BAB II
LANDASAN TEORI

2.1 Twitter

Twitter adalah layanan *microblogging* yang dirilis secara resmi pada 13 Juli 2006 (Mustafa, 2013). Twitter merupakan salah satu jejaring sosial yang paling mudah digunakan, karena hanya memerlukan waktu yang singkat tetapi informasi yang disampaikan dapat langsung menyebar secara luas (Setyani, 2013). Menurut Twitter (2013), Twitter adalah sebuah situs web yang memiliki dan dioperasikan oleh Twitter Inc, yang menawarkan jaringan sosial berupa *microblog* sehingga memungkinkan penggunanya untuk mengirim dan membaca pesan *Tweets*. *Tweets* merupakan aktifitas utama yang pendek berisikan tulisan berupa teks yang terdiri dari 140 karakter yang ditampilkan pada halaman profil pengguna. Akan tetapi pada tanggal 07 November 2017 bertambah hingga 280 karakter. Semua pengguna dapat mengirim dan menerima *tweets* melalui situs Twitter, aplikasi eksternal yang kompatibel (telepon seluler), atau dengan pesan singkat (SMS) yang tersedia di Negara-negara tertentu, pengguna dapat menulis pesan berdasarkan topic dengan menggunakan tanda # (*hashtag),* sedangkan untuk menyebutkan atau membalas pesan dari pengguna lain bisa menggunakan tanda @ (Twitter, 2013). Menurut (Habibi, Setyohadi & Ernawati, 2016),, selama beberapa tahun terakhir, Twitter menjadi sangat popular. Jumlah pengguna Twitter setiap hari adalah lebih dari 65 juta.

2.2 Emosi

Emosi berasal dari bahasa Latin yaitu Emovere yang memiliki arti bergerak menjauh. Arti dari kata ini memiliki maksud kecenderungan bertindak merupakan hal yang mutlak dalam emosi. Emosi manusia merupakan peranan penting dalam komunikasi (Winarsih, 2016). Komunikasi dapat dilakukan dari informasi verbal dan non-verbal. Verbal dapat berupa bahasa yang ditulis dengan tulisan yang diperoleh dari kata, kalimat, paragraf, dan sebagainya. Sedangkan non-verbal adalah sebuah isyarat tubuh (Destuardi, 2009).

Menurut (Kleinginna dalam Morgan dkk, 1968) menyatakan bahwa emosi itu adalah mengatakan sesuatu tentang apa yang kita rasakan ketika kita sedang emosional, menyebut secara psikologis atau secara badaniah, dasar dari perasaan emosional, emosi berpengaruh terhadap persepsi, pikiran, dan perilaku, menjelaskan dorongan atau motivasional, perlengkapan dari emosi-emosi tertentu seperti takut dan marah, dan menunjuk bagaimana emosi diekspresikan ke dalam bahasa, ekspresi wajah, dan *gesture* (bahasa tubuh).

Sedangkan menurut Daniel Goleman (2002) yang dikutip oleh Nugroho (2016) mengatakan bahwa emosi merujuk pada suatu perasaan dan pikiran yang khas, suatu keadaan biologis dan psikologis, dan serangkaian kecenderungan untuk bertindak. Emosi terjadi karena adanya stimulus atau sebuah peristiwa yang bisa netral, positif, dan negatif yang melibatkan faktor psikologis maupun faktor fisiologis.

2.3 *Text Mining*

Menurut Harlian (2009) *text mining* memiliki definisi menambang data yang berupa teks dimana sumber data biasanya didapatkan dari dokumen, yang tujuannya adalah mencari kata-kata yang dapat mewakili isi dari dokumen sehingga dapat dilakukan analisa keterhubungan antar dokumen. Menurut (Prilianti dan Wijaya, 2014) *text mining* adalah salah satu langkah dari analisis teks yang dilakukan secara otomatis oleh komputer untuk menggali informasi yang berkualitas dari suatu rangkaian teks yang terangkum dalam sebuah dokumen. *Text mining* mengarah ke proses pengambilan informasi berkualitas tinggi dari teks (Saraswati, 2011). *Text mining* merupakan teknik yang digunakan untuk menangani masalah klasifikasi, *clustering*, *information extraction* dan *information retrival* (Berry & Kogan, 2010). Menurut Han & Kamber (2006) pada dasarnya proses kerja dari *text mining* banyak mengadopsi 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. Kunci dari proses ini adalah menggabungkan informasi yang berhasil diekstraksi dari berbagai sumber (Hearst, 2003).

Tahap-tahap *text mining* secara umum adalah *text preprocessing* dan *feature selection* (Feldman, 2007). Dikarenakan kebanyakan informasi (perkiraan umum mengatakan lebih dari 80%) saat ini disimpan sebagai teks, *text mining* diyakini memiliki potensi nilai komersial tinggi (Bridge, 2011).

2.4 Klasifikasi Teks

Klasifikasi merupakan suatu proses untuk menemukan model yang menjelaskan atau mampu membedakan kelas data yang bertujuan untuk memperkirakan kelas yang tidak diketahui dari suatu objek. Menurut (Feldman, 2004) klasifikasi merupakan salah satu cara untuk dapat mengorganisasikan dokumen. Dokumen yang memiliki isi yang sama akan dikelompokkan ke dalam kategori yang sama. Dengan kata lain, orang-orang yang melakukan pencarian informasi dapat mudah melewatkan kategori yang tidak relevan dengan informasi yang dicari atau yang tidak menarik perhatian.

Pada tahapan klasifikasi teks ini terdapat dua proses yang harus dilakukan yaitu :

1. Proses *training*

   Proses *training* digunakan pada *training set* yang telah diketahui label-labelnya untuk membangun model.

2. Proses *testing*

   Proses ini dilakukan untuk mengetahui keakuratan model yang akan dibangun pada proses *training*, maka digunakanlah data *testing set* untuk memprediksi label-labelnya (Munawarah, Soesanto & Faisal, 2016).

Pendekatan umum yang digunakan pada klasifikasi adalah *training set* yang berisikan *record* yang mempunyai label kelas yang diketahui harus sudah tersedia. Tujuan dari *training set* ini untuk membangun model klasifikasi yang akan diaplikasikan ke *test set,* yang berisikan *record-record* yang label kelasnya tidak diketahui.

2.5 *Text Prepprocessing*

Text mining memiliki tahap awal yaitu *text prepocessing*. *Text preprocessing* ini memiliki tujuan untuk mempersiapkan teks menjadi data yang akan diolah pada tahapan selanjutnya. Tujuan selanjutnya adalah mengubah data

teks dari yang tidak terstruktur menjadi data terstruktur. Pada tahap ini dilakukan guna untuk menghilangkan *noise* agar dapat menyeragamkan kata dan mampu mengurangi *volume* kata.

Pada tahapan *text perpocessing* memiliki beberapa proses yaitu *cleaning, case folding, tokenizing* dan *filtering,stemming*

Gambar 2.1 Tahap *Preprocessing* (Leman & Andesa, 2015)

1. *Case Folding* merupakan proses pengubahan karakter huruf menjadi seragam sehingga menjadi huruf kecil, juga menghilangkan tanda baca dan angka
2. *Cleaning* merupakan proses penghilangan kata-kata yang tidak diperlukan dari *tweet* dan karakter-karakter untuk mengurangi *noise* pada proses pengklasifikasian. Kata yang perlu dihilangkan seperti *username* dan *mention*, dan sebagainya.
3. *Token* merupakan proses pemecahan sekumpulan karakter atau kalimat dalam suatu teks pada *tweet* ke dalam satuan kata dan menentukan struktur sintaksis dari tiap kata tersebut.
4. Normalisasi merupakan proses koreksi ejaan kata terhadap kata yang tidak standar, misalnya “tiduur” menjadi “tidur”. Terhadap singkatan yang tidak baku seperti “tdk” menjadi “tidak”.
5. *Filtering* merupakan tahap untuk mengambil kata-kata penting dari hasil token. Selain itu *filtering* juga merupakan proses memperbaiki kata-kata yang dibutuhkan namun tidak sesuai misalnya “baguuuuusssss” diubah menjadi “bagus”. Dengan kata lain mengubah bahasa Indonesia yang tidak baku menjadi kata baku yang sesuai dengan Kamus Besar Bahasa Indonesia. *Filtering* juga mampu membuang kata yang tidak berpengaruh dalam proses klasifikasi yang biasa disebut dengan *stopword*. Mislanya membuang kata gaul seperti “gw”, kata hubung seperti “di”, “ke”, “yang”, “dari”,”dan”, petunujuk waktu dan kata tanya.

6. *Stemming* merupakan tahap mencari kata dasar dengan cara menghilangkan imbuhan awakan dan akhiran.

2.6 Pembobotan Dan Seleksi Fitur

Tahap selanjutnya setelah melakukan *text preprocessing* adalah melakukan pemilihan fitur agar mendapatkan hasil klasifikasi yang lebih maksimal. Pada tahap ini merupakan tahap penting dalam klasifikasi teks. Pemilihan fitur dapat dilakukan dengan melakukan pengamatan terhadap distribusi fekuensi kemunculan kata dan jumlah *feature* (Trilaksono, 2015).

Seleksi fitur merupakan salah satu teknik terpenting. Teknik ini dapat mengurangi peningkatan ukuran *term* pada saat proses *training*, dan dapat mengurangi *noise* dengan menghapus fitur yang tidak relevan, agar mampu meningkatkan akurasi klasifikasi (Naufal, 2015). Dokumen dapat dinyatakan dalam model ruang vektor dengan diwakili dengan sebuah vektor dari *keyword* yang di ekstrak. Vektor dokumen yang telah didapat, kemudian dilakukan pembobotan yang mampu mewakili seberapa pentingnya *keyword* tersebut dalam dokumen.

Pemilihan fitur memiliki fungsi yaitu untuk mendapatkan nilai *trheshold* parameter dalam klasifikasi SVM. Dilakukan dengan pengamatan terhadap distribusi frekuensi kemunculan kata dan jumlah *feature* (Pratama, Trilaksono, 2015). Pendekatan seleksi fitur yang digunakan dalam penelitian ini yaitu *Term Frequency Inverse Document Frequensy* (TF-IDF).

2.6.1 *Term Frequency Inverse Document Frequency*

TF-IDF adalah jenis pembobotan yang sering digunakan dalam *Text Mining*. TF-IDF merupakan metode pembobotan *term* dengan menggunakan *term frequency* (jumlah *term* yang terdapat pada tiap dokumen) serta *inverse document frequency* (invers jumlah dokumen yang memuat suatu *term*) (Somantri, Wiyono & Dairoh, 2016). Pembobotan ini adalah suatu pengukuran statistik untuk mengukur seberapa penting sebuah kata dalam kumpulan dokumen. Tingkat kepentingan meningkat apabila ketika sebuah kata muncul beberapa kali dalam sebuah dokumen tetapi diimbangi dengan frekeunsi kemunculan kata tersebut dalam kumpulan dokumen.

Yang perlu dihitung lebih dahulu adalah *Term Frekuensi* yaitu frekuensi kemunculan kata di tiap dokumen. Kemudian hitung *Inverse Document Frequensi* yaitu nilai bobot suatu kata dihitung dari seringnya suatu kata muncul di beberapa dokumen. Semakin sering suatu kata muncul di banyak dokumen. Semakin sering suatu kata muncul di banyak dokumen, maka nilai IDFnya akan kecil. Rumus dalam menentukan pembobot dengan TF-IDF adalah sebagai berikut:

$$w_{ij} = tf \times idf \qquad (2.1)$$

$$idf = \log\left(\frac{N}{dfi}\right) \qquad (2.2)$$

Dengan : i = 1,2,....,p (Jumlah variabel)

j = 1,2,....,N (Jumlah data)

Dimana *wij* adalah bobot dari kata *i* pada artikel ke *j*, *N* merupakan jumlah seluruh dokumen, *tfij* adalah jumlah kemunculan kata i pada dokumen *j*, *dfj* adalah jumlah artikel *j* yang mengandung kata *i*. TF-IDF dilakukan agar data dapat dianalisis dengan menggunakan *support vector machine* (Asiyah and Fithriasari 2016).

2.7 *Stemming*

Tweet yang akan di klasifikasikan adalah tweet berbahasa Indonesia, maka algoritma *stemming* yang akan digunakan adalah algoritma *Enhanced Confix Stripping* (ECS). *Stemming* adalah proses mengurangi varian morphologi menjadi dalam satu bentuk kata dasar (*root*) (B.Comp. Sc, 2007). Algoritma *stemming* kata pada bahasa Indonesia dengan performa yang paling baik saat ini (memiliki jenis kesalahan *stemming* yang paling sedikit) adalah Algoritma *Enhanced Confix Stripping* (ECS) (Arifin, Mahendra and Ciptaningtyas 2009) *Stemming* merupakan sebuah teknik untuk mentransformasikan kata-kata dalam sebuah dokumen teks menjadi bentuk kata dasar (Agusta & Ledy, 2009). *Stemming* pada bahasa Indonesia lebih sulit dilakukan karena pada bahsa

Indonesia terdapat imbuhan awalan (*prefixes*), sisipam (*infixes*), akhiran (*suffixes*), dan kombinasi awalan dan akhiran (*confixes*) (Sagala, Lydia, & Rahmat, 2015).

Algoritma *Enhanched Confix Stripping* ini mempunyai tahapan proses sebagai berikut (Prasetyo and Susanti 2012) :

1. Kata yang hendak di *stemming* dicari terlebih dahulu pada kamus. Jika ditemukan, berarti kata tersebut adalah kata dasar, jika tidak maka langkah 2 dilakukan.
2. Cek *rule precedence*. Apabila suatu kata memiliki pasangan awalan-akhiran "be-lah", "be-an", "me-i", "di-i", "pe-i", atau "te-i" maka langkah stemming selanjutnya adalah (5, 3, 4, 6). Apabila kata tidak memiliki pasangan awalan- akhiran tersebut, langkah stemming berjalan normal (3, 4, 5, 6).
3. Hilangkan *inflectional particle* P ("-lah", "-kah", "-tah", "-pun") dan kata ganti kepunyaan atau possessive pronoun PP ("-ku", "-mu", "-nya").
4. Hilangkan *Derivation Suffixes* DS ("-i", "-kan", atau "-an").
5. Hilangkan *Derivational Prefixes* DP {"di-","ke-","se-","me-","be-","pe", "te-"}.

a. Identifikasikan tipe awalan dan hilangkan. Awalan ada dua tipe:
   1. Standar: "di-", "ke-", "se-" yang dapat langsung dihilangkan dari kata.
   2. Kompleks: "me-", "be-", "pe", "te-" adalah tipe-tipe awalan yang dapat bermorfologi sesuai kata dasar yang mengikutinya.

b. Cari kata yang telah dihilangkan awalannya ini di dalam kamus. Apabila tidak ditemukan, maka langkah 5 diulangi kembali. Apabila ditemukan, maka keseluruhan proses dihentikan.

6. Jika semua gagal, maka masukan kata yang diuji pada algoritma ini dianggap sebagai kata dasar.

Merujuk pada penelitian yang dilakukan oleh (Tahitoe, Purwitasari, 2010) yang dikutip oleh (Sagala, Lydia & Rahmat, 2015), tahapan kerja algoritma *Enhanced Confix Stripping Stemmer* adalah sebagai berikut:

Tabel 2.1 Aturan Pemenggalan Awalan Algoritma *Enhanced Confix Stripping Stremmer* (Mahendra, 2008).

Aturan	Format Kata	Pemenggalan
1	berV…	ber-V… | be-r-V…
2	berCAP…	ber-CAP… dimana C!=„r“ & P!=“er“
3	berCAerV…	ber-CAerV… dimana C!=“r“
4	belajar	bel-ajar
5	beC_1erC_2…	be-C_1erC_2… dimana C_1!={„r“ | „l“}
6	terV…	ter-V… | te-rV…
7	terCerV…	ter-CerV… dimana C!=“r“
8	terCP…	ter-CP… dimana C!=“r“ dan P!=“er“
9	teC_1erC_2…	te-C_1erC_2… dimana C_1!=“r“
10	me{l|r|w|y}V…	me-{l|r|w|y}V…
11	mem{b|f|v}…	mem-{b|f|v}…
12	mempe…	mem-pe…
13	mem{rV|V}…	me-m{rV|V}… | me-p{rV|V}…
14	men{c|d|j|s|z}…	men-{c|d|j|s|z}…
15	menV…	me-nV… | me-tV…
16	meng{g|h|q|k}…	meng-{g|h|q|k}…
17	mengV…	meng-V… | meng-kV… | (mengV-… jika V=“e“)
18	menyV…	meny-sV…
19	mempA…	mem-pA… dimana A!=“e“
20	pe{w|y}V…	pe-{w|y}V…
21	perV…	per-V… | pe-rV…
23	perCAP…	per-CAP… dimana C!=“r“ dan P!=“er“
24	perCAerV…	per-CAerV… dimana C!=“r“
25	pem{b|f|V}…	pem-{b|f|V}…
26	pem{rV|V}…	pe-m{rV|V}… | pe-p{rV|V}…
27	pen{c|d|j|z}…	pen-{c|d|j|z}…

Aturan	Format Kata	Pemenggalan
28	penV…	pe-nV… | pe-tV…
29	pengC…	peng-C…
30	pengV…	peng-V… | peng-kV… | (pengV-… jika V="e")
31	penyV…	peny-sV…
32	pelV…	pe-lV… kecuali "pelajar" yang menghasilkan "ajar"
33	peCerV…	per-erV… dimana C!={r|w|y|l|m|n}…
34	peCP…	pe-CP… dimana C!={r|w|y|l|m|n} dan P!="er"
35	terC_1erC_2…	ter-C_1erC_2… dimana C_1!="r"
36	peC_1erC_2…	pe-C_1erC_2… dimana C_1!={r|w|y|l|m|n}

2.8 *Support Vector Machine*

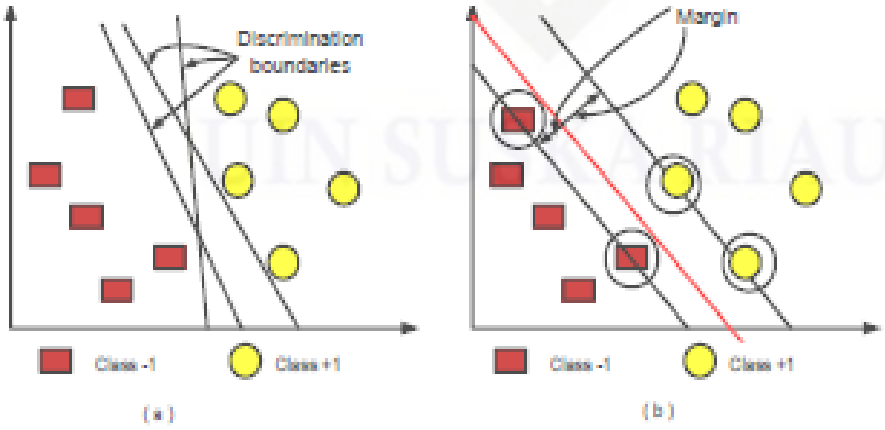
Support Vector Machine dikembangkan oleh Boser, Guyon, Vapnik, kemudian dipresentasikan pertama kali pada tahun 1992 di Annual Workshop on Computional Learning Theory. *Support Vector Machine* (SVM) adalah sistem pembelajaran yang menggunakan ruang hipotesis berupa fungsi-fingsi linier dalam sebuah ruang fitur berdimensi tinggi, dilatih dengan algoritma pembelajaran yang didasarkan dengan teori optimasi dengan mengimplementasikan *learning bias* yang berasal dari teori pembelajaran statistik (Cristianini dkk 2000). Menurut Santosa (2007) *Support Vector Machine* adalah suatu teknik untuk melakukan prediksi, baik dalam kasus klasifikasi maupun regresi. *Support Vector Machine* adalah metode klasifikasi yang bekerja dengan cara melakukan pencarian *hyperplane* dengan margin terbesar. *Hyperplane* ialah garis batas pemisah data antar kelas dan margin sebagai jarak antar *hyperplane* dengan data terdekat pada tiap-tiap kelas. Adapun data terdekat dengan *hyperplane* pada masing-masing kelas inilah yang disebut dengan *support vector* (J. Yunliang, et al., 2010). SVM banyak digunakan untuk klasifikasi data berupa text dengan menghasilkan tingkat akurasi yang lebih baik. Tetapi dalam hal ini untuk proses klasifikasi dokumen, seringkali ditemukan hasil yang kurang baik

dikarenakan jumlah data dokumen yang besar dan bervariasi (Trivedi S, Pardos A & Sar N, 2008) dikutip oleh (Somantri, Wiyono & Dairoh, 2016). SVM memiliki prinsip dasar linier *classifier* yaitu kasus klasifikasi yang secara linier dapat dipisahkan, akan tetapi SVM telah dikembangkna agar dapat bekerja pada permasalahan non-linier dengan memasukkan konsep kernel pada ruang kerja yang meiliki dimensi yang tinggi (Octaviani, Wilandari & Isprianti, 2014). SVM memiliki kelebihan diantaranya adalah dalam menentukan jarak menggunakan *support vector* sehingga proses komputasi menjadi cepat (Vapnik, 1995).

Menurut Saraswati (2011) metode SVM mengambil himpunan data *input* dan memprediksi setiap masukan yang diberikan, dimana masukan adalah anggota dari salah satu kelas dari dua kelas yang ada, hal ini membuat sebuah SVM sebagai penggolong non probabilistik linier biner. Sebab SVM adalah *classifier* maka diberikan suatu himpunan pelatihan yang mana data-data tersebut ditandai sebagai milik salah satu dari dua kategori. Algoritma pelatihan SVM membangun sebuah model yang memprediksi apakah data yang baru jatuh kedalam suatu kategori atau yang lain.

Didalam ruang kerja berdimensi tinggi, akan dicari *hyperlane* yang dapat memaksimalkan margin antara kelas data. Menurut Nugroho (2008) *hyperplane* klasifikasi linier SVM dinotasikan sebagai berikut :

$$f(x) = w^T x + b \qquad (2.2)$$



Gambar 2.2 Konsep *hyperplane* SVM yang Memisahkan Kedua *Class* -1 dan +1 (Nugroho, 2008)

Pada Gambar 2.2-a memeperlihatkan beberapa *pattern* yang merupakan anggota dari dua buah *class* yaitu +1 dan -1. *Pattern* yang berada pada kelas -1 disimbolkan dengan warna merah, dan *pattern* pada *class* +1 disimbolkan dengan warna kuning. Masalah klasifikasi dapat diterjemahkan dengan menemukan garis (*hyperplane*) yang memisahkan antara jedua kelomok tersebut (Cortes & Vapnik, 1995). Garis pemisah (*discrimination boundaries*) yang ditunjukkan pada Gambar 2.2-a adalah salah satu alternatif garis pemisah yang memisahkan kedua kelas.

Hyperplane pemisah terbaik antara kedua kelas dapat ditemukan dengan mengukur *hyperplane* dan mencari titik maksimalnya. Margin merupakan jarak antara *hyperplane* dengan *pattern* terdekat dari masing-masing kelas. *Pattern* yang paling dekat ini disebut dengan *support vector* (Naufal, Wahono & Syukur, 2015). Garis solid pada Gambar 2.2-b menunjukkan *hyperplane* terbaik yaitu terletak pada tengah-tengah kedua kelas, dan titik merah dan kuning yang berada dalam lingkaran disebut dengan *support vector*. Usaha untuk mencari *hyperplane* terbaik adalah merupakan inti dari proses pembelajaran pada *support vector machine* (Nugroho, 2008).

Bidang pembatas pertama membatasi kelas pertama sedangkan bidang pembatas kedua membatasi kelas kedua, sehingga diperoleh rumus sebagai berikut :

$$x_i \mathrm{w} + \mathrm{b} \geq 1, y_i = +1 \quad (2.3)$$

$$x_i \mathrm{w} + \mathrm{b} \leq 1, y_i = -1 \quad (2.4)$$

Dimana w adalah normal bidang dan *b* adalah posisi bidang alternatif terhadap pusat koordinat. Nilai margin antara bidang pembatas berdasarkan rumus jarak garis ke titik pusat adalah sebagai berikut :

$$\frac{1-b-(-1-b)}{||w||} = \frac{2}{||w||} \quad (2.5)$$

Untuk mengklasifikasikan data yang tidak mampu dipisahkan secara linier formula SVM harus dimodifikasi karena tidak ditemukan solusi. Dengan demikian formula pencarian bidang pemisah terbaik berubah menjadi sebagai berikut :

$$\min \frac{1}{2}||w||^2 + \mathrm{C}(\sum_{i=1}^{n} \epsilon_1) \quad (2.6)$$

Dengan $y_1(x_i \mathrm{w} + b) \geq 1\text{-}ב_i \epsilon \geq 0$ (2.7)

C merupakan parameter yang menetukan besar kesalahan dalam klasifikasi data dan nilainya ditentukan oleh pengguna. Peran C yaitu meminimalkan kesalahan pelatihan dan mengurangi kompleksitas model. Pemilihan parameter pada algoritma SVM dengan menggunakan metod *grid seacrh* karena sangat handal jika diaplikasikan pada *dataset* yang mempunyai atribut sedikit daripada metode *random search* (Bergstra & Bengio, 2012).

SVM memiliki karakteristik sebagai berikut (Anto,2003) :

1. Secara prinsip SVM adalah *linear classifier*
2. Pattern recognition dilakukan dengan mentransformasian data input space ke ruan yang berdimensi lebih tinggi, dan optimisasi dilakukan pada ruang vektor yang baru tersebut. Hal ini membedakan SVM dari solusi pattern recognition pada umumnya, yang melakukan optimisasi parameter pada ruang hasil transformasi yang berdimensi lebih rendah daripada dimensi input space.
3. Menerapkan srategi *Structural Risk Minimization* (SRM).
4. Prinsip kerja SVM pada dasarnya hanya mampu menangani klasifikasi dua kelas.

Pada dasarnya SVM adalah metode yang digunakan hanya untuk klasifikasi dua kelas (*binary calssification*). Kemudian muncul beberapa metode diusulkan agar SVM mampu menyelesaikan permasalahan klasifikasi *multi-class* dengan cara mengombinasikan beberapa *binary classifier* (J.Z.Liang, 2004). Metode yang diusulkan adalah metode *one-against-one*. Adapun metode *one-against-one* ini akan dikostuksi sejumlah *k(k-1)/2* model klasifikasi SVM dengan masing-masing model yang ada dilatih menggunakan data dari dua kelas yang berbeda. Dengan demikian data pada kelas *i* dan *j* SVM akan menyelesaikan permasalahan klasifikasi biner untuk *multi-class.* Penelitian ini menggunakan metode *one-against-one.*

SVM memiliki kelebihan yaitu sebagai berikut (Anto, 2003) :

1. *Generalisasi*

   Generalisasi didefinisikan sebagai kemampuan suatu metode (SVM, neural network, dsb) untuk mengklasifikasikan suatu pattern, yang tidak termasuk data yang dipakai dalam fase pembelajaran metode itu. Vapnik menjelaskan bahwa generalization error dipengaruhi oleh dua faktor yaitu error terhadap training set, dan faktor yang lain dipengaruhi oleh dimensi VC (Vapnik-Chervokinensis).

2. *Curse of dimensionality*

   Curse of dimensionality didefinisikan sebagai masalah yang dihadapi suatu metode pattern recognition dalam mengestimasikan parameter dikarenakan jumlah sampel data yang relatif sedikit dibandingkan dimensional ruang vektor data tersebut. Semakin tinggi dimensi dari ruang vektor informasi yang diolah, membawa konsekuensi dibutuhkannya jumlah data dalam proses pembelajaran.

3. *Feasibility*

   SVM mampu diimplementasikan dengan mudah. Karena proses penentuan *support vector* dapat dirumuskan dalan QP *problem*.

Selain itu, SVM memiliki kekurangan yaitu (Anto, 2003) sulit digunakan dalam problem bersekala besar.

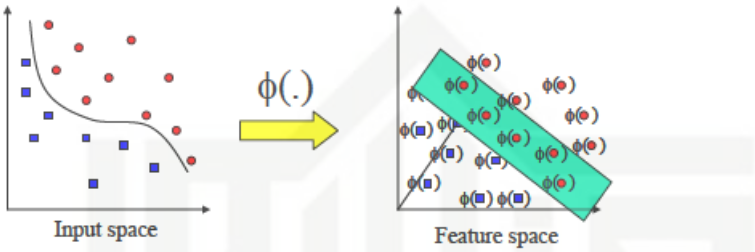
2.8.1 SVM pada *NonLinearly Separable* Data

NonLinearly Separable data adalah data yang tidak dapat dipisahkan secara linier. Untuk itu formula SVM harus dimodifikasi karena tidak ada solusi lain yang bisa ditemukan. Bidang pembatas harus diubah sehingga menjadikannya lebih fleksibel dalam kondisi tertentu dengan penambahan variabel ξi (i ≥ ∀i ξ 0, : ξ i = 0 jika jika x_i diklasifikasikan dengan benar) menjadi x_i.w + b ≥1-ξi untuk kelas 1 dan x_i.w + b ≤ −1+ξi untuk kelas 2. Bidang pemisah terbaik dengan penambahan variabel ξi disebut dengan *soft margin hyperplane*. Dengan demikian formulasi pencarian bidang pemisah terbai berubah menjadi

$$\min \frac{1}{2} |w|^2 + C(\sum_{i=1}^{n} \xi i) \quad (2.7)$$

$$s.t.\, y_1(w.x_1 + b) \geq 1\text{-}\xi i \quad (2.8)$$

C merupakan parameter yang menentukan besar penalti akibat kesalahan dalam klasifikasi data dan nilainya ditentukan oleh pengguna. Metode lain agar dapat mengklasifikasikan data yang tidak dapat dipisahkan secara linier adalah dengan menstransformasikan data kedalam dimensi ruang fitur sehingga mampu dipisahkan secara linier pada ruang fitur tadi.



Gambar 2.3 Transformasi dari vector input ke ruang input (Krisantus Sembiring, 2007)

Caranya dengan mempetakan data dengan menggunakan fungsi transformasi ke dalam ruang fitur sehingga menghasilkan bidang pemisah yang mampu memisahkan data sesuai kelasnya masing-masing. Untuk mengatasi masalah yang berdimensi tinggi dari vektor input, yang mengakibatkan komputasi pada ruang fitur memiliki jumlah fitur yang tak terhingga, maka digunakanlah sebuah *kerneltrick*. Syarat kernel yaitu memenuhi teorema Mercer yang menyatakan bahwa matriks kernel yang dihasilkan harus bersifat semi-definite. Berikut adalah fungsi kernel yang biasa digunakan :

a. Kernel Linier

$K(x_i, x) = x^t, x$ (2.9)

b. Polynomial Kernel

$K(x_i, x) = (\gamma . x_i^T x + r)^p , \gamma > 0)$ (2.10)

c. Radial Basis Function (RBF)

$K(x_i, x) = \exp(-\gamma |x_i - x|^2) \gamma > 0$ (2.11)

d. Sigmoid Kernel

$K(x_i, x) = \tanh(\gamma x_i^T x + r)$ (2.12)

Menurut(hsu 2002), fungsi kernel yang direkomendasikan untuk diuji pertama kali adalah fungsi kernel RBF. Karena kernel RBF mempunyai performansi sama dengan kernel linier pada parameter tertentu, memiliki perilaku

seperti terdapat pada fungsi kernel sigmoid dengan parameter tertentu dan rentang nilainya kecil. Selain itu karena kernel RBF menghasilkan tingkat kesalahan klasifikasi yang kecil serta mempercepat perhitungan komputasinya (XU et al., 2014). Pada penelitian ini akan diterapkan kernel *Radial Basis Function* (RBF) dengan menggunakan parameter C dan γ.

2.8.2 *Cross Validation and Grid Search*

Cross validation adalah metode statistic untuk mengevaluasi dan membandingkan algoritma pembelajaran dengan membagi data menjadi dua bagian yaitu data latih dan *dataset*. Semua data yang dikelompokkan kedalam dua bagian tersebut akan secara bergantian digilir kedalam bagian lainnya secara berurutan (Refailzadeh, et, al., 2008).

Grid seacrh merupakan metode untuk mencari nilai parameter terbaik dengan memberi range nilai pada parameter tersebut (Naufal, 2015). Sedangkan fungsi kernel yang akan digunakan adalah Radial Basis Function. Dalam kernel RBF ada dua parameter yaitu C dan γ yang nilainya perlu untuk diketahui agar mampu menghasilkan akurasi yang tinggi dalam pelatihan.

Algortima *grid search* ini biasanya menggunakan fungsi *k-fold cross validation*. Pencarian parameter terbaik akan dilakukan dengan cara membagi data menggunakan *k-fold cross validation* yaitu pada penelitian ini menggunkan 10-*fold cross validation*. Dalam *k-fold cross validation dataset* yang utuh akan dipecah secara random menjadi ‘k’ subset dengan ukuran yang sama dan saling eksklusif satu dengan yang lainnya. Setiap kali pelatihan semua akan dilatih pada semua *fold* kecuali hanya satu *fold* yang disisakan untuk pengujian.

2.8.3 *Confusion Matrix*

Confusion Matrix merupakan metode untuk menghitung tingkat akurasi, dengan menghitung jumlah prediksi benar dan salah dari sebuah metode klasifikasi berbanding dengan data sesungguhnya atau prediksi target menggunakan *Matrix* (NxN) dimana N adalah jumlah kelas (Junaidi) :

$$\text{Akurasi} = \frac{Jumlah\ prediksi\ benar}{Jumlah\ seluruh\ prediksi} \times 100\% \qquad (2.13)$$

2.9 *Multiclass SVM*

Pada awalnya SVM dikembangkan untuk persoalan klasifikasi dua kelas, kemudian dikembangkan kembali untuk klasifikasi multikelas (Santosa, 2007). ada dua pilihan untuk mengimplementasikan multi class SVM yaitu dengan menggabungkan beberapa SVM biner atau menggabungkan semua data yang terdiri dari beberapa kelas ke dalam sebuah bentuk permasalah optimasi. Namun, pada pendekatan yang kedua permasalahan optimasi yang harus diselesaikan jauh lebih rumit.

Tabel 2.2 Penelitian Terkait

No	Penulis	Judul	Tahun	Metode Klasifikasi	Jumlah Data	Akurasi
1	Elly Susilowati	Implementasi Metode Support Vector Machine Untuk Melakukan Klasifikasi Kemacetan Lalu Lintas Pada Twitter	2015	Support Vector Machine (SVM)	934	90%
2	Raudlatul Munawarah, Oni Soesanto, dan M. Reza Faisal	Penerapan Metode Support Vector Machine Pada Diagnosa Hepatitis	2016	Support Vector Machine (SVM)	579	83%

3	Uswatun Hasanah, Lintang Resita, M.Andhicha Pratama, dan Imam Cholissodin	Perbandingan Metode SVM, Fuzzy-KNN, Dan BDT-SVM Untuk Klasifikasi Detak Jantung Hasil Elektrokardiografi	2016	SVM, Fuzzy-KNN, BDT-SVM	140	SVM 81.30%, Fuzzy-KNN 81.25%, BDT-SVM 70.00%
4	Agus Winoto	Prediksi Umur Pahat Dengan Metode SVM	2011	Support Vector Machine (SVM)	50	90,03%
5	Lestari Handayani, Dan Fitriandini	Prediksi Kebangkrutan Perusahaan Menggunakan Support Vector Machine (SVM)	2010	Support Vector Machine (SVM)	50	90.78%
6	Siti Nur Asiyah Dan Kartika Fithriasari	Klasifikasi Berita Online Menggunakan Metode Support Vector Machine dan	2016	Support Vector Machine (SVM) dan K-Nearest Neighbor (KNN)	500	SVM 93,2% dan KNN 60%

		K-Nearest Neighbor				
7	Moh. Yamin Darsyah	Klasifikasi Tuberkulosis Dengan Pendekatan Metode Support Vector Machine (SVM)	2014	Support Vector Machine (SVM)	700	98%
8	Pusphita Anna Octaviani, Yuciana Wilandari, Dan Dwi Isprianti	Penerapan Metode Klasifikasi Support Vector Machine (SVM) Pada Data Akreditasi Sekolah Dasar (SD) Di Kabupaten Magelang	2014	Support Vector Machine (SVM)	419	93,90%
9	Liza Wikarsa,Sherly Novianti	A Twxt Mining Application of Emotion	2015	Naive Bayes	105	83%
	Thahir	Classifications of Twitter's				

		Users Using Naive Bayes Method				
10	Uma Nagarsekar, Priyanka Kulkarni, Aditi Mhapsekar.	Emotion Detection from The SMS of the Internet	2013	SVM dan Naive Bayes	1500	SVM 82.73% Naive Bayes 79.83%
11	Muljono, Aninnisa Sri Winarsih dan Catur SUpriyanto	Evaluation of Classification Methods for Indonesian Text Emotion Detection	2016	SVM-SMO, Naive Bayes, J48, KNN	789	SVM-SMO 85.5%, KNN 68.1%, J48 80,8%, Naive Bayes 80,2%