BAB 2
LANDASAN TEORI

2.1 Klasifikasi Berita

Klasifikasi dokumen berita digital menurut kategori tertentu diperlukan untuk mempermudah pencarian oleh pembaca. Peningkatan jumlah dokumen berita yang cukup besar tidak sebanding dengan ketersediaan editor ahli sehinggal diperlukan klasifikasi secara otomatis. Ketersedian dokumen digital di Internet yang berlimpah akan menyulitkan masyarakat untuk mengaksesnya jika dokumen tersebut tidak diatur secara layak. Pengaturan berita yang umum adalah dengan melakukan klasifikasi pada masing-masing artikel berita tersebut. Klasifikasi tersebut dapat didasarkan pada kondisi yang ada dalam masyarakat ataupun menurut standar khusus. Sebagai contoh, klasifikasi yang umum adalah politik, pendidikan, hiburan, ekonomi, olah raga, ilmu pengetahuan dan sebagainya. Jumlah klasifikasi tersebut sifatnya selalu berkembang. Proses klasifikasi dilakukan dengan melibatkan tenaga khusus yang memahami proses klasifikasi suatu artikel berita. [4]

Berita telah menjadi kebutuhan pokok manusia seiring dengan berkembangnya teknologi dan internet. Perkembangan teknologi dan internet ini menyebabkan proses pendistribusian informasi pada berita beralih dari cara penyampaian era media cetak menuju era digital. Berita yang disajikan dalam bentuk teks pada media digital, biasanya dikelompokkan berdasarkan isinya seperti berita olahraga, ekonomi, sains, dan lain sebagainya. Permasalahan yang muncul adalah penggunaan media digital dalam penyampaian informasi menyebabkan

jumlah berita digital yang dirilis oleh portal berita tiap harinya menjadi sangat banyak [1].

2.2 Text Classification

Dalam beberapa dekade terakhir, masalah klasifikasi teks telah dipelajari dan ditangani secara ekstensif dalam banyak aplikasi praktis. Terutama terobosan terbaru dalam Natural Language Processing (NLP) dan text mining. Saat ini banyak peneliti yang tertarik untuk mengembangkan aplikasi yang memanfaatkan metode klasifikasi teks [5]. Klasifikasi teks adalah sebuah pekerjaan untuk menentukan apakah sebuah dokumen adalah milik dari sebuah kategori yang telah ditentukan sebelumnya [6].

Saat ini, text mining telah mendapat perhatian dalam berbagai bidang, antara lain dibidang keamanan, biomedis, pengembangan perangkat lunak dan aplikasi, media online, pemasaran, dan akademik. Seperti halnya dalam data mining, aplikasi text mining pada suatu studi kasus, harus dilakukan sesuai prosedur analisis. Langkah awal sebelum suatu data teks dianalisis menggunakan metode-metode dalam text mining adalah melakukan pre-processing teks. Selanjutnya, setelah didapatkan data yang siap diolah, analisis text mining dapat dilakukan [7].

Penggunaan kategori dalam website yang menampikan berita juga dapat meningkatkan utilitas kecepatan proses pencaharian data karena data telah dikelompokan berdasarkan kategori tertentu secara teratur dan signifikan. Tahapan dalam klasifikasi teks antara lain adalah [8] :

1. Preprocessing

   Merupakan tahapan untuk merpresentasikan dokumen dalam bentuk fitur vektor, yang berarti harus memisahkan teks menjadi kata terpisah. Dalam tahap preprocessing, dilakukanlah penghapusan stopwords pada dokumen, dengan tujuan untuk menghapus kata-kata umum dan tak bermakna yang disesuaikan dengan kosakata bahasa yang digunakan. Setelah stopwords dihapus, dilakukan tahapan stemming yang digunakan untuk mencari kata dasar dari kata yang telah diekstraksi dari dokumen.

2. Rekayasa Fitur

   Pada tahap ini merupakan tahapan latih yang yang terdiri dari tahapan seleksi fitur, dictionary construction, dan feature weighting. Tujuan dari rekayasa fitur adalah untuk menghapus semua fitur yang tidak relevan dan selalu muncul pada semua dokumen.

3. Generasi Model Klasifikasi

   Tahap ini merupakan tahap untuk membangun algoritme klasifikasi, dalam penelitian ini menggunakan metode Support Vector Machine (SVM) atau juga melakukan generasi model classifier berdasarkan hasil pelatihan oleh dokumen sebelumnya yang akan digunakan untuk mengklasifikasikan dokumen yang tidak diketahui kategorinya.

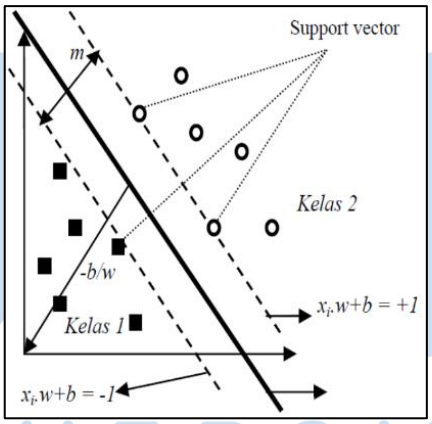
4. Pengkategorian Dokumen

   Merupakan tahapan untuk melakukan klasifikasi dari dokumen baru yang tidak diketahui asal kategori dari dokumen tersebut, dengan

catatan bahwa dokumen baru tersebut telah melewati tahap preprocessing dan feature weighting.

2.3 Support Vector Machine (SVM)

Support Vector Machine (SVM) adalah metode klasifikasi yang pertama kali dikenalkan oleh Vapnik pada tahun 1998. Pada dasarnya, metode ini bekerja dengan cara mendefinisikan batas antara dua kelas dengan jarak maksimal dari data yang terdekat. Untuk mendapatkan batas maksimal antar kelas maka harus dibentuk sebuah *hyperplane* (garis pemisah) terbaik pada input space yang diperoleh dengan mengukur margin hyperplane dan mencari titik maksimalnya. Margin merupakan jarak antara hyperplane dengan titik terdekat dari masing-masing kelas. Titik terdekat inilah yang disebut sebagai *support vector*. SVM dapat melakukan klasifikasi data yang terpisah secara linier (linearly separable) dan non-linier (nonlinear separable) [12]. *Hyperline* SVM dapat dilihat pada Gambar 2 di bawah ini [9].



Gambar 2.1 Konsep Hyperlane pada SVM
Sumber : [12]

Data yang berada pada bidang pembatas disebut dengan support vector. Dalam Gambar 6.2, dua kelas dapat dipisahkan oleh sepasang bidang pembatas yang

sejajar. Bidang pembatas pertama membatasi kelas pertama sedangkan bidang pembatas kedua membatasi kelas kedua, sehingga diperoleh:

$$\begin{aligned} \boldsymbol{x_i}\,\boldsymbol{w} + \ b \geq +1, y_i = +1 \\ \boldsymbol{x_i}\,\boldsymbol{w} + \ b \leq -1, y_i = -1 \\ \text{i} = 1,2,\ldots, \text{p} \end{aligned} \qquad \ldots(2.1)$$

x pada rumus diatas melambangkan judul dari berita, **y** melambangkan label dari kategori berita, sedangkan **i** melambangkan data label ke-i. **w** adalah normal bidang dan **b** adalah posisi bidang alternatif terhadap pusat koordinat. Nilai margin (jarak) antara bidang pembatas (berdasarkan rumus jarak garis ke titik pusat) adalah .

$$\frac{1-b-(-1-b)}{||w||} = \frac{2}{||w||}. \qquad \ldots(2.2)$$

Nilai margin ini dimaksimalkan dengan tetap memenuhi persamaan. Dengan mengalikan ***b*** dan **w** dengan sebuah konstanta, akan dihasilkan nilai margin yang dikalikan dengan konstata yang sama. Kedua bidang pembatas pada persamaan diatas direpresentasikan dalam bentuk persamaan, $y_i(\boldsymbol{x_i}\boldsymbol{w} + b) - 1 \geq 0$ maka pencarian bidang pemisah dengan nilai margin terbesar dapat dirumuskan menjadi masalah optimasi konstrain, yaitu:

$$min\frac{1}{2}\left|\left|\boldsymbol{w}\right|\right|^2 \qquad \ldots(2.3)$$
$$\text{dengan } y_i(\boldsymbol{x_i}\boldsymbol{w} + b) - 1 \geq 0$$

Jadi persoalan pencarian bidang pemisah terbaik dapat dirumuskan pada persamaan sebagai berikut:

$$\text{dengan } \max_a L_D = \sum_{i=1}^{n} a_i - \frac{1}{2}\sum_{i,j=1}^{n} a_i\, a_j y_i y_j x_i x_j \qquad \ldots(2.4)$$
$$\sum_{i=0}^{n} a_i\, y_i = 0, a_i \geq 0$$

Dari persamaan diatas dapat diperoleh nilai α_i yang nantinya digunakan untuk menemukan **w**. Terdapat nilai α_i untuk setiap data training yang memiliki nilai $\alpha i>0$ adalah *support vector* sedangkan sisanya memiliki nilai $\alpha i=0$. Dengan demikian fungsi keputusan yang dihasilkan hanya dipengaruhi oleh *support vector*.

Untuk menemukan hyperplane optimal, biasanya mengambil sebagian besar rekaman berlabel sebagai training set. Namun, hyperplane pemisah hanya ditentukan oleh beberapa sampel penting (Support Vectors, SVs), sehingga tidak perlu melatih model SVM di seluruh training set [14]. Salah satu kelebihan dari metode SVM adalah mampu menangani kasus dengan input space yang berdimensi tinggi. Namun, SVM juga memiliki kelemahan, salah satunya adalah komputasi yang lama untuk proses klasifikasi [15].

2.4 Preprocessing

Preprocessing adalah suatu tahapan mengubah teks asli sebagai masukan dan menerapkan beberapa rutinitas dasar untuk mengubah atau menghilangkan unsur tekstual yang tidak berguna dalam pengolahan lebih lanjut [16]. Berikut merupakan beberapa tahapan dalam preprocessing.

a) Case Folding

Merupakan proses untuk mengubah semua teks dokumen menjadi huruf kecil.

b) Tokenization

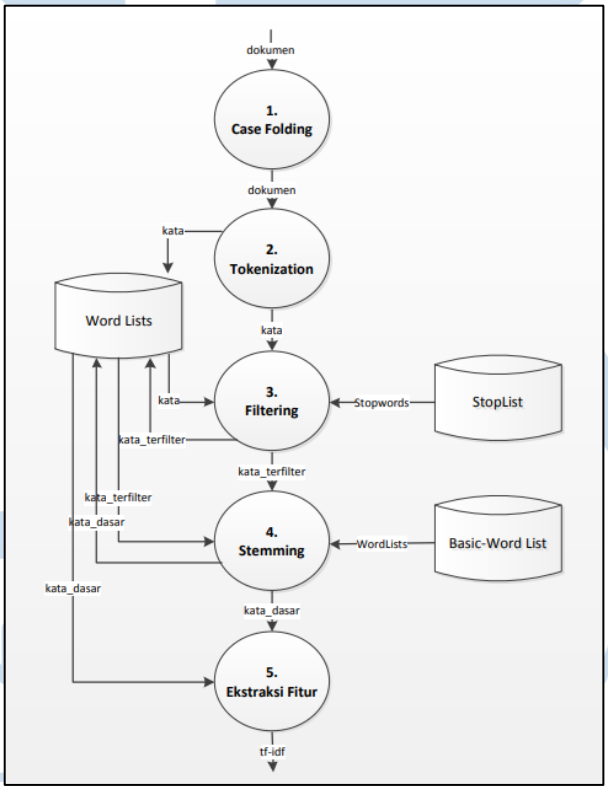
Proses untuk memecah teks dokumen menjadi kalimat, kemudian memecahnya menjadi kata-kata. Proses ini juga dilakukan untuk menghilangkan angka, tanda baca dan spasi.

c) Filtering

Filtering merupakan proses untuk membuang katakata yang tidak bermakna dalam dokumen. Daftar kata-kata yang tidak bermakna disimpan dalam sebuah basis pengetahuan bernama stoplist. Pada penelitian sebelumnya, *stoplist* yang digunakan terdiri dari 906 kata, yang merupakan gabungan dari *stoplist* dan *most common words*.

d) Stemming

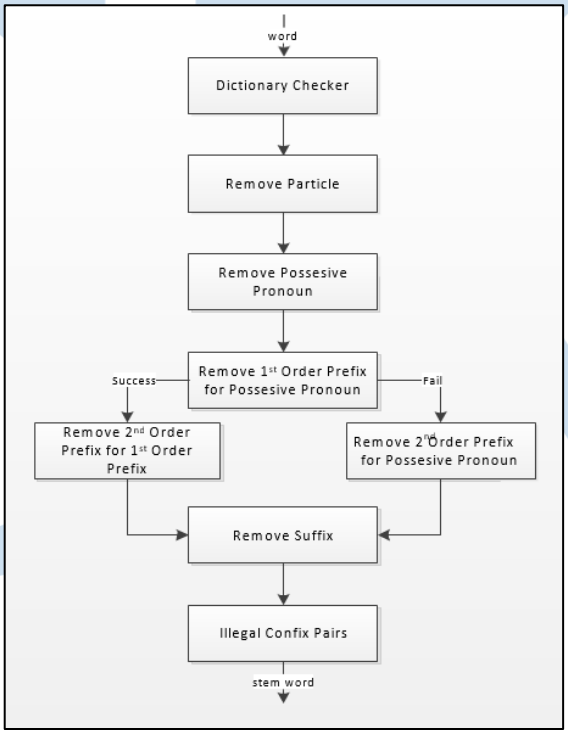
Stemming merupakan proses untuk mencari kata dasar dari setiap kata dalam dokumen dengan membuang imbuhan, baik awalan maupun akhiran. Proses ini menggunakan basis pengetahuan bernama *basic-words list* sebagai kamus kata dasar.



Gambar 2.2 Bagan Proses Preprocessing
Sumber : [10]

2.5 Metode Ekstraksi Fitur TF-IDF

Ekstraksi fitur adalah proses menemukan nilai fitur yang terdapat dalam dokumen untuk proses *text mining*. Ekstraksi fitur adalah bagian yang sangat penting dari pemrosesan dokumen Mesin pencari, karena sangat menentukan keberhasilan proses *text mining*. TF-IDF adalah metode ekstraksi fitur yang banyak digunakan dan populer. Dalam penelitian ini telah diterapkan metode TF-IDF. Pemrosesan dokumen melibatkan *case folding, tokenization, filtering, stemming* dari proses preprocessing dan ekstraksi fitur. Hasil ekstraksi fitur berupa matriks yang berisi urutan kata unik dari semua dokumen dan nilai fitur TF-IDF dari setiap kata pada dokumen. Berikut beberapa tahapan dalam metode ekstraksi fitur TF-IDF [10] :



Gambar 2.3 Bagan Proses TF-IDF
Sumber : [10]

Pada skema Term Frequency-Inverse Document Frequency (TF-IDF), TF dihitung berdasarkan jumlah kemunculan setiap kata dalam tiap dokumen, dan IDF

dihitung berdasarkan jumlah kemunculan kata dalam keseluruhan dokumen. Setelah melalui proses normalisasi, nilai TF dibandingkan terhadap nilai IDF (pada umumnya berbentuk skala logaritma). Hasil akhirnya berupa matriks term-document X, dimana kolom-kolomnya berisi nilai TF-IDF untuk setiap dokumen. Oleh karena itu, skema TF-IDF mengurangi ukuran panjang dokumen yang bervariasi menjadi dokumen dengan ukuran yang tetap [10].

2.6 Metode Confusion Matrix

Hasil evaluasi judul berita menggunakan algoritma *Support Vector Machine* (SVM) akan dilakukan dengan menggunakan metode *confusion matrix*. Metode ini menggunakan matriks untuk mewakili hasil klasifikasi, seperti yang ditunjukkan pada gambar berikut.

Tabel 2.1 Confusion Matrix
Sumber : [11]

Correct Classification		**Classified as**	
		+	-
	+	True positive	False positive
	-	False negative	True negative

True Positive adalah jumlah *record* positif yang berhasil diklasifikasikan sebagai positif, dan *false positive* adalah *record* positif yang salah diklasifikasikan sebagai negatif. *False negative* adalah *record* negatif yang salah diklasifikasikan sebagai positif, dan *true negative* adalah *record* negatif yang telah berhasil diklasifikasikan sebagai *record* negative [18]. Berikut merupakan beberapa metode pengujian *confusion matrix* yang dapat menghasilkan perhitungan dengan 4 *output*.

$$\text{Precisions} = \frac{TP}{TP+FP} \times 100\% \quad \ldots(2.5)$$

Pada rumus 2.5 diatas, nilai *preciosions* mendefinisikan tingkat akurasi antara informasi yang diminta oleh pengguna dan jawaban yang diberikan oleh sistem.

$$\text{Recall} = \frac{TP}{TP+FN} \times 100\% \quad \ldots(2.6)$$

Pada rumus 2.6 diatas, nilai *recall* mendefinisikan tingkat keberhasilan sistem untuk mengambil informasi.

$$\text{Accuracy} = \frac{TP+FN}{TP+TN+FP+FN} \times 100\% \quad \ldots(2.7)$$

Pada rumus 2.7 diatas, nilai *accuracy* mendefenisikan sebagai tingkatan kedekatan antara nilai prediksi dan nilai sebenarnya.

$$\text{F1-Score} = \frac{2 \times Precisions \times Recall}{Precisions + Recall} \times 100\% \quad \ldots(2.8)$$

Pada rumus 2.8 diatas, nilai F1-Score mendefinisikan perbandingan antara rata-rata nilai *precisions* dan nilai *recall* yang dibobotkan untuk menghitung akurasi dari *testing* yang dilakukan.