhan
1	berV...	ber-V... be-rV...
2	belajar...	bel-ajar
3	beC ₁ erC ₂ ...	be-C ₁ erC ₂ ...dimana C ₁ !={r l}
4	terV...	ter-V... te-rV...
5	terCer...	ter-Cer...dimana C!=r
6	teC ₁ erC ₂	te-C ₁ erC ₂ ...dimana C ₁ !={r}
7	me{l r w y}V...	me-{l r w y}V...
8	mem{b f v}...	mem-{b f v}...
9	mempe...	mem-pe...
10	mem{rV V}...	me-m{rV V}... me-p{rV V}...
11	men{c d j z}...	men-{c d j z}...
12	menV...	me-nV... me-tV...
13	meng{g h q k}...	meng-{g h q k}...
14	mengV...	meng-V... meng-kV...

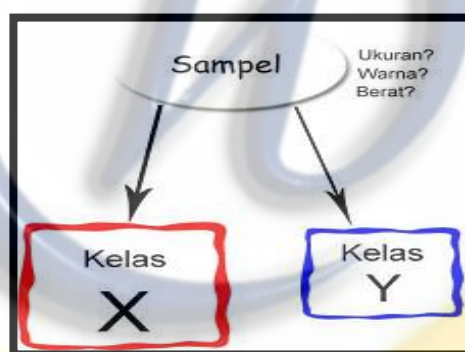
Gambar 2.1 Aturan Peluruhan kata Dasar

2.2.5 Stoplist

Stopword adalah bentuk kata umum (*common words*) yang biasanya muncul dalam jumlah besar dan dianggap tidak memiliki makna. Contoh *stopword* dalam bahasa Indonesia diantaranya, “yang”, “di”, “ke”. Contoh tabel stoplist akan dijelaskan di bab selanjutnya.

2.3 Klasifikasi

Klasifikasi adalah proses untuk menemukan model atau fungsi yang menjelaskan atau membedakan konsep atau kelas data, dengan tujuan untuk dapat memperkirakan kelas dari suatu obyek (Agus Mulyanto 2009). Oleh karena itu, kelas yang ada tentulah lebih dari satu. Penentuan kelas dari suatu dokumen dilakukan dengan cara membandingkan nilai probabilitas suatu sampel berada di kelas yang satu dengan nilai probabilitas suatu sampel berada di kelas yang lain.



Gambar 2.2 Ilustrasi contoh proses klasifikasi

2.4 *Naïve Bayes*

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)^[7]. Dalam penelitian ini yang menjadi data uji adalah dokumen *tweets*. Ada dua tahap pada klasifikasi dokumen. Tahap pertama adalah pelatihan terhadap dokumen yang sudah diketahui kategorinya. Sedangkan tahap kedua adalah proses klasifikasi dokumen yang belum diketahui kategorinya.

Dalam algoritma *naïve bayes classifier* setiap dokumen direpresentasikan dengan pasangan atribut “ $X_1, X_2, X_3 \dots X_n$ ” dimana X_1 adalah kata pertama, X_2 adalah kata kedua dan seterusnya. Sedangkan V adalah himpunan Kategori *Tweet*. Pada saat klasifikasi algoritma akan mencari probabilitas tertinggi dari semua kategori dokumen yang diujikan (V_{map}), dimana persamaannya adalah sebagai berikut :

$$V_{\text{MAP}} = \underset{V_j \in V}{\text{arg max}} \frac{P(x_1, x_2, x_3, \dots, x_n | V_j) P(V_j)}{P(x_1, x_2, x_3, \dots, x_n)}$$

Untuk $P(X_1, X_2, X_3 \dots X_n)$ nilainya konstan untuk semua kategori (V_j) sehingga persamaan dapat ditulis sebagai berikut :

$$V_{\text{MAP}} = \underset{V_j \in V}{\text{arg max}} P(x_1, x_2, x_3, \dots, x_n | V_j) P(V_j)$$

Persamaan diatas dapat disederhanakan menjadi sebagai berikut :

$$V_{\text{MAP}} = \underset{V_j \in V}{\text{arg max}} \prod_{i=1}^n P(x_i | V_j) P(V_j)$$

Keterangan :

V_j = kategori tweet $j = 1,2,3,..n$ Dimana dalam penelitian ini

j_1 = kategori tweet sentimen negatif,

j_2 = kategori tweet sentimen positif,

j_3 = kategori tweet sentiment netral.

$P(X_i|V_j)$ = Probalitas X_i pada kategori V_j

$P(V_j)$ = Probalitas dari V_j

Untuk $P(V_j)$ dan $P(X_i|V_j)$ dihitung pada saat pelatihan dimana persamaannya adalah sebagai berikut :

$$P(V_j) = \frac{|docs\ j|}{|contoh|}$$

$$P(x_i|V_j) = \frac{n_k + 1}{n + |kosakata|}$$

Keterangan :

$|docs\ j|$ = jumlah dokumen setiap kategori j

$|contoh|$ = jumlah dokumen dari semua kategori

n_k = jumlah frekuensi kemunculan setiap kata

n = jumlah frekuensi kemunculan kata dari setiap kategori

$|kosakata|$ = jumlah semua kata dari semua kategori.

2.5 *KNIME Analytics Platform*

Pengembangan *KNIME* dimulai Januari 2004 oleh *Developer* perangkat lunak di *Konstanz University* sebagai produk *proprietary*. Tim pengembang asli yang dipimpin oleh *Michael Berthold* berasal dari sebuah perusahaan di *Silicon Valley* menyediakan perangkat lunak untuk industri farmasi. *KNIME* telah dikembangkan sejak hari pertama menggunakan profesional proses rekayasa perangkat lunak yang ketat karena itu jelas dari awal bahwa itu akan digunakan di perusahaan-perusahaan skala besar. Tujuan awalnya adalah untuk menciptakan sebuah platform pengolahan data modular, sangat sederhana dan *opensource* yang memungkinkan untuk integrasi yang dapat digunakan di banyak platform seperti *windows, linux, mac os*. Eksplorasi visual yang tanpa fokus pada area aplikasi tertentu. Platform ini dimaksudkan untuk menjadi kolaborasi dan penelitian Platform dan juga harus melayani sebagai platform integrasi untuk berbagai proyek analisis data lainnya.^[8]

The KNIME Analytics merupakan landasan menggabungkan ratusan node pengolahan data *input/output, preprocessing* dan pembersihan, pemodelan, analisis dan data mining serta berbagai *tools* interaktif, seperti plot pencari, koordinat paralel dan lain-lain. Aplikasi ini mengintegrasikan semua modul analisis weka data mining terkenal dan plugin tambahan memungkinkan *R-script* yang akan dijalankan dan menawarkan akses ke perpustakaan besar statistik.^[8] *KNIME* didasarkan pada platform *Eclipse* dan, melalui *API modular*, mudah *extensible* ketika diinginkan, node kustom dan jenis dapat diimplementasikan di *KNIME* dalam beberapa jam sehingga memperpanjang *KNIME* untuk memahami dan memberikan dukungan tingkat pertama untuk data yang sangat-domain tertentu.^[8]

2.6 *Twitter*

Twitter merupakan sebuah situs media sosial mikroblog yang memungkinkan penggunaannya untuk mengirim dan membaca pesan singkat yang disebut dengan *tweets* atau kicauan. Penulisan *tweets* sendiri maksimum 140 karakter. *Twitter* didirikan pertama kali oleh *Jack Dorsey* dari perusahaan

podcasting Odeo. Awalnya layanan ini diberi nama “*twtr*”, terinspirasi dari nama situs *sharing* foto *flickr*. Pada juli 2006 media sosial ini diluncurkan untuk publik dengan nama “*Twitter*”. Fitur yang terdapat pada *twitter* ^[5], antara lain:

1. Halaman Utama (*Home*)

Halaman utama merupakan kumpulan tweets yang di posting oleh pengguna baik informasi, berita, gambar bahkan cek-in di suatu wilayah atau tempat dengan waktu bersamaan.

2. Profil (*Profile*)

Profil merupakan kumpulan tweet yang pernah dibuat pengguna dan informasi mengenai data diri.

3. *Following*

Following merupakan akun seseorang yang mengikuti dengan akun twitter lain. tweet yang di posting oleh following akan muncul di halaman utama.

4. *Followers*

Followers merupakan pengguna yang ingin berteman dan mengetahui aktifitas tweet seseorang. tweet yang di posting oleh followers akan muncul di halaman utama.

5. *Mentions*

Konten ini merupakan balasan dari percakapan agar pengguna dapat langsung menandai orang yang akan diajak bicara.

6. *Favorite*

Tweets ditandai sebagai favorit agar tidak hilang dan mudah untuk dicari untuk menemukan tweet sebelumnya.

7. *Pesan Langsung(Direct Message)*

Direct message merupakan media pengiriman pesan secara privasi antar pengguna.

8. *Hastag*

Hastag ditandai dengan “#” berfungsi untuk mencari topik yang sejenis.

9. *List*

Pengguna twitter dapat mengelompokan *followers* kedalam satu grup sehingga memudahkan untuk dapat melihat informasi secara keseluruhan.

10. *Trending Topic*

Topik yang sedang ramai dibicarakan oleh banyak pengguna dalam suatu waktu bersamaan.

2.6.1 *Twitter API (Application Programming Interface)*

Application Programming Interface merupakan sebuah fungsi atau perintah-perintah yang digunakan untuk mengembangkan bahasa dalam *system calls* untuk memudahkan *developer* dengan bahasa atau fungsi yang lebih terstruktur. Para *developer* menggunakan *twitter API* untuk membuat aplikasi, website, dan informasi lain yang berinteraksi dengan *Twitter*. Dokumentasi mengenai *twitter API* dapat dilihat pada <http://dev.twitter.com>

System call Interface berfungsi sebagai penghubung antara API dan *system call* yang dimengerti oleh sistem operasi. *System call Interface* menerjemahkan perintah dalam API dan memanggil fungsi yang tersedia dalam *System call* yang diperlukan. Perintah dari *user* tersebut, diterjemahkan oleh program menjadi perintah *open ()*. *Twitter API* terdiri dari 3 (tiga) bagian yaitu ^[6]:

1. *Search API*

Dirancang untuk memudahkan user dalam mengelola *query search* di konten *twitter*. *User* dapat menggunakannya untuk mencari *keyword* berdasarkan kata khusus atau mencari tweet lebih *spesifik* berdasarkan *username twitter*.

2. *Representational State Transfer (REST) API*

Twitter REST memberikan *core data dan core twitter objects*, 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*.

3. *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.

2.7 Penelitian Terdahulu

Penelitian mengenai klasifikasi sentimen berbahasa Indonesia telah dilakukan oleh Yudi Wibisono (2013). Pada papernya, Yudi Wibisono melakukan klasifikasi sentimen terhadap *Opini Microblogging*. Pada penelitian tersebut digunakan dua kata kunci yaitu perbandingan opini masyarakat mengenai *provider* Indosat dan Telkomsel. Hasil eksperimen menunjukkan akurasi untuk model klasifikasi *tweet* opini dan non opini adalah 91.83% sedangkan untuk model klasifikasi opini negatif dan positif mencapai 93.40%. Sedangkan untuk *keyword extraction* masih kurang memuaskan karena hanya 34.48% *keyword* yang dapat diambil. Penelitian yang serupa juga dilakukan oleh Paulina Aliandu (2013). Penelitian ini melakukan eksperimen untuk melakukan klasifikasi sentimen terhadap data yang diperoleh dari *Twitter* dengan mengambil *Tweet* akun Presiden RI @SBYudhoyono baik sentimen positif, negatif ataupun netral. Aliandu menerapkan *Naive Bayes Method* untuk klasifikasi sentimen tersebut dan dapat mengklasifikasi dengan baik dengan akurasi 79,42% (Aliandu, 2013).

No	Peneliti Tahun	Judul	Metode	Hasil (%)
1	Yudi Wibisono,2013	Sistem Analisis Opini <i>Microblogging</i> Berbahasa Indonesia	Naïve Bayes pembotan TF-IDF	<ul style="list-style-type: none"> klasifikasi <i>tweet</i> opini dan non opini adalah 91.83% klasifikasi opini negatif dan positif mencapai 93.40% keyword extraction masih kurang memuaskan karena hanya 34.48%
2	Aliandu,2013	<i>Twitter Used by Indonesian President SentimentAnalysis of Timeline</i>	Naïve Bayes Suport Vector Machine (SVM)	<ul style="list-style-type: none"> Mengklasifikasi dengan baik dengan akurasi 79,42%
3	Sandi Fajar Rodiyansyah, Edi Winarko ,2013	Klasifikasi Posting Twitter Kemacetan Lalu Lintas Kota Bandung Menggunakan Naïve Bayes Clasification	Naïve Bayes Clasification	<ul style="list-style-type: none"> Nilai akurasi tertinggi sebesar 91,60%

Tabel 2.3 Hasil Penelitian Terdahulu

2.8 Pengembangan Sistem

Penelitian yang terkait dengan analisis sentimen telah banyak dilakukan sebelumnya, tetapi untuk opinion mining dengan menggunakan bahasa Indonesia tidak banyak digunakan karena struktur bahasa yang sangat kompleks. Pada penelitian selanjutnya penulis mengembangkan sistem yang sudah ada dengan menambahkan kosa kata baru kedalam proses *sentiment* dan menambahkan proses pengambilan data menggunakan *tools* yang memudahkan untuk para pengembang selanjutnya, data yang diambil menggunakan *tools* KNIME diantaranya 12 *tweet* akun dinas kota Bandung, 2 *tweet* kecamatan, 2 *tweet* kelurahan dan 2 *account tweet* walikota.. Dalam penelitian terdahulu proses pengambilan data menggunakan *crawler* dimana proses tersebut membutuhkan banyak waktu dalam pengerjaan.

Proses *text mining* pada sistem terdahulu menggunakan beberapa metode diantaranya TF-IDF dan Ngram. TF-IDF adalah metode untuk pembobotan kalimat dalam sebuah dokumen, sedangkan Ngram adalah metode untuk memecah kata menjadi suku kata. Pada proses yang dikembangkan penulis proses *text mining* hanya menggunakan TF (*Term Frequency*) karena pada kalimat *twitter* tidak sebanyak kalimat pada dokumen. Sedangkan metode ngram tidak digunakan pada sistem ini dikarenakan *keyword* yang telah didapat pada proses *data training* telah masuk pada kategori yaitu positif, negatif dan, netral.

Dilihat dari permasalahan yang ada, maka diperlukan sebuah solusi berupa analisis terhadap opini masyarakat kota Bandung tentang Dinas terkait, sehingga dapat diketahui informasi sentimen. Adapun permasalahan pengkategorian opini negatif, positif dan netral dapat diselesaikan dengan menggunakan metode *Naïve Bayes*. Maka maksud dalam pengembangan sistem yang dibuat penulis adalah menerapkan metode *Naïve Bayes* dengan mengambil *tweet* opini untuk dilihat pengkategorian dan hasil akurasi yang tepat.