BAB 2
LANDASAN TEORI

2.1 *Text Preprocessing*

Text preprocessing merupakan serangkaian tahap dalam persiapan data awal berupa teks yang akan diterapkan metode yang melibatkan algoritma pembelajaran mesin dimana badan teks (*text corpora*) yang awalnya masih dalam bentuk data tidak terstruktur yang terdiri atas dokumen-dokumen berisi kata-kata (atau dalam kajian seperti ini lebih dikenal sebagai *terms*, yang berarti istilah-istilah dalam bahasa Inggris), diubah menjadi bentuk data terstruktur berupa *Document-Term Matrix*. *Text preprocessing* umumnya terdiri atas beberapa proses, dalam kajian ini proses-proses text preprocessing yang dilibatkan diantaranya adalah penghilangan tanda baca, *casefolding, filtering, stemming* dan tokenisasi. [12]

Tahap penghilangan tanda baca merupakan tahap dimana semua tanda baca dilucuti dari badan teks agar tanda baca tersebut tidak muncul menjadi fitur dengan sendirinya atau bagian dari fitur berupa kata/istilah yang disertai tanda baca pada tahap seleksi fitur maupun tahap klasifikasi berikutnya. [16]

Tahap *casefolding* merupakan tahap dimana *case* dari seluruh huruf yang ada di dalam badan teks diseragamkan (bisa dalam bentuk semua huruf kapital atau semua huruf kecil). Pendekatan yang lebih konvensional adalah untuk menyeragamkan seluruh huruf yang ada di dalam badan teks sebagai huruf kecil sebagaimana juga diberlakukan dalam kajian ini. [16]

Tahap *Filtering* atau juga dikenal sebagai tahap Penghilangan *Stop Words* adalah tahap penghilangan kata-kata/istilah-istilah yang masuk dalam kategori *Stop Words*. *Stop Words* sendiri didefinisikan sebagai kumpulan kata-kata lazim, yang karena sifat kelaziman itu sendiri dianggap akan berkontribusi minim untuk dijadikan diskriminan efektif sebagai fitur dalam proses klasifikasi. Contoh jenis-jenis kata/istilah yang masuk dalam kategori Stop Words adalah kata

penghubung, kata ganti orang dan preposisi yang pastinya lazim dalam suatu teks bahasa alami sehingga tidak dapat dijadikan sebagai fitur dalam proses klasifikasi teks yang indikatif terhadap suatu himpunan pengelompokkan tertentu. [16]

Tahap *Stemming* adalah tahap dimana kata-kata/istilah-istilah dalam badan teks yang tidak dalam bentuk akarnya (kata-kata berimbuhan atau dalam bentuk tidak tak beraturan bukan akar) dirubah kedalam bentuk dasarnya (*stem*). Proses ini dicapai melalui penghilangan imbuhan dan merubah bentuk kata-kata yang tak beraturan yang tidak dalam bentuk akarnya menjadi kata-kata dalam bentuk akar sebagaimana sesuai dengan kaidah pengakaran kata yang terdapat di dalam lexicon *Stemmer*. [16]

Tahap tokenisasi adalah tahap memisahkan setiap kata yang terindentifikasi atau terpisahkan dengan kata yang lainnya oleh pemisah spasi yang akan dipecah dari abstrak menjadi kata-kata tunggal [16], dimana masing-masing token (kata-kata tunggal sebagai hasil pecahan tahap tokenisasi) akan dijadikan fitur dalam bentuk data terstruktur setelah melalui tahap vektorisasi kata.

2.2 Vektorisasi Kata dalam Bentuk *Document-Term Matrix*

Data yang awalnya tidak terstruktur yang berwujud sebagai badan teks (*text corpora*) yang terdiri atas kumpulan dokumen-dokumen (*document*) berisi kata-kata/istilah-istilah (*term*) yang telah melalui tahap *preprocessing* dapat disusun menjadi suatu data terstruktur dalam bentuk *Document-Term Matrix*, yaitu suatu representasi matriks dimana indeks kolomnya mewakili setiap kata yang ada dalam badan teks yang telah terseleksi melalui tahap *preprocessing* sebelumnya dan indeks barisnya mewakili satuan dokumen-dokumen yang menyusun badan teks tersebut. Sementara isi elemen-elemen *Document-Term Matrix* yang memiliki indeks baris yang mewakili dokumen dan indeks kolom kata/istilah tersebut berisi nilai pembobotan dari masing-masing fitur kata/istilah tersebut terhadap dokumen-dokumen dalam badan teks.

	afrika	agam	agen	agenda	agendain	agenpulsa
0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...
1138	0.0	0.0	0.0	0.0	0.0	0.0
1139	0.0	0.0	0.0	0.0	0.0	0.0
1140	0.0	0.0	0.0	0.0	0.0	0.0
1141	0.0	0.0	0.0	0.0	0.0	0.0
1142	0.0	0.0	0.0	0.0	0.0	0.0

Gambar 2.2 – *Document-Term Matrix* tanpa nilai pembobotan

2.3 Pembobotan *Term Frequency – Inverse Document Frequency*

Term weighting atau pembobotan kata bertujuan untuk memberikan bobot/nilai pada elemen-elemen dalam *Document-Term Matrix* yang berkaitan dengan nilai pembobotan setiap kata untuk masing-masing dokumen yang ada dalam badan teks yang sudah terseleksi melalui tahap *pre-processing*. Ada berbagai macam jenis pembobotan kata yang bisa diterapkan, salah satunya adalah pembobotan *Term Frequency – Inverse Document Frequency* (TF-IDF). Perhitungan bobot ini membutuhkan dua parameter, yaitu *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF). *Term Frequency* merupakan banyaknya jumlah kata/istilah (*term*) tertentu yang ada dalam suatu dokumen. Sementara *Inverse Document Frequency* adalah frekuensi kemunculan kata/istilah pada seluruh dokumen. Nilai IDF berbanding terbalik dengan jumlah dokumen

yang mengandung kata/istilah tertentu. Kata/Istilah yang jarang muncul pada seluruh dokumen memiliki nilai IDF yang lebih besar dari nilai IDF kata/istilah yang sering muncul. Jika pada setiap dokumen mengandung kata/istilah tertentu, maka nilai IDF kata/istilah tersebut bernilai 0. Hal ini menunjukan bahwa kata/istilah yang muncul pada seluruh dokumen merupakan kata/istilah yang tidak berguna untuk membedakan dokumen berdasarkan topik tertentu.

$$\mathbf{W_{dt} = tf_{dt} \times idf_t = tf_{dt} \times \log(\frac{N}{dft})} \qquad \mathbf{(2.1)}$$

W_{dt} : Bobot istilah ‘t’ terhadap dokumen ‘d’

tf_{dt} : jumlah kemunculan istilah ‘t’ dalam dokumen ‘d’

N : Jumlah dokumen secara keseluruhan

d_{ft} : jumlah dokumen yang mengandung istilah ‘t’

2.4 Seleksi Fitur

Adanya fitur yang sangat banyak dikenal dengan istilah *high dimensional data*. Data dengan dimensi yang besar membawa beberapa masalah pada penerapan metode yang melibatkan pembelajaran mesin. Masalah tersebut diantaranya model pembelajaran sulit untuk memiliki kinerja yang optimal pada data berdimensi tinggi. Semakin banyak fitur yang digunakan maka semakin kompleks suatu model pembelajaran mesin harus memodelkan permasalahan. Selain itu, permasalahan *high dimensional data* juga menyebabkan mudah terjadi *overfitting* dan juga sulit untuk diproses secara komputasi, baik dari segi memori (*high space complexity*) maupun waktu (*high time complexity*). Masalah-masalah *high dimensional data* dapat diselesaikan menggunakan metode seleksi fitur. Metode seleksi fitur dapat membantu memilih fitur yang paling informatif dan relevan. Berdasarkan tekniknya, seleksi fitur pada dasarnya dapat dibagi menjadi metode *filter* dan metode *wrapper*.

Metode *filter* adalah metode seleksi fitur yang tidak bergantung pada algoritma pembelajaran mesin yang diterapkan dengan seleksi fitur, tetapi membutuhkan hitungan statistika untuk me-*ranking* (memberi peringkat) fitur. Contoh metode *filter* antara lain Chi-Squared Test. Metode ini mengevaluasi secara bebas dari metode pembelajaran mesin klasifikasi yang diterapkan setelah metode seleksi fitur lalu memberikan peringkat pada fitur-fitur yang ada dan menyaring (me-*filter*) fitur dengan peringkat yang paling unggul atau fitur yang melebihi suatu nilai ambang batas tertentu. Metode *filter* menggunakan kriteria penilaian yang tepat yang mencangkup jarak, informasi, ketergantungan dan konsistensi. Metode ini dapat memproses dataset menghasilkan fitur yang relevan dengan sederhana dan cepat secara komputasi, tetapi metode ini hanya mempertimbangkan fitur secara sendiri-sendiri tanpa mempertimbangkan interaksi antar fitur, sehingga dapat menurunkan kemampuan dalam mengklasifikasi suatu masalah. [17]

Metode *wrapper* membutuhkan satu algoritma pembelajaran mesin dan mengevaluasi kinerjanya. Contoh *metode wrapper* diantaranya *forward feature*

selection, backward feature selection dan *genetic algorithm*. Metode ini melakukan seleksi fitur bersamaan dengan membuat model. Metode ini bekerja lebih baik daripada metode filter karena mengevaluasi semua kemungkinan kombinasi fitur dan memilih kombinasi yang menghasilkan hasil terbaik untuk algoritma pembelajaran mesin yang akan diterapkan setelah proses seleksi fitur. Fitur-fitur dipilih berdasarkan kontribusinya terhadap akurasi klasifikasi. Namun metode ini membutuhkan waktu komputasi yang lama dan juga mahal dari segi kompleksitas ruang. [17]

2.5 *Chi-Squared Statistic* (Statistika X^2)

Chi-Squared secara umum digunakan untuk mengukur derajat atau sejauh apa ketidakadaan akan ketergantungan antara istilah dan kategori dan dibandingkan terhadap distribusi X^2 dengan derajat kebebasan sebesar satu [12]. Rumusan X^2 didefinisikan sebagai:

$$\mathbf{X^2}(t, c) = \frac{\boldsymbol{D} \text{ x } (\boldsymbol{PN}-\boldsymbol{MQ})^{\boldsymbol{2}}}{(\boldsymbol{P}+\boldsymbol{M})(\boldsymbol{Q}+\boldsymbol{N})(\boldsymbol{P}+\boldsymbol{Q})(\boldsymbol{M}+\boldsymbol{N})} \quad \textbf{(2.2)}$$

dimana D merupakan jumlah dokumen keseluruhan. P adalah dokumen- dokumen dari suatu kelas 'c' yang mengandung istilah 't'. Q adalah jumlah dokumen mengandung t yang tidak termasuk dalam c. M adalah jumlah dokumen yang termasuk kategori c yang tidak mengandung t dan N adalah jumlah dokumen dari kelas lain yang tidak mengandung t. Dalam kajian ini, istilah-istilah yang terseleksi akan direpresentasikan berdasarkan kemunculannya di dalam sebuah matriks istilah dokumen. [15]

2.6 *K-Means Clustering*

K-Means merupakan salah satu dari algoritma clustering yang paling sering dan juga yang paling mudah diterapkan. K-Means termasuk dalam kategori unsupervised learning dan dapat digunakan untuk menyelesaikan berbagai masalah. Ide yang mendasari K-Means adalah melakukan clustering objek berdasarkan kemiripan antar elemen dalam sebuah cluster. prinip dasarnya adalah untuk menentukan centroid dan menentukan anggota cluster.

Centroid merupakan pusat cluster. Jumalh centroid identik dengan jumlah cluster. Penentuan centroid awal ditentukan secara random kemudian titik centroid baru dapat dicari melalui proses perhitungan secara berulang-ulang hingga ditentukan titik centroid akhir, yaitu titik centroid yang tidak berpindah-pindah lagi mesikpun telah melalui berberapa iterasi selanjutnya.

Menentukan anggota cluster berarti mengelompokkan sampel ke masing-masing cluster yang dihitung dengan membandingkan jarak antara sampel dengan centroid masing-masing cluster. Sampel yang paling dekat dengan centroid akan dikelompokkan menjadi satu cluster dengan centroid tersebut.

Perhitungan jarak centroid dengan masing-masing sampel dapat dicari menggunakan rumus Euclidean distance, Manhattan distance dan berbagai rumus lainnya. [18]

2.7 Algoritma FS-CHICLUST

Seleksi Fitur FS-CHICLUST yang pertama kali diusul oleh Sarkar pada tahun 2014 yang mengaplikasikan seleksi fitur ini dalam penerapan kategorisasi teks menggunakan metode Naive-Bayes yang merupakan pendekatan seleksi fitur dengan dua tahap yaitu tahap *filter* dan tahap *wrapper*. Tahap *filter* FS-CHICLUST menerapkan seleksi fitur univariat dengan metrik *Chi-Squared* untuk menyeleksi fitur-fitur penting yang diikuti dengan tahap *wrapper* berupa *feature clustering* yang penerapannya serupa dengan *K-Means clustering* untuk mereduksi ruang fitur lebih lanjut.

Metode seleksi fitur univariat pada tahap pertama berfungsi untuk mereduksi ruang pencarian. Pata tahap ini istilah-istilah penting diseleksi terlebih dahulu

melalui penyaringan metrik Chi-Squared, yaitu suatu proses dimana hanya istilah-istilah yang memiliki nilai lewat ambang batas yang diistilahkan sebagai thresh akan dipertimbangkan dan digunakan pada tahapan proses selanjutnya (Dalam kajian ini nilai thresh ditentukan sebagai nol).

Suatu matriks yang disebut sebagai matriks dokumen-istilah merepresentasikan masing-masing istilah yang telah diseleksi pada tahap sebelumnya sebagai kolom dan setiap dokumen sebagai baris, dimana isi atau nilai dari matriks dokumen-istilah berupa nilai pembobotan TF-IDF yang telah diperoleh sebelumnya untuk masing-masing istilah terhadap dokumen pada setiap sel matriks yang berbeda.

Setelah matriks dokumen-istilah dibentuk, tahap berikutnya adalah *feature clustering* untuk menyeleksi istilah-istilah penting lebih lanjut. Hal ini dicapai dengan mentranspos matriks dokumen-istilah yang dibentuk dan menerapkan metode K-Means clustering terhadapnya. Jumlah cluster yang ditentukan pada kajian ini untuk melangsungkan metode K-Means clustering adalah nc, yang nilainya diambil dari akar dari jumlah dokumen yang dipertimbangkan dalam kajian ini. Istilah-istilah yang paling representatif dari setiap cluster kemudian diseleksi, yaitu yang jaraknya paling dekat berdasarkan kaidah jarak Euclidean dengan pusat cluster dan menambahkannya satu per satu ke himpunan fitur (F) sehingga n(F) = nc (himpunan fitur F merupakan output dari algoritma FS-CHICLUST yang berisi fitur-fitur yang terseleksi yang awalnya berupa himpunan kosong sebelum algoritma FS-CHICLUST dilangsungkan). [15]

2.8 *Classifier* Naïve-Bayes Multinomial

Model multinomial memperhitungkan frekuensi setiap kata yang muncul pada dokumen. Misal terdapat dokumen ‘d’ dan himpunan kelas ‘c’ [16]. Untuk memperhitungkan kelas dari dokumen ‘d’, maka dapat dihitung dengan rumus:

$$\mathbf{P(c|}\textit{\textbf{term dokumen}}\ \mathbf{d) = P(c) \times P(t_1|c) \times P(t_2|c) \times P(t_3|c) \times \ldots \times P(t_n|c)} \quad \mathbf{(2.3)}$$

P(c) = Probabilitas prior dari kelas c
t_n = Kata dokumen ke-n
P(c|term dokumen d) = Probabilitas suatu dokumen termasuk kelas c
$P(t_n|c)$ = Probabilitas kata ke-n dengan diketahu kelas c

Probabilitas prior kelas c ditentukan dengan rumus [16]:

$$\mathbf{P(c) = \frac{N_c}{N}} \quad \mathbf{(2.4)}$$

Keterangan :
Nc = Jumlah kelas c pada seluruh dokumen
N = Jumlah seluruh dokumen

Rumus Multinomial yang digunakan dengan pembobotan kata TF-IDF adalah sebagai berikut [16]:

$$\mathbf{P(t_n|c) = \frac{Wct+1}{\sum W' \in VW' ct+B'}} \quad \mathbf{(2.5)}$$

Keterangan :
Wct = Nilai pembobotan tfidf atau W dari *term* t di kategori c
$\sum W' \in VW' ct$ = Jumlah total W dari keseluruhan *term* yang berada di kategori c.
B' = Jumlah W kata unik (nilai idf tidak dikali dengan tf) pada seluruh dokumen.

2.9 *Confusion Matrix*

Proses evaluasi pada penelitian ini menggunakan perhitungan akurasi, precision dan recall dari hasil klasifikasi yang disajikan dengan tabel confusion matrix [16]:

Realita	Sistem				
	Kelas-1	Kelas-2	....	Kelas-n	Total
Kelas-1	True Positive	Error	....	Error	Total Kelas-1
Kelas-2	Error	True Positive	...	Error	Total Kelas-2
...	Error	Error	...	Error	...
Kelas-n	Error	Error	...	True Positive	Total Kelas-n
	Prediksi Kelas-1	Prediksi Kelas-2	...	Prediksi Kelas-n	

Gambar 2.9 – Confusion Matrix

Proses Evaluasi pada makalah ini menggunakan perhitungan precision, recall,danf1-scoredari hasil klasifikasi yang disajikan dengan rumus perhitungan sebagai berikut [16]:

$$\textbf{precision} = \frac{TP(Kelas-i)}{Prediksi(Kelas-i)} \quad \textbf{(2.6)}$$

$$\textbf{recall} = \frac{TP(Kelas-i)}{Total(Kelas-i)} \quad \textbf{(2.7)}$$

Dari perolehan hasil perhitungan *precision* dan *recall*, nilai F1-Score bisa diperoleh sebagai [16]:

$$\textbf{F1-Score} = \frac{2\ x\ Precision\ x\ Recall}{Precision + Recall} \quad \textbf{(2.8)}$$

Nilai akurasi keseluruhan dapat diperoleh dengan rumus [16]:

$$\textbf{Accuracy} = \frac{TP(Kelas-1) + TP(Kelas-2) + \cdots + TP(Kelas-n)}{Total(Kelas-1) + Total(Kelas-2) + \cdots + Total(Kelas-n)}\ x\ 100\% \quad \textbf{(2.9)}$$

Keterangan:

TP = *True Positive*

TN= *True Negative*

FP = *False Positive*

FN= *False Negative*


UNIVERSITAS KOMPUTER INDONESIA
UNIKOM