BAB II
LANDASAN TEORI

1.1 Sistem KPTA FTI UII

Sistem KPTA FTI UII adalah sistem yang meng-*handle* pencatatan, pendaftaran, serta manajemen data Kerja Praktek dan Tugas Akhir mahasiswa Fakultas Teknologi Industri. Dari mulai pendaftaran akun, pendaftaran tugas akhir, *progress* dan ujian pendadaran, semua di-*handle* dalam sistem. Sistem ini telah aktif semenjak Juni 2017. Sistem KPTA FTI UII baru aktif dan digunakan oleh jurusan Teknik Informatika UII (Laksita, 2016).

1.2 *Text Mining*

Text mining (penambangan teks) adalah penambangan yang dilakukan oleh komputer untuk mendapatkan sesuatu yang baru, sesuatu yang tidak diketahui sebelumnya atau menemukan kembali informasi yang tersirat secara implisit, yang berasal dari informasi yang diekstrak secara otomatis dari sumber-sumber data teks yang berbeda-beda (Feldman & Sanger, 2007). Pada dasarnya proses kerja dari *text mining* banyak mengapdopsi 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. Tahap-tahap *text mining* secara umum adalah *text preprocessing* dan *feature selection* (Feldman & Sanger, 2007).

1.2.1 *Text Preprocessing*

Tahapan ini bertujuan untuk mempersiapkan teks menjadi data yang siap untuk diproses pada tahapan berikutnya (Mahmudy & Widodo, 2014), untuk proses *mining* yang lebih lanjut (perhitungan bobot, peringkasan, klasifikasi dokumen, dan lain-lain). Artibut atau data yang digunakan untuk proses klasifikasi adalah data Judul TA dan konsentrasi . Tahapan dari *text preprocessing* adalah sebagai berikut:

1. *Case Folding*

   Pada tahap *case folding* merupakan proses penyeragaman bentuk huruf yaitu dengan mengubah semua huruf menjadi huruf kecil (Hidayatullah & Sn, 2014).

2. *Remove Punctuation*

Pada tahap *remove punctuation* adalah tahap menghapus tanda baca (*unicode*) dalam satu kalimat.

3. *Tokenizing*

   Pada tahap *tokenizing* adalah tahap pemotongan *string input* berdasarkan tiap kata yang menyusunnya (Hidayatullah & Sn, 2014). Dengan menggunakan karakter spasi atau tanda baca sebagai pemisah.

4. *Stopword Removal*

   Stopwords adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan *bag-of-words*. Contoh *stopwords* adalah "yang", "dan", "di", "dari" dan seterusnya (Wibisono & Khodra, 2005). Sebelum proses *stopword removal* dilakukan, harus dibuat daftar *stopword* (*stoplist*). Jika termasuk di dalam *stoplist* maka kata-kata tersebut akan dihapus dari deskripsi sehingga kata-kata yang tersisa di dalam deskripsidianggap sebagai kata-kata yang mencirikan isi dari suatu dokumen atau kata kunci. Tahapan ini terbukti dapat meningkatkan tingkat akurasi (Darujati & Gumelar, 2012) .

1.2.2 *Feature Selection*

Salah satu metode pembobotan adalah TF-IDF (*Term Frequency-Inverse Document Frequency)* merupakan metrik yang umum digunakan dalam proses kategorisasi teks (Hidayatullah & Sn, 2014). Metode ini juga terkenal efisien, mudah dan memiliki hasil yang akurat. Metode ini akan menghitung nilai *Term Frequency* (*TF*) dan *Inverse Document Frequency* (*IDF*) pada setiap token (kata) di setiap dokumen dalam korpus. Nilai bobot suatu kata (*term*) menyatakan kepentingan bobot tersebut dalam merepresentasikan data. Pada pembobotan TF-IDF, bobot akan semakin besar jika frekuensi kemunculan kata semakin tinggi, tetapi bobot akan berkurang jika kata tersebut semakin sering muncul pada data lainnya. Metode ini akan menghitung bobot setiap token di dokumen dengan rumus:

$$idf = \log(\frac{N}{df}) \qquad (1.1)$$

N = jumlah judul TA

df = Banyaknya judul TA dimana suatu kata (*term*) muncul

1.3 *Naïve Bayes Classifier*

Dalam algoritma *Naïve Bayes classifier* setiap dokumen direpresentasikan dengan pasangan atribut "x_1, x_2, x_3,...x_n" dimana x_1 adalah kata pertama, x_2 adalah kata kedua dan seterusnya. Sedangkan V adalah himpunan kategori konsentrasi. Pada saat klasifikasi algoritma akan mencari probabilitas tertinggi dari semua kategori dokumen yang diujikan (V_{MAP}), di mana persamaannya adalah sebagai berikut :

$$V_{MAP} = \frac{P(x_1, x_2, x_3, \dots x_n|Vj)P(Vj)}{P(x_1, x_2, x_3, \dots x_n|Vj)} \quad (1.2)$$

Untuk P(x_1, x_2, x_3,...x_n) nilainya konstan untuk semua kategori (Vj) sehingga persamaan dapat ditulis sebagai berikut :

$$V_{MAP} = P(x_1, x_2, x_3, \dots x_n|Vj)P(Vj) \quad (1.3)$$

Persamaan diatas dapat disederhanakan menjadi sebagai berikut :

$$V_{MAP} = P(x_1|Vj)P(Vj) \quad (1.4)$$

Keterangan :

V_j = Kategori konsentrasi j =1, 2, 3,…n. Di mana dalam penelitian ini

j_1 = kategori konsentrasi Sistem Informasi

j_2 = kategori konsentrasi Multimedia

j_3 = kategori konsentrasi Jaringan dan Keamanan Komputer

j_4 = kategori konsentrasi Informatika Teori Dan Sistem Cerdas

j_5 = kategori konsentrasi Informatika Medis

j_6 = kategori konsentrasi Rekayasa Perangkat Lunak

$P(x_i|V_j)$ = Probabilitas x_i pada kategori V_j

$P(V_j)$ = Probabilitas dari V_j

Untuk $P(V_j)$ dan $P(x_i|V_j)$ dihitung pada saat pelatihan di mana persamaannya adalah sebagai berikut :

$$P(V_j) = \frac{|\text{docs j}|}{|\text{contoh}|} \quad (1.5)$$

$$P(xi|Vj) = \frac{nk+1}{n+|kosakata|} \quad (1.6)$$

Keterangan :

|docs j| = jumlah dokumen setiap kategori j

|contoh| = jumlah dokumen dari semua kategori

n_k = jumlah frekuensi kemunculan setiap kata

n = jumlah frekuensi kemunculan kata dari setiap kategori

|kosakata| = jumlah semua kata dari semua kategori

1.4 *Text Similarity*

Text Similarity memiliki banyak aplikasi dan telah dipelajari secara ekstensif di *Natural Language Processing* (NLP) dan komunitas *Information retrieval* (IR). Misalnya, kombinasi *corpus* dan metode berbasis pengetahuan telah diciptakan untuk menilai kesamaan kata.(Yih, et al., 2011)

1.4.1 Cosine Similarity

Metode *Cosine Similarity* merupakan metode yang digunakan untuk menghitung *similarity* (tingkat kesamaan) antar dua buah objek. Secara umum penghitungan metode ini didasarkan pada *vector space similarity measure*. Metode *cosine similarity* ini menghitung *similarity* antara dua buah objek (misalkan D1 dan D2) yang dinyatakan dalam dua buah *vector* dengan menggunakan *keywords* (kata kunci) dari sebuah dokumen sebagai ukuran. (Nurdiana, et al., 2016)

$$Cosine = \frac{\sum_{n=1}^{j} nA \; x \; nB}{\sqrt{\sum_{n=1}^{j}(nA)^2} \; \sqrt{\sum_{n=1}^{j}(nB)^2}} \quad (2.7)$$

Keterangan:

j = |A ∩ B|

nA = jumlah kemunculan kata indeks ke-n dari daftar kata pada kalimat A.

nB = jumlah kemunculan kata indeks ke-n dari daftar kata pada kalimat B.

1.5 Evaluasi Performa Klasifikasi

Evaluasi performa klasifikasi dilakukan untuk mengetahui tingkat ketepatan klasifikasi yang dibuat. Metode evaluasi klasifikasi yang digunakan dalam penelitian ini adalah *Holdout*. Dalam metode ini menggunakan sebanyak setengah atau dua per tiga dari data keseluruhan untuk

keperluan proses *training* sedangkan sisanya digunakan untuk keperluan *testing* (Hidayatullah & Ma'arif, 2016). Secara khusus dua pertiga dari data dialokasikan dalam kelompok data *training*, dan sepertiga sisanya ke dalam kelompok data *testing*. Data *training* digunakan untuk memperoleh model, dan akurasinya diestimasi menggunakan data *testing*.

Pengujian lain dilakukan dengan menggunakan *confusion matrix*. *Confusion matrix* merupakan salah satu tools penting dalam metode visualisasi yang digunakan pada mesin pembelajaran yang biasanya memuat dua kategori atau lebih (Weighting, 2007) akurasi dianalisis menggunakan suatu matriks kontingensi yaitu suatu matriks bujur sangkar yang memuat jumlah piksel yang diklasifikasi dan tersusun. *Confusion matrix* bertujuan untuk menentukan nilai akurasi dari hasil klasifikasi judul TA. Adapun evaluasi pengukuran kinerja dari sistem mengunakan metode perhitungan menggunakan *precision, recall,* dan *accuracy*. *Precision* adalah tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. *Recall* adalah tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi. Sedangkan *accuracy* adalah tingkat kedekatan antara nilai prediksi dengan nilai aktual. Contoh *confusion matrix* ditunjukan pada Tabel 3.6.

Tabel 3.1 *Confusion Matrix* Klasifikasi Judul TA

		Kelas Prediksi	
		1	0
Kelas Sebenarnya	1	TP	FN
	0	FP	TN

Keterangan untuk tabel 3.6 dinyatakan sebagai berikut:

True Posstive (TP): jumlah dokumen dari kelas 1 benar dan diklasifikan sebagai kelas 1.

True Negative (TN): jumlah dokumen dari kelas 0 benar diklasifikasikan sebagai kelas 0.

False Positive (FP): jumlah dokumen dari kelas 0 salah diklasifikasikan sebagai kelas 1.

False Negative (FN): jumlah dokumen dari kelas 1 salah diklasifikasikan sebagai kelas 0.

Dari *confusion matrix* dapat diperoleh:

$$Precision = \frac{TP}{TP+FP} \times 100\% \quad (2.8)$$

$$Recall = \frac{TP}{TP+TN} \times 100\% \quad (2.9)$$

$$Accuracy = \frac{TP+TN}{TP+FN+FP+TN} \times 100\% \quad (2.10)$$

1.6 Penelitian Terkait

1.6.1 *Text Classification*

Berikut adalah penelitian yang menjadi referensi penulis nutuk menentukan metode untuk *text classification* :

(Hidayatullah & Ma'arif, 2016) melakukan klasifikasi konsentrasi skripsi yang menggunakan metode *Naive Bayes Classifier* (NBC) dan *Support Vector Machine* (SVM) dengan hasil yang didapat metode NBC lebih baik dibandikan dengan metode SVM dengan akurasi 98% sementara SVM hanya 73%. Namun penelitian tersebut hanya sebatas pemodelan sistem dan pada penelitian ini penulis akan melakukan implementasi untuk klasifikasi konsentrasi TA dengan menggunakan metode *Naïve Bayes Classifier* (NBC).

(Hamzah, 2012) melakukan klasifikasi teks dengan *Naive Bayes Classifier* (NBC) untuk pengelompokan teks berita dan akdemis. Hasil yang didapat dari penelitian ini bahwa algoritma NBC memiliki kinerja yang cukup tinggi untuk klasifikasi dokumen teks, baik dokumen berita maupun dokumen akademik. Pada klasifikasi dokumen berita didapatkan akurasi yang lebih tinggi (maksimal 91%) dibandingkan dengan dokumen akademik (maksimal 82%). Dari hasil penelitian ini semakin menguatkan penulis untuk menggunakan metode *Naïve Bayes Classifier* (NBC).

(Kurniawan, Effendi, & Sitompul, 2012) melakukan klasifikasi konten berita dengan metode *Text Mining*, Setelah melakukan studi literatur, perancangan, analisis, implementasi dan pengujian aplikasi pengklasifikasian berita secara otomatis dengan menggunakan metode *Naïve Bayes Classifier* (NBC), maka dapat disimpulkan aplikasi ini sudah mampu melakukan proses klasifikasi data berita secara otomatis dan proses klasifikasi semakin akurat jika data latih yang digunakan dalam pembelajaran berjumlah banyak. Untuk penelitian berikutnya diharapkan sistem ini tidak hanya untuk mengklasifikasi berita melainkan bisa juga digunakan untuk mengklasifikasikan dokumen lain seperti kesenian, olahraga, dan jurnal. Hasil penelitian ini

semakin menguatkan penulis untuk menggunakan algoritma NBC di mana penelitian ini sudah mampu melakukan klasifikasi secara otomatis.

1.6.2 *Text Similarity*

Berikut adalah penelitian yang menjadi referensi penulis nutuk menentukan metode untuk *text similarity*:

(Nurdiana et al., 2016) melakukan perbandingan *cosine*, *jaccard* dan *k-nearest neighbor(K-NN).* Dari penelitian ini didapat hasil Metode *cosine*, *jaccard* dan *k-nearest neighbor* (K-NN) yang digunakan pada proses klasifikasi dokumen *teks* dengan hasil akhir dari percobaan 33 kali dengan *key* yang berbeda dan total 6326 dokumen didapat metode *cosine* yang nilai kemiripannya tertinggi yaitu 41% dari metode *jaccard* 19% dan *k-nearest neighbor* (K-NN) 40%. Berdasarkan penelitian ini penulis memutuskan untuk menggunakan algoritma *cosine similarity* untuk perhitungan kemiripan proposal TA.

Dari beberapa penelitian tentang *text classification* dan *text similarity* di atas maka akan dilakukan penelitian klasifikasi judul TA dengan metode *Naïve Bayes* dan perhitungan kemiripan proposal TA dengan menggunakan metode *cosine similarity*.