BAB II

TINJAUAN PUSTAKA

2.1 Netflix

Netflix adalah layanan *streaming* yang menawarkan berbagai acara televisi, film, film dokumenter, dan anime yang diakses dengan perangkat yang terhubung ke internet. Pengguna Netflix bisa menonton sepuasnya, kapan pun, dimana pun, dengan media apa pun dengan biaya langganan per bulan. Kantor pusat Netflix berada di Los Gatos, California. Netflix didirikan pada tahun 1997 oleh Reed Hasting sand Marc Randolph di Scotts Valley, California.

Model bisnis awal Netfilix adalah penjualan *Digital Video Disc* (DVD) dan rental melalui pengiriman. Satu tahun setelah berdiri, Netfilix fokus kepada penyewaan DVD dari pada penjualan DVD, sehingga bisnis usaha penjualan DVD ditinggalkan. Pada tahun 2007, Netflix memperluas bisnisnya dengan mengenalkan media *streaming* namun tetap mempertahankan layanan penyewaan DVD dan Blu-ray. Perusahaan ini memperluas usahanya secara internasional, dengan layanan *streaming* tersedia di Canada pada tahun 2010 dan sejak saat itu layanan tersebut semakin berkembang. Sejak Januari 2016, layanan Netflix telah beroperasi lebih dari 190 negara, layanan ini tersedia secara bebas di Internet kecuali daratan China, Suriah, Republik Krimea, dan Indonesia (beberapa penyedia layanan internet memblokirnya karena masalah sensor).

Sejak Juli 2018, Netflix memiliki lebih dari 130 juta total pelanggangan secara internasional, *term*asuk 57.38 juta di Amerika Serikat sendiri. Upaya mereka adalah memproduksi konten baru, mengamankan hak untuk konten tambahan, dan perbedaan melalui 190 negara telah mengakibatkan perusahaan untuk mengajukan miliaran utang jangka panjang..

Kantor pusat Netflix beralamat di 121 Albright Way, Los Gatos, California, Amerika Serikat. Mereka juga memiliki kantor di Belanda, Brasil, India, Jepang dan Korea Selatan.

2.2 *Twitter*

Twitter adalah layanan jejaring sosial dan mikroblog daring yang memungkinkan penggunanya untuk mengirim dan membaca pesan berbasis teks dengan batas 140 karakter, akan tetapi pada tanggal 7 November 2017 bertambah hingga 280 karakter yang dikenal dengan sebuah kicauan (*tweet*). *Twitter* didirikan pada bulan Maret 2006 oleh Jack Dorsey, dan situs *Twitter* mulai diluncurkan pada bulan Juli. *Twitter* Inc berbasis di San Fransisco dengan kantor tambahan di New York, Boston, dan San Antonio. Sejak diluncurkan *Twitter* telah menjadi salah satu dari sepuluh situs yang paling sering dikunjungi di internet. Pengguna yang tidak terdaftar hanya bisa membaca kicauan, sedangakan pengguna terdaftar bisa menulis, menyukai, dan menyebarkan kicauan.

Tingginya popularitas *Twitter* menyebabkan layanan ini telah dimanfaatkan untuk berbagai keperluan dalam berbagai aspek, misalnya sebagai sarana protes, kampanye politik, sarana pembelajaran, dan sebagai media komunikasi darurat.. Semua pengguna dapat mengirim dan menerima *tweet* melalui situs *Twitter*, aplikasi eksternal yang kompatibel (telepon seluler), atau dengan pesan singkat (SMS) yang tersedia di negara-negara tertentu (*Twitter*, 2013). Pengguna dapat menulis pesan berdasarkan topik dengan menggunakan tanda # (hashtag). Sedangkan untuk menyebutkan atau membalas pesan dari pengguna lain bisa menggunakan tanda @. Fitur yang terdapat dalam *Twitter*, antara lain:

1. Laman Utama (*Home*)

Pada halaman utama kita bisa melihat *tweets* yang dikirimkan oleh orang-orang yang menjadi teman kita atau yang kita ikuti (*following*).

2. Profil (*Profile*)

Pada halaman ini yang akan dilihat oleh seluruh orang mengenai profil atau data diri serta *tweets* yang sudah pernah kita buat.

3. Pengikut *(Followers)*

Pengikut adalah pengguna lain yang ingin menjadikan kita sebagai teman. Bila pengguna lain menjadi pengikut akun seseorang, maka *tweets* seseorang yang ia ikuti tersebut akan masuk ke dalam halaman utama.

4. Akun yang diikuti *(Following)*

Kebalikan dari pengikut, *following* adalah akun seseorang yang mengikuti akun pengguna lain agar *tweets* yang dikirim oleh orang yang diikuti tersebut masuk ke dalam halaman utama.

5. *Mentions*

Biasanya konten ini merupakan balasan dari percakapan agar sesama pengguna bisa langsung menandai orang yang akan diajak bicara.

6. Pesan Langsung (*Direct Message*)

Fungsi pesan langsung lebih bisa disebut SMS karena pengiriman pesan langsung di antara pengguna.

7. Hashtag (#)

Hashtag "#" yang ditulis di depan topik tertentu agar pengguna lain bisa mencari topik yang sejenis yang ditulis oleh orang lain juga

8. Topik Terkini (*Trending Topic*)

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

2.3 Analisis Sentimen

Menurut (Nasukawa & Yi, 2003) Analisis sentimen adalah sebuah teknik atau cara yang digunakan untuk mengidentifikasi bagaimana sebuah sentimen diekspresikan menggunakan teks dan bagaimana sentimen tersebut bisa dikategorikan sebagai sentimen positif maupun sentimen negatif. Sedangkan menurut (Coletta et al., 2014), analisis sentimen adalah proses yang digunakan untuk menentukan opini, emosi dan sikap yang dicerminkan melalui teks, dan biasanya diklasifikasikan menjadi opini negatif dan positif

Berdasarkan tiga pendapat di atas, dapat disimpulkan bahwa analisis sentimen adalah sebuah proses untuk menentukan sentimen atau opini dari seseorang yang diwujudkan dalam bentuk teks dan bisa dikategorikan sebagai sentimen posisif, negatif, atau netral.

Sebagaimana yang sudah dituliskan sebelumnya bahwa pengguna internet banyak menuliskan pengalaman, opini dan segala hal yang menjadi perhatian mereka. Tulisan tentang apa yang mereka rasakan ini berupa perasaan positif, netral maupun negatif yang bisa diungkapkan dengan cara yang cukup kompleks (Troussas et al., 2013).

2.4 Text Mining

Text mining atau penambangan teks adalah proses ekstraksi pola berupa informasi dan pengetahuan yang berguna dari sejumlah besar sumber data teks, seperti dokumen Word, PDF, kutipan teks, dll. Menurut (Berry & Kogan, 2010), *text mining* merupakan teknik yang digunakan untuk menangani masalah klasifikasi, *clustering*, *information extraction dan information retrival*.

Pada dasarnya proses kerja dari *text mining* banyak mengapdopsi dari penelitian data *mining* namun yang menjadi perbedaan adalah pola yang digunakan oleh *text mining* diambil dari sekumpulan bahasa alami yang tidak terstruktur sedangkan dalam data *mining* pola yang diambil dari *database* yang terstruktur (Han & Kamber, 2006). Tahap-tahap *text mining* secara umum adalah *text preprocessing* dan *feature selection* (Feldman & Sanger 2007, Berry & Kogan 2010). Dimana penjelasan dari tahap-tahap tersebut adalah sebagai berikut :

2.4.1 *Text Preprocessing*

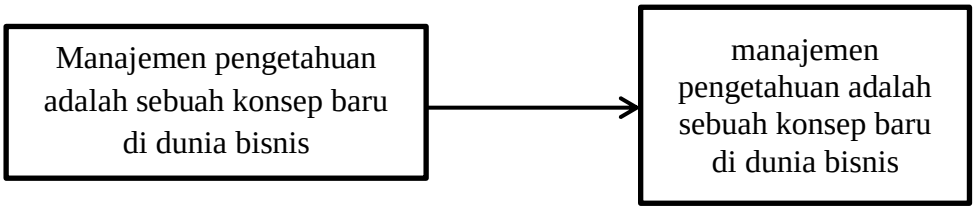
Tahap *text preprocessing* adalah tahap awal dari *text mining*. Tahap ini mencakup semua rutinitas, dan proses untuk mempersiapkan data yang akan digunakan pada operasi *knowledge discovery* sistem *text mining* (Feldman & Sanger, 2007). *Text preprocessing* berfungsi untuk mengubah data teks yang tidak terstruktur atau sembarang menjadi data yang terstruktur. Tahapan dalam proses *preprocessing* adalah sebagai berikut:

a. Case Folding

Case Folding adalah suatu bentuk *text preprocessing* yang paling sederhana dan efektif meskipun sering diabaikan. Tujuan dari *case folding* untuk mengubah semua huruf dalam dokumen menjadi huruf kecil. Hanya huruf "a" sampai "z" yang diterima. Karakter selain huruf dihilangkan dan dianggap *delimiter*. Beberapa cara yang digunakan pada *case folding* adalah sebagai berikut: mengubah teks menjadi *lower case*, menghapus angka, menghapus tanda baca, dan menghapus *whitepace* (karakter kosong).

b. Tokenizing

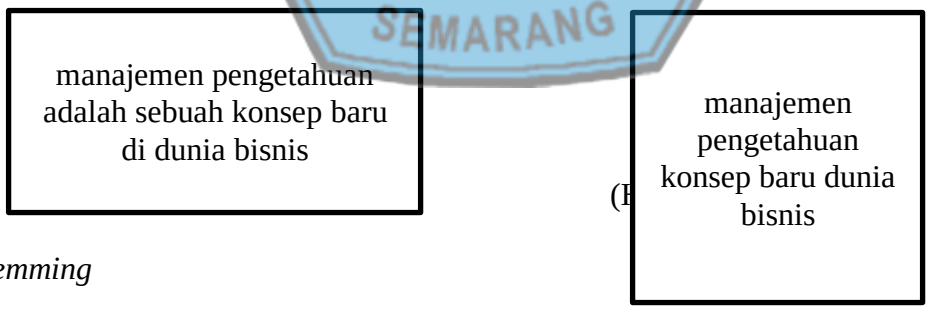
Tokenizing adalah proses penguraian deskripsi yang semula berupa kalimat-kalimat menjadi kata-kata. Kata, angka, simbol, tanda baca dan entitas penting lainnya dapat dianggap sebagai token.



(Teks input) (Teks Output)

c. Filtering (Stopword Removal)

Filtering adalah tahap mengambil kata-kata penting dari hasil token dengan menggunakan algoritma *stoplist* (membuang kata kurang penting) atau *wordlist* (menyimpan kata penting). *Stopword* adalah kata umum yang biasanya muncul dalam jumlah besar dan dianggap tidak memiliki makna. Contoh *stopword* dalam bahasa Indonesia adalah "yang", "dan", "di", "dari", dll. Makna di balik penggunaan *stopword* yaitu dengan menghapus kata-kata yang memiliki informasi rendah dari sebuah teks, kita dapat fokus pada kata-kata penting sebagai gantinya.



(F

d. Stemming

Stemming adalah proses mendapatkan kata dasar dengan menghilangkan imbuhan kata.. Misalnya kata "mendengarkan", "dengarkan", "didengarkan" akan ditransformasi menjadi kata "dengar". Proses *stemming* pada teks Bahasa Indonesia berbeda dengan Bahasa Inggris. Pada Bahasa Inggris yang diperlukan hanya proses menghilangkan akhiran (*sufiks*). Sedangkan pada

teks Bahasa Indonesia semua kata imbuhan baik itu akhiran (*sufiks*) dan awalan (*prefiks*) juga dihilangkan.

membela menguatnya dikatakan dibandingkan

(hasil filtering)

bela menguat kata dibanding

(hasil stemming)

2.4.2 *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 (Do et al, 2006., Feldman & Sanger, 2007, Berry & Kogan 2010). Tahap *feature selection* merupakan tahap lanjutan dari *preprocessing*. Salah satu fungsi dari *feature selection* adalah pemilihan *term* atau kata-kata apa saja yang dapat mewakili dokumen yang akan dianalisis dengan melakukan pembobotan terhadap setiap *term*. *Term* dapat berupa kata atau frase dalam suatu dokumen yang dapat digunakan mengetahui konteks dari dokumen tersebut.

1. *Term Frequency Inverse Document Frequency*

Term Frequency Inverse Document Frequency (TF-IDF) merupakan sebuah metode pembobotan yang dilakukan untuk ekstraksi data teks. Tujuan dari TF-IDF adalah untuk menemukan jumlah kata yang diketahui (tf) setelah dikalikan dengan beberapa banyak *tweet* dimana suatu kata tersebut muncul (idf). Metode TF-IDF dilakukan dengan menghitung bobot dengan cara integrasi antara *term frequency* (tf) dan *inverse document frequency* (idf). Berikut merupakan rumus untuk menemukan pembobot dengan TF-IDF.

$$idf = \log \frac{N}{df_j} \qquad (2.1)$$

$$w_{ij} = tf_{ij}\, x\, idf_j \quad (2.2)$$

2.5 Naïve Bayes Classifier (NBC)

Teorema Bayes merupakan teorema yang mengacu pada probabilitas bersyarat. Secara umum teorema Bayes dapat dinotasikan pada persamaan berikut.

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)} \quad (2.3)$$

Naive Bayes Classifier merupakan algoritma yang digunakan untuk mencari nilai probabilitas tertinggi untuk mengklasifikasi data uji pada kategori yang paling tepat (Feldman & Sanger, 2007). Metode *Naive Bayes Classification* merupakan salah satu metode yang dapat mengklasifikasikan teks. Kelebihan NBC adalah algoritmanya sederhana tetapi memiliki akurasi yang tinggi. Terdapat dua tahap dalam klasifikasi *tweet*. Tahap pertama adalah pelatihan terhadap *tweet* yang telah diketahui kategorinya. Sedangkan tahap kedua adalah proses klasifikasi tweet yang belum diketahui kategorinya (Falahah & Nur, 2015 dalam). Dalam algoritma NBC setiap dokumen direpresentasikan dengan pasangan atribut "$a_1, a_2, \dots, a_n$ dimana a_1 adalah kata pertama, a_2 adalah kata kedua dan seterusnya. Sedangkan V adalah himpunan kategori *tweet*.

Pada saat klasifikasi algoritma akan mencari pro- babilitas tertinggi dari semua kategori dokumen yang diujikan (V_{MAP}). Adapun persamaan V_{MAP}adalah sebagai berikut.

$$V_{MAP} = \arg\max P(V_j) \quad {}_i P(a_i|v_j) \quad (2.4)$$

Nilai (V_j) dihitung pada saat *training*, didapat dengan rumus sebagai berikut :

$$P(V_j) = \frac{|doc_j|}{|training|} \quad (2.5)$$

dimana |*doc j*| merupakan jumlah tweet yang memiliki kategori j dalam training. Sedangkan |*training*| merupakan jumlah *tweet* dalam contoh yang digunakan untuk *training*. Untuk setiap probabilitas kata $\boldsymbol{a_i}$ untuk setiap kategori P($\boldsymbol{a_i}|\boldsymbol{v_j}$), dihitung pada saat training.

$$\boldsymbol{P\ a_i\ v_j\ } = \frac{\boldsymbol{n_i+1}}{\boldsymbol{|n+kosakata|}} \qquad (2.6)$$

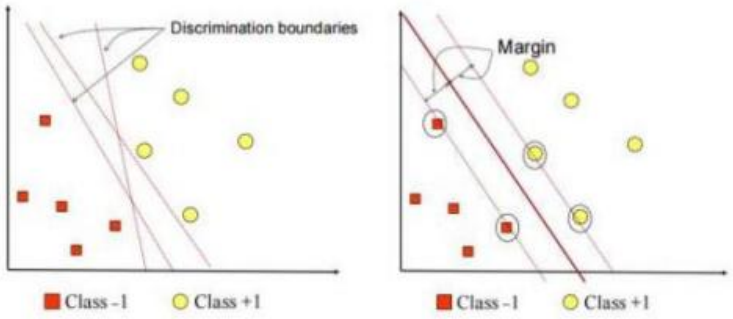
dimana ni adalah jumlah kemunculan kata ai dalam *tweet* yang berkategori $\boldsymbol{v_j}$, sedangkan n adalah banyaknya seluruh kata dalam *tweet* dengan kategori $\boldsymbol{v_j}$ dan |kosakata| adalah banyaknya kata dalam data *training*.

2.6 Support Vector Machine (SVM)

Support Vector Machine (SVM) diperkenalkan oleh Vapnik pada tahun 1992 sebagai suatu teknik klasifikasi yang efisien untuk masalah nonlinear. *Support Vector Machine* (SVM) juga dikenal sebagai teknik pembelajaran mesin (*machine learning*) paling mutakhir setelah pembelajaran mesin sebelumnya yang dikenal sebagai *Neural Network* (NN). Konsep SVM dapat dijelaskan secara sederhana sebagai usaha mencari *hyperplane* terbaik yang berfungsi sebagai pemisah dua buah kelas pada input space. SVM berusaha menemukan fungsi pemisah (*hyperplane*) dengan memaksimalkan jarak antar kelas, dapat dilihat pada gambar 2.1. Dengan cara ini, SVM dapat menjamin kemampuan generalisasi yang tinggi untuk data-data yang akan datang (Suyanto, 2017). *Support Vector Machine* dapat digunakan untuk klasifikasi yang diterapkan pada deteksi tulisan tangan, pengenalan objek, identitas suara dan lain-lain (Ulwan, 2016).

2.6.1 SVM Pada Data Terpisah Secara Linear

SVM pada data terpisah secara linear adalah penerapan metode SVM pada data yang dapat dipisahkan secara linear. Misalkan $\{\boldsymbol{x_1}, \ldots, \boldsymbol{x_n}\}$ adalah dataset dan $\boldsymbol{y_i}$ = {+1, -1} adalah label kategori untuk *dataset*.



Gambar 2.1 Ilustrasi SVM Menemukan Hyperlane Terbaik
Sumber : (Oktarialdi, 2014)

Gambar Ilustrasi SVM menemukan hyperplane terbaik yang memisahkan dua kelas -1 dan +1. Alternatif Bidang Pemisah (kiri) dan Bidang Pemisah Terbaik dengan Margin (m) Terbesar (kanan).

Pada gambar di atas pattern dibagi menjadi dua kelas, yaitu : positif (dinotasikan +1) dan negatif (dinotasikan -1). Pattern yang negatif diberi simbol kotak warna merah dan patern yang positif diberi simbol lingkaran warna kuning. Pada gambar di atas terdapat garis yang memisahkan kedua kelompok, garis tersebut disebut *Hyperlane*. *Hyperlane* terbaik antara kedua kelompok diperoleh dari mengukur margin *hyperlane* tersebut dan mencari titik maksimlanya. Margin adalah jarak antara hyperplane tersebut dengan data terdekat dari masing-masing kelas. *Subset data training* set yang paling dekat ini disebut sebagai *support vector*. Pada gambar 2.1 yang sebelah kanan adalah gambar dengan *hyperlane* terbaik. *Hyperlane* yang terbaik adalah *Hyperlane* terbaik adalah hyperlane yang berada di tengah-tengah kedua kelompok. Titik kotak dan lingkaran yang berada dalam lingkaran hitam disebut *support vector*. Upaya mencari lokasi *hyperplane* optimal ini merupakan inti dari proses pembelajaran pada *Support Vector Machine*. Data yang tersedia dinotasikan sebagai $\boldsymbol{x \epsilon R^d}$ sedangkan label masing-masing kelas dinotasikan $\boldsymbol{y_i} \ni \boldsymbol{-1, +1}$ untuk *i*= 1,2,3, … , n.

Diasumsikan kedua kelas tersebut terpisah secara sempurna oleh *hyperlane* berdimensi d. Persamaan *hyperlane* dapat ditulis sebagai berikut :

$$\boldsymbol{w} . \boldsymbol{x_i} + \boldsymbol{b} = \mathbf{0} \tag{2.7}$$

Pattern $\boldsymbol{x_i}$ yang termasuk kelas -1 (sampel negatif) misal $\boldsymbol{x_a}$ dapat dirumuskan sebagai *pattern* yang memenuhi pertidaksamaan :

$$\boldsymbol{w} . \boldsymbol{x_a} + \boldsymbol{b} = \mathbf{-1} \tag{2.8}$$

Pattern yang termasukkelas +1 (sampel positif) misal $\boldsymbol{x_b}$

$$\boldsymbol{w} . \boldsymbol{x_b} + \boldsymbol{b} = \mathbf{+1} \tag{2.9}$$

Klasifikasi kelas data pada SVM pada persamaan (2.8) dan (2.9) dapat digabungkan dengan notasi

$$\boldsymbol{y_i} \ \boldsymbol{w} . \boldsymbol{x_i} + \boldsymbol{b} \ \geq \mathbf{1}, \boldsymbol{i} = \mathbf{1}, \mathbf{2}, \ldots, \boldsymbol{N} \tag{2.10}$$

Margin optimal diperoleh dengan cara memaksimalkan nilai jarak antara jarak dan titik terdekatnya, yaitu $\frac{1}{w}$. Dalam kasus ini dapat dirumuskan sebagai *Quadratic Programming* (QP) *problem*, yaitu mencari titik minimal persamaan , dengan memperlihatkan *constraint* persamaan

$$\boldsymbol{min_w} = \frac{\mathbf{1}}{\mathbf{2}} \ \boldsymbol{w}^{\ 2} \tag{2.11}$$

dengan $\boldsymbol{w}$ adalah vektor normal. Selanjutnya, menggunakan metode *Lagrange multiplier* dengan persamaan sebagai berikut :

$$\boldsymbol{Ld} = \ \sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{N}} \boldsymbol{\alpha_i} - \frac{\mathbf{1}}{\mathbf{2}} \ \sum_{\boldsymbol{i,j}} \boldsymbol{\alpha_i \alpha_j y_i y_j x_i . x_j} \tag{2.12}$$

Persamaan Ld digunakan untum mencari nilai-nilai $\boldsymbol{\alpha_i}$ (*support vector*) dengan membuat Ld optimum. Ld optimum didapat dengan cara mencari turunan parsial Ld terhadap

$\boldsymbol{\alpha}$. Langkah selanjutnya adalah mencari nilai *w* dan *b*. *Hyperlane* (batas keputusan) yang dinotasikan *w* diperoleh dari persamaan berikut:

$$\boldsymbol{w} = \sum_{i=1}^{N} \boldsymbol{\alpha_i y_i x_i} \text{ dan } \boldsymbol{b = 1 - w^T x} \quad (2.13)$$

2.6.2 SVM Pada Data Tidak Terpisah Secara Linear dengan Metode Kernel

Pada kasus data yang tidak dapat dipisahkan secara linear maka SVM harus dimodifikasi agar dapat diproses dan data bisa diinterpretasikan. Permasalahan ini dapat diselesaikan dengan menggunakan metode Kernel . Metode Kernel bekerja dengan mentransformasikan data ke dalam dimensi ruang fitur sehingga dapat dipisahkan secara linear pada *feature space*. *Feature space* memiliki dimensi lebih tinggi dari pada vektor input (*input space*). Komputasi pada *feature space* sangat besar sehingga ada kemungkinan *feature space* memiliki jumlah *feature* tak terhingga.

Jika terdapat fungsi *kernel* K sehingga $\boldsymbol{K(x_i, x_d) = \emptyset(x_i)\ \emptyset(x_d)}$, maka fungsi $\emptyset(\boldsymbol{x_k})$ tidak perlu diketahui. Sehingga diperoleh fungsi berikut :

$$\boldsymbol{f(x_d) = \sum_{i=1}^{ns} \propto_i y_i K(x_i, x_d) + b \quad x_i = support\ vector} \quad (2.14)$$

dimana :

$\propto_{\boldsymbol{i}}$= Koefisien *lagrange*

$\boldsymbol{x_i}$ = Support vector

$\boldsymbol{y_i}$ = Kelas data

$\boldsymbol{ns}$ = Jumlah Support vektor

$\boldsymbol{x_d}$ = Data yang diklasifikasikan

Syarat fungsi untuk bisa menjadi fungsi *kernel* adalah memenuhi teorema *Mercer* yang menyatakan bahwa matriks kernel harus bersifat positif *semidefinite*. Fungsi kernel yang umum digunakan adalah :

a. *Polynomial Kernel*

Kernel trick polynomial diformulasikan untuk digunakan dalam menyelesaikan masalah klasifikasi, dimana dataset pelatihan yang digunakan sudah normal. Berikut persamaan:

$$\boldsymbol{K}\ \boldsymbol{x},\boldsymbol{y}\ = (\boldsymbol{x}.\boldsymbol{y}+\boldsymbol{c})^{\boldsymbol{d}} \tag{2.15}$$

b. *Radial Basis Function (RBF)*

Kernel Gaussian ini merupakan *kernel* yang paling banyak digunakan dalam penyelesaian masalah klasifikasi untuk dataset yang tidak terpisah secara *linear*, dikarenakan pada *kernel* ini memiliki akurasi prediksi yang sangat baik. Persamaan yang dimiliki sebagai berikut:

$$\boldsymbol{K}\ \boldsymbol{x},\boldsymbol{y}\ = \exp\ \frac{-\ \boldsymbol{x}-\boldsymbol{y}\ ^2}{2.\sigma^2} \tag{2.16}$$

Pada penelitian ini digunakan Kernel RBF karena berdasarkan penelitian terdahulu menghasilkan nilai akurasi yang lebih tinggi dibandingkan Kernel lainnya.

c. *Sigmoig Kernel*

Sigmoid merupakan *kernel trick* SVM yang merupakan pengembangan dari jaringan saraf tiruan, dimana *kernel* ini dinyatakan dengan persamaan berikut:

$$\boldsymbol{K}\ \boldsymbol{x},\boldsymbol{y}\ = \tan \boldsymbol{h}\ (\boldsymbol{\sigma}\ \boldsymbol{x}.\boldsymbol{y}\ + \boldsymbol{c}) \tag{2.17}$$

2.7 Pengukuran Performa

Pengukuran performa dilakukan untuk menentukan bagus/tidaknya sebuah klasifikasi. Terdapat beberapa cara untuk mengukur performa diantaranya akurasi, *recall*, dan *precission*.

Dalam mengukur ketepatan klasifikasi, perlu diketahui jumlah pada setiap kelas prediksi dan kelas aktual yang terdiri dari *TP* (*True Positif*) yaitu jumlah *tweet* bersentimen positif yang tepat terprediksi dalam kelas positif, *TN* (*True Negatif*) yaitu *tweet* bersentimen negatif yang tepat terprediksi dalam kelas negatif,. *FP (False Positif)* yaitu *tweet* bersentimen negatif yang terprediksi dalam kelas positif, dan *FN (False Negatif)* yaitu *tweet* bersentimen positif yang terprediksi dalam kelas. Berikut merupakan *confusion matrix* yang memuat keempat nilai tersebut.

Tabel 2.1 *Confussion Matrix*

Kelas Aktual	**Kelas Prediksi**	
	Positif	Negatif
Positif	Trur *Positif* (TP)	False *Negatif* (FN)
Negatif	False *Positif* (FP)	True *Negatif* (TN)

Confusion matrix sangat berguna untuk menganalisis kualitas *classifier* dalam mengenali *term* dari kelas yang ada. TP dan TN menyatakan bahwa *classifier* mengenali *term* dengan benar, artinya *term* positif dikenali sebagai positif dan *term* negatif dikenali sebagai *negatif*. Sebaliknya, FP dan FN menyatakan bahwa *classifier* salah dalam mengenali *term*, *term negatif* dikenali sebagai positif dan *term* negatif dikenali sebagai positif. Terdapat beberapa rumus umum yang dapat digunakan untuk menghitung performa klasifikasi. Hasil dari nilai *accuracy, precision, recall, F Measure* biasa ditampilkan dalam persentase.

a. Accuracy

Accuracy adalah jumlah proporsi prediksi yang benar. Adapun rumus penghitungan akurasi dapat dilihat pada persaman 2.9. (Lim dkk., 2006)

$$\boldsymbol{accuracy = \frac{TP+TN}{All}\ X\ 100\%} \quad (2.18)$$

b. Precision

Precision adalah proporsi jumlah dokumen teks yang relevan terkenali diantara semua dokumen teks yang terpilih oleh sistem. Rumus *precision* dapat dilihat pada persamaan 2.19. (Lim dkk., 2006)

$$\boldsymbol{precision} = \frac{\boldsymbol{TP}}{(\boldsymbol{TP}+\boldsymbol{FP})} \quad (2.19)$$

c. Recall

Recall adalah proporsi jumlah dokumen teks yang relevan terkenali diantara semua dokumen teks relevan yang ada pada koleksi. Rumus recall dapat dilihat pada persamaan 2.20. (Lim dkk., 2006)

$$\boldsymbol{recall} = \frac{\boldsymbol{TP}}{(\boldsymbol{TP}+\boldsymbol{FN})} \quad (2.20)$$

d. F Measure

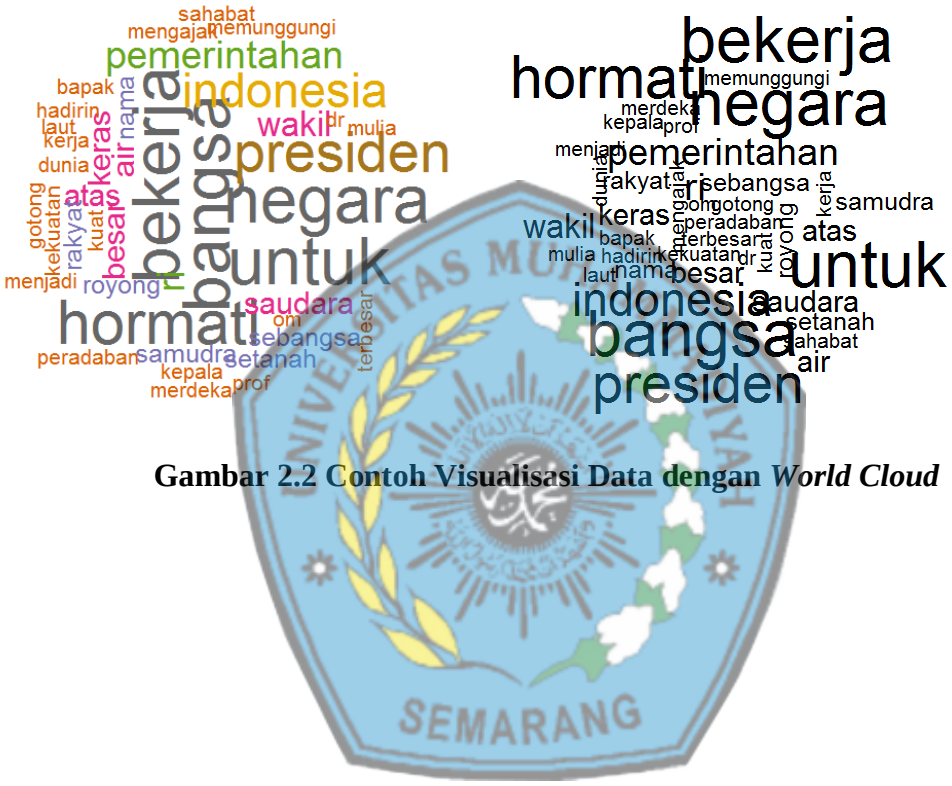
F-measure merupakan kompromi dari recall dan precision untuk mengukur kinerja keseluruhan pengklasifikasi. Berikut merupakan cara perhitungan f-measure. (Hotho dkk, 2005)

$$\boldsymbol{F\ Measure} = \frac{\boldsymbol{2\ x\ recall\ x\ precision}}{(\boldsymbol{recall}+\boldsymbol{precision})} \quad (2.21)$$

2.8 Word cloud

Word cloud (disebut juga *text cloud* atau *tag cloud*) adalah representasi visual baru dari data teks, biasanya digunakan untuk menvisualisasikan teks yang muncul dari sebuah web / metadata. *Tag* adalah kata tunggal , dan setiap kata ditampilkan dengan ukuran, warna, dan font berbeda. Frekuensi dari kata yang sering muncul ditunjukkan melalui ukuran huruf kata tersebut.

Semakin besar ukuran kata menunjukkan semakin besar frekuensi kata tersebut muncul dalam dokumen. Format ini berguna untuk dengan cepat memahami kata yng paling menonjol untuk menentukan keunggulan relatifnya. Dengan menggunakan *word cloud,* gambaran frekuensi kata-kata dapat ditampilkan dalam bentuk yang menarik namun informatif. (Setiabudi, Nur Andi. 2015). Berikut contoh word cloud :



Gambar 2.2 Contoh Visualisasi Data dengan *World Cloud*