



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

LANDASAN TEORI

2.1. Sentimen Analisis

Sentimen Analisis disebut juga dengan *opinion mining* (pengembangan opini) merupakan proses untuk meng-ekstrak suatu opini atau pendapat dari dokumen untuk topik tertentu (Kaur, Mangat, and Nidhi 2017). Tujuan sentimen analisis adalah untuk menentukan *attitude* dari pembicara ataupun penulis yang berhubungan dengan topik tertentu (Cahyono and Saprudin 2019). Dari topik tertentu, analisis sentimen dapat digunakan untuk menentukan nilai kesukaan atau ketidaksukaan seseorang terhadap suatu barang (Haryanto, Muflikhah, and Fauzi 2018). Nilai sentimen analisis tersebut dapat ditentukan ke dalam 3 kategori seperti positif, negatif, atau netral.

2.2. User Feedback

User Feedback sangat penting karena untuk pemeliharaan perangkat lunak dan evolusi aplikasi seluler yang terpandu oleh permintaan yang terkandung dalam *user feedback* (Grano et al. 2017). Dalam menyelidiki *user feedback* yang dilaporkan pengguna untuk aplikasi yang digunakan, pengguna dapat memberikan informasi yang berharga tentang fitur yang harus diperhatikan atau embantu dalam memahami masalah yang terkait pada aplikasi, *User feedback* yang disampaikan oleh pengguna akan disampaikan kepada *developer* agar aplikasi yang digunakan bisa terpelihara. *User Feedback* yang diberikan oleh pengguna biasanya terbagi atas dua macam, yaitu komentar positif atau komentar negative (Haryanto, Muflikhah, and Fauzi 2018). Disisi lain analisis yang lebih mendalam terhadap *user feedback* tersebut dapat

mengevaluasi *developer* dan memberikan saran terhadap aplikasi tersebut untuk meningkatkan aplikasi agar mendapatkan kepuasan pelanggan yang tinggi.

2.3. Data Preprocessing

Data Preprocessing atau pemrosesan awal dokumen merupakan tahapan awal yang berfungsi dalam mengtransformasikan dokumen ke dalam bentuk representasi lain. Tujuan dari tahap ini adalah untuk mempermudah proses pencarian *query* ke dokumen, dan mempermudah dalam proses mengurutkan dokumen – dokumen yang diambil (Cahyono and Sapudin 2019). Dalam penelitian ini tahapan yang dilakukan *case folding*, *tokenize*, *stopword removal*, *stemming* dengan menggunakan sastrawi.

a. Case Folding

Dalam tahapan ini akan dilakukan perubahan huruf dalam dataset berubah menjadi huruf kecil. Hasil yang diterima hanya huruf ‘a’ sampai dengan ‘z’. sedangkan karakter lain selain huruf akan dihilangkan atau dianggap menjadi *delimiter*. Contoh untuk *case folding* adalah kata “Aku adalah Pelaut” akan menjadi “aku adalah pelaut”.

b. Tokenizing

Tahapan ini dilakukan proses pemotongan *string* berdasarkan tiap kata yang menyusunnya, serta membedakan karakter tertentu yang dapat diperlakukan sebagai pemisah kata atau bukan. Contoh untuk *tokenizing* adalah kata “aplikasi yang digunakan sangat baik” akan diubah menjadi perpotongan pada setiap katanya seperti “aplikasi”, “yang”, “digunakan”, “sangat”, “baik”.

c. *Stopword removal*

Selanjutnya untuk tahap ini akan menghilangkan kata-kata pada daftar *stopwords* yang tidak memiliki arti. *Corpus* yang digunakan dalam *stopword removal* berasal dari PySastrawi seperti kata “yang”, “dan”, “di”, “dari”.

d. *Stemming*

Stemming merupakan proses menghilangkan infleksi kata ke bentuk dasarnya, namun bentuk dasar tersebut tidak berarti sama dengan akar kata (*root word*). Dalam *stemming* bahasa Indonesia diperlukannya menghilangkan sufiks dan prefiks dan itu membuat berbeda dari *stemming* pada teks berbahasa Inggris. Untuk melakukannya dapat menggunakan Python Sastrawi yang diterapkan oleh Algoritma Nazief dan Adriani dalam melakukan *stemming* berbahasa Indonesia. Contoh kata yang akan di *stemming* seperti “menulis”, “menuliskan”, “dituliskan” akan ditransformasikan menjadi kata “tulis”.

2.4. Support Vector Machine

Support Vector Machine (SVM) dalam *machine learning* merupakan pembelajaran *supervised* dengan algoritma pembelajaran yang *associated*. SVM bekerja dengan cara mencari sebuah *hyperlane* atau garis pembatas pemisah antar kelas yang mempunyai margin atau jarak antar *hyperlane* dengan data paling terdekat pada setiap kelas yang paling besar (Somantri and Apriliani 2018). Hal ini dikarenakan SVM memiliki kemampuan untuk menemukan solusi global yang optimal. Dengan margin yang besar maka kemungkinan model yang dihasilkan *overfit* menjadi kecil (Agastya 2018).

Umumnya, masalah yang ada di dunia nyata mempunyai bentuk non-linearly separable, sehingga *class* tidak dapat dipisahkan oleh *hyperlane* secara sempurna. Maka dari itu, diperlukan modifikasi SVM dengan memasukkan fungsi *kernel* (Haryanto, Muflikhah, and Fauzi 2018). Fungsi *kernel* memberikan kemudahan dalam proses pembelajaran SVM untuk menentukan *support vector machine* (Haryanto, Muflikhah, and Fauzi 2018). Fungsi *kernel* yang dapat dirumuskan pada persamaan 2.1.

$$K(x_i . x_j) = \Phi(x_i) . \Phi(x_j) \quad \dots(2.1)$$

Fungsi *kernel* tersebut untuk menyelesaikan masalah *non-linear*, data x_i dan x_j berawal dari pemetaan data oleh fungsi Φ ke ruang vektor yang tinggi. Sehingga kelas dapat dipisahkan secara linear oleh sebuah *hyperlane*, Lalu data x_i dan x_j akan dilakukan *dotproduct*. Karena umumnya transformasi ini tidak diketahui, dan sangat sulit untuk dipahami secara mudah, maka perhitungan dot product dapat digantikan dengan fungsi kernel $K(x_i . x_j)$ yang mendefinisikan secara implisit transformasi Φ (Munawarah, Soesanto, dan Faisal 2016).

Dalam penelitian ini akan melakukan klasifikasi dengan *support vector machine* menggunakan kernel Gaussian RBF. Penggunaan kernel Gaussian RBF dikarenakan bisa mendapatkan nilai akurasi yang lebih baik dibandingkan fungsi kernel lainnya. Menurut penelitian yang terkait fungsi kernel *Support Vector Machine* digunakan untuk klasifikasi penyakit tanaman kedelai dan didapatkan fungsi kernel Gaussian RBF mendapatkan hasil yang terbaik pada akurasinya dibandingkan fungsi

kernel linear (Feta and Ginanjar 2019). Fungsi $K(x_i . x_j)$ merupakan fungsi kernel yang menunjukkan pemetaan *non-linear* pada *feature space*. Untuk rumus fungsi *kernel* Gaussian RBF persamaan seperti yang dirumuskan pada persamaan 2.2.

$$K(x_i . x_j) = \exp\left(-\left(\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)\right) \dots(2.2)$$

2.5. Chi Square

Chi Square merupakan salah satu dari *feature selection* pada klasifikasi teks. Seleksi fitur ini dilakukan untuk mereduksi fitur-fitur yang tidak relevan dalam proses klasifikasi yang menggunakan teori statistika untuk menguji independensi sebuah term dengan kategorinya (Anisah, Honggowibowo, and Pujiastuti 2016). Chi Square termasuk metode dalam tipe seleksi fitur supervised, dimana mampu menghilangkan fitur-fitur dengan tanpa mengurangi dari tingkat akurasi yang dihasilkan (Somantri and Apriliani 2018).

Dalam penelitian ini yang digunakan dalam pemilihan *feature selection* dengan *chi square* (χ^2). Pada tahap ini, tiap kata yang diperoleh dihitung menggunakan persamaan 2.3

$$\chi^2(t, c) = \frac{N(AD - CB)^2}{(A + C)(B + D)(A + B)(C + D)} \dots(2.3)$$

Keterangan :

A = banyaknya dokumen dalam kategori c yang mengandung term t

B = banyaknya dokumen yang bukan kategori c tetapi mengandung term t

C = banyaknya dokumen dalam kategori c tetapi tidak mengandung term t

D = banyaknya dokumen yang bukan kategori c dan tidak mengandung term t

N = total keseluruhan dokumen

Nilai yang diperoleh pada setiap kata akan diperbandingkan dengan nilai kritis pada uji statistik *chi square*. Nilai kritis adalah batas ekstrim pada uji statistik yang diperoleh dari derajat kebebasan dan nilai taraf nyata atau *alpha*. Derajat kebebasan adalah banyaknya pengamatan bebas dari total pengamatan N, sedangkan taraf nyata atau *alpha* adalah batas kritis pada *chi square*. Nilai yang sudah diperbandingkan dengan nilai kritis akan menjadi fitur seleksi yang akan digunakan pada dataset yang akan digunakan.

2.6. Evaluasi Performa

Evaluasi performa yang dilakukan untuk menguji hasil dari klasifikasi dengan cara mengukur nilai performa dari sistem yang dibuat. Parameter pengujian yang digunakan untuk evaluasi adalah diperoleh dari tabel *confusion matrix* untuk perhitungan tingkat akurasinya. Berikut *confusion matrix* umum yang dapat dilihat pada table 1.

Tabel 2.1 *Confusion Matrix* (Agastya 2018)

	Terklasifikasi Positif	Terklasifikasi Negatif
Riil Positif	TP	FN
Riil Negatif	FP	TN

Tabel diatas untuk TP (*True Positif*), FP (*False Positif*), TN (*True Negatif*), FN (*False Negatif*). Adapun untuk menghitung akurasi pada persamaan 4,5,6,7.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad \dots(2.4)$$

$$Precision = \frac{TP}{(TP + FP)} \quad \dots(2.5)$$

$$Recall = \frac{TP}{(TP + FN)} \quad \dots(2.6)$$

$$F1 = 2 \cdot \frac{(Precision \times recall)}{(Precision + recall)} \quad \dots(2.7)$$