BAB II
TINJAUAN PUSTAKA

2.1. Penelitian Pendahulu

Sebagai bahan acuan dalam mengerjakan tugas akhir ini akan dipaparkan hasil penelitian pendahulu yang digunakan sebagai referensi oleh penulis diantaranya :

Dalam jurnal Jurusan Teknik Elektro, Fakultas Teknik, Universitas Negeri Semarang, Indonesia yang dilakukan oleh Hirroe Wijaya Ani Kesuma dan Feddy Setio Pribadi dengan judul "*Penerapan Cosine Similarity dalam Aplikasi Kitab Undang-Undang Hukum Dagang (Wetboek Van Koophandle Voor Indonesia)*". Penelitian ini bertujuan untuk membuat aplikasi pengganti kitap yang mampu mecari bagian pasal maupun ayat pada Kitab Undang Undang Hukum Dagang. Penelitian ini menggunakan metode *Cosine Similiarity* dengan besaran nilai uji kinerja sistem (performance measure) yang menghasilkan nilai 55,04% untuk recall, 63,33% untuk precission dan 56,93% untuk f-measure (Kesuma & Pribadi, 2016).

Dalam Jurnal Fakultas Teknologi Industri dan Informatika, Institut Teknologi Telkom Purwokerto yang dilakukan oleh Ade Riyani, Muhammad Zidny Naf'an, dan Auliya Burhanuddin dengan judul "Penerapan Cosine Similarity dan Pembobotan TF-IDF untuk Mendeteksi Kemiripan Dokumen". Pada penelitian ini memiliki tujuan untuk mendeteksi kemiripin suatu dokumen dengan dokumen lain agar tingkat plagiarism antar dokumen dapat diketahui. Penelitian ini menngunakan TF-IDF dalam pembobotan dan metode *Cosine Similiarity* untuk menghitung kemiripan suatu dokumen. Dari hasil pengujian dan analisis pada penelitian ini maka dapat disimpulkan bahwa algoritma cosine similarity dan pembobotan TF-IDF telah berhasil mendeteksi kemiripan suatu dokumen. Proses stemming pada preprocessing sangat berpengaruh terhadap nilai kemiripan hasil jika dilakukan stemming lebih tinggi. Nilai rata-rata perbedaan nilai kemiripan saat dilakukan stemming dan tidak dilakukan stemming adalah 10%. Kekurangan jika dilakukan proses stemming adalah waktu untuk

pemrosesan lebih lama dibandingkan tidak dilakukan proses stemming. (Riyani, Naf'an, & Burhanuddin, 2019).

2.2. Landasan Teori

Pada sub bab ini dijelaskan mengenai dasar teori atau tinjauan pustaka yang berkaitan dengan topik penelitian tugas akhir dan digunakan untuk mendukung penyelesaian masalah. Pokok –pokok yang akan dibahas mengenai landasan umum dari bahasn laporan. Sebagai berikut :

2.2.1. Text Mining

Text Mining adalah proses analisa teks yang biasanya bersumber dari suatu dokumen dengan tujuan mencari kata– kata yang dapat mewakili seluruh isi dari sebuah dokumen. Dari kata – kata tersebut kemudian dianalisa keterhubungan, keterkaitan dan kelas antar dokumen. Dokumen yang digunakan dapat bersumber dari dokumen Word, PDF, kutipan teks, media sosial, SMS dan lainnya. (Hartanto, 2017).

Dalam *text mining* terdapat beberapa Langkah yang dilakukan sebagai berikut :

1. *Tokenizing,* merupakan proses menguraikan deskripsi yang semula berupa kalimat menjadi kata. Kata diperoleh dengan cara memotong kalimat menggunakan karakter spasi sebagai pemisah antar kata (Melita, Amrizal, Suseno, & Dirjam, 2018). Pada proses *tokenizing* juga dilakukan proses *case folding* yakni proses mengubah seluruh karakter huruf besar menjadi huruf kecil.
2. *Tagging/Stemming*, merupakan tahap untuk mencari kata dasar atau kata inti dari sebuah kata (Anna & Hendin, 2018). Misalkan kata “menabung” dan “tabungan” yang akan dirubah menjadi “tabung”, tujuan dilakukannya tahap ini adalah untuk mengambil inti pembahasan dari sebuah kata.
3. *Filtering*, merupakan tahap dalam *text mining* yang bertujuan untuk membuang kata yang memiliki partisipasi kecil dalam membangun makna kalimat (Anna & Hendin, 2018). Misalnya kata “aku”, “apa”, “asal”, kata penghubung dan lain sebagainya. Proses *filtering*

menggunakan *stop word list* yang berisi kumpulan kata – kata yang tidak unik.

4. *Analysing*, merupakan tahap seberapa jauh keterhubungan antar suatu kata(*term*) terhadap suatu dokumen atau kalimat dengan menghitung nilai/bobot keterhubungan. Algoritma TF/IDF digunakan dalam proses perhitungan bobot (W) terminologi kata. Algoritma ini digunakan untuk menghitung bobot setiap kata yang paling umum digunakan pada information retrieval. Metode ini juga terkenal efisien, mudah dan memiliki hasil yang akurat. (Ratniasih, Sudarma, & Gunantara, 2017).

2.2.2. Vector Space Model

Vector Space Model (VSM) adalah salah satu model aljabar yang berfungsi untuk mengukur kemiripan kata pada masing-masing dokumen. Dalam model ini akan direpresentasikan dengan *term frequency* (Anggeriani, Fitria, & Hafid, 2017). Jika suatu kata tidak terdapat pada suatu dokumen yang lain akan disimbolkan dengan nilai nol. VSM biasanya digunakan dalam penyaringan informasi, temu balik informasi, pengindeksan, dan perankingan relevansi. Alur kerja VSM diilustrasikan pada gambar 2.1.

	$Kata_1$	$Kata_2$	$Kata_3$
$Ulasan_1$	$Bobot_{11}$	$Bobot_{21}$	$Bobot_{31}$
$Ulasan_2$	$Bobot_{12}$	$Bobot_{22}$	$Bobot_{32}$
$Ulasan_3$	$Bobot_{13}$	$Bobot_{23}$	$Bobot_{33}$
....	....	....	....
$Ulasan_n$	$Bobot_{kn}$	$Bobot_{kn}$	$Bobot_{kn}$

Gambar 2.1 Alur kerja *Vector Space Model*

Term Frequency (TF) – *Invers Document Frequency* (IDF) merupakan sebuah skema pembobotan yang sering digunakan dalam VSM bersama dengan *cosine similarity* untuk menentukan kesamaan antara dua buah dokumen. TF-IDF mempertimbangkan frekuensi kata - kata yang berbeda dalam semua dokumen dan mampu membedakan dokumen. Dalam *Vector Space Model*, setiap *vector*

disusun oleh term dan bobot yang mewakili dokumen. (Triana, Saptono, & Sulistyo, 2014)

Metode ini menggabungkan dua konsep untuk perhitungan bobot, yaitu frekuensi kemunculan sebuah kata didalam sebuah dokumen tertentu dan inverse frekuensi dokumen yang mengandung kata tersebut. Frekuensi kemunculan kata di dalam dokumen yang diberikan menunjukan seberapa penting kata tersebut di dalam dokumen. Metode ini akan menghitung bobot setiap term di dokumen dengan rumus (Riyani, Naf'an, & Burhanuddin, 2019):

$$W_{d,t} = tf_{d,t} x IDF_{d,t} + 1 \quad (2.1)$$

Dengan:

$W_{d,t}$ = bobot dokumen dari *term*

$tf_{d,t}$ = *terms frequency* (jumlah kemunculan kata pada dokumen)

$IDF_{d,t}$ = *Inverse Document Frequency*

Untuk menghitung Bobot terlebih dahulu kita harus menghitung IDF dan *Term frequency. Term Frequency* adalah banyaknya kata/*term* yang terdapat pada dokumen. Selanjutnya untuk menghitung *Inverse Document Frequency* dapat dilihat pada persamaan 2.2 berikut:

$$IDF_{d,t} = log_2(\frac{D}{df_{d,t}}) \quad (2.2)$$

Dengan:

D = Jumlah seluruh dokumen termasuk dokumen *query*

$df_{d,t}$ = Jumlah dokumen yang mengadung *term* tersebut

2.2.3. *Cosine Similiarity*

Setelah pembobotan tiap term dilakukan, diperlukan perhitungan untuk melakukan perankingan untuk mengukur kemiripan antara vektor *query* (dokumen yang dites kemiripannya) dan vektor dokumen yang akan dibandingkan dalam hal ini adalah data latih. Salah satu metode yang biasa digunakan dalam perhitungan kemiripan adalah pengukuran *cosine*, yang menentukan sudut antara vektor dokumen dan vektor query dan didefinisikan sebagai berikut:

$$sim\ (\vec{d}_j, \vec{q}) = \frac{\vec{d_j}.\vec{q}}{\|\vec{d_j}\|.\|\vec{q}\|} = \frac{\sum_{i=1}^{t}(w_{ij}.w_{iq})}{\sqrt{\sum_{i=1}^{t} w^2{}_{ij} \cdot \sum_{i=1}^{t} w^2{}_{iq}}} \quad (2.3)$$

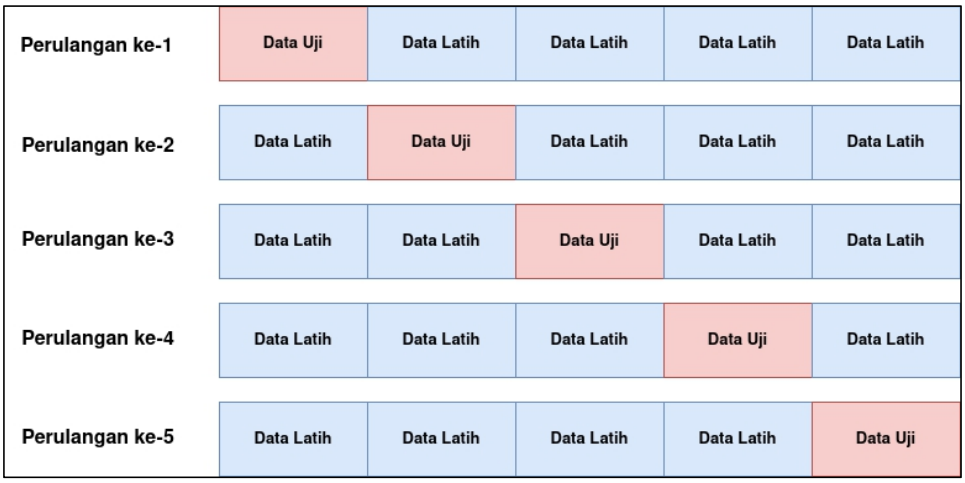
Dimana W adalah bobot dari term t pada dokumen, penyebut dalam persamaan ini disebut faktor normalisasi yang berfungsi untuk menghilangkan pengaruh panjang dokumen atau di sini disebut sebagai m*agnitude* sedangkan untuk pembilangnya adalah nilai dari *dot product*. Normalisasi ini diperlukan karena dimana dokumen panjang akan cenderung memiliki nilai lebih besar karena memiliki frekuensi kemunculan kata yang besar pula. (Triana, Saptono, & Sulistyo, 2014).

Nilai dari *Cosine Similiarity* berkisar mulai dari nol sampai satu, nol (0) artinya dokumen tersebut tidak memiliki kemiripan sama sekali dan satu (1) yang berarti dokumen tersebut identik.

2.2.4. *K-Fold Cross Validation*

Cross Validation atau estimasi rotasi merupakan salah satu teknik validasi yang bertujuan untuk melakukan prediksi model dan memperkirakan seberapa akurat sebuah model prediktif ketika dijalankan dalam praktiknya (Azis, Purnawansyah, Fattah, & Putri, 2020). Salah satu teknik dari validasi silang adalah *K-Fold Cross Validation*.

K-Fold Cross Validation merupakan teknik yang dapat digunakan apabila memiliki jumlah data yang terbatas (jumlah instance tidak banyak). *K-Fold Cross Validation* merupakan salah satu metode untuk mengetahui persentase keberhasilan suatu sistem dengan melakukan proses perulangan dengan mengacak atribut data latih dan data uji sehingga sistem tersebut akan lebih teruji (Pitria, 2014). *K-Fold Cross Validation* sendiri merupakan teknik membagi data sejumlah K bagian set data dengan ukuran yang sama sesuai dengan yang diinginkan. Alur kerja *Cross Validation* diilustrasikan pada gambar 2.2 (Azis, Purnawansyah, Fattah, & Putri, 2020).



Gambar 2.2 Alur kerja *Cross Validation*

Dalam proses ini dilakukan pengkombinasian antara data yang digunakan sebagai data latih dan data yang digunakan sebagai data uji. Tujuannya adalah untuk mencari tau akurasi dari masing masing kombinasi yang akan menentukan performa dari sistem prediksi yang dibuat.

2.2.5. *Stemming Library*

Proses *stemming* disini merupakan langkah awal dalam penerapan metode *cosine similiarity* tujuannya adalah untuk mengekstrak kata dasar pada masing - masing kata dalam suatu kalimat. Pada penelitian ini, peneliti menggunakan *library* untuk membantu proses *stemming* dengan menggunakan salah satu *package* dari node.js yang bernama sastrawijs. *Package* ini memiliki fungsi untuk melakukan proses *stemming,* dan algoritma *stemming* yang dipakai adalah Nazief dan Adriani.