BAB II

KERANGKA TEORETIK, KERANGKA BERPIKIR DAN HIPOTESIS PENELITIAN

2.1. Kajian Teoritis

2.1.1. *Information Retrieval* (IR)

Pengertian dari kata *Information retrieval*dapat berarti sangat luas. Dalam kasus pembelajaran, arti kata *Information retrieval* adalah menemukan suatu bahan yang biasanya berbentuk dokumen dari suatu data yang tidak terstruktur yang dapat memenuhi kebutuhan informasi dari sebuah penyimpanan yang besar dan biasanya disimpan didalam komputer (Manning, 2009: 1).

Dunia sudah berkembang dengan sangat pesat, dimana dahulu hanya sedikit orang yang melakukan pencarian informasi, dan sekarang ratusan juta orang terlibat dalam melakukan pencarian informasi. Keyword yang tepat sangat berpengaruh pada *information retrieval*. Bahkan dengan keyword yang tepat, terkadang banyak informasi yang tidak relevan ikut muncul. Informasi atau data yang dicari dapat berupa berupa teks, image, audio, video dan lain-lain.

Menurut Hasibuan(1996: 42), diacu dalam Hasugian (2006: 3) “Secara garis besar komponen STBI terdiri dari pemakai (*user*), dokumen, dan pencocokan (*matching*)”.

1. *User* adalah orang yang menggunakan atau memanfaatkan sistem *information retrieval*dalam rangka kegiatan pengelolaan dan pencarian informasi.

2. Dokumen merupakan istilah yang digunakan untuk seluruh bahan pustaka seperti artikel, buku, laporan penelitian dan sebagainya. Seluruh bahan pustaka dapat disebut sebagai dokumen.
3. Pencocokan (*matching*) adalah proses membandingkan antara istilah yang tercantum dalam peryataan pemakai (query) dengan istilah yang tercantum dalam dokumen.

Permasalahan yang muncul saat ini adalah bagaimana caranya agar mendapatkan informasi yang relevan dalam melakukan pencarian informasi dalam suatu koleksi dokumen. Masalah tersebut memicu banyak orang untuk melakukan penelitian terhadap kasus *information retrieval*, khususnya informasi berbentuk teks. Untuk menanggulangi masalah ini, diperlukan suatu sistem yang dapat mempermudah proses *information retrieval* agar pengguna dapat dengan mudah mendapatkan informasi yang banyak serta relevan sesuai dengan kebutuhan pengguna.

2.1.2 Klasifikasi Dokumen

Klasifikasi dokumen adalah proses pengelompokan dokumen sesuai dengan kategori yang dimilikinya. (Trisedya dan Jais, 2009: 1). Dengan proses klasifikasi, setiap dokumen akan dikelompokkan menjadi beberapa kelompok. Dokumen yang sudah diklasifikasi, akan lebih mudah dicari oleh user karena dokumen tersebut terorganisir dengan baik, seperti tujuan dari klasifikasi dokumen itu sendiri yaitu untuk mempermudah user dalam melakukan pencarian dokumen.

Pengklasifikasian dokumen otomatis sudah banyak diterapkan pada aplikasi-aplikasi yang sering kita gunakan, contohnya yaitu spam filtering pada email.

Aplikasi ini bekerja dengan memperhatikan kata-kata yang terdapat dalam setiap isi email yang masuk, sehingga *spam filtering* menentukan bahwa email tersebut termasuk spam atau bukan spam.

Manfaat dari klasifikasi dokumen adalah untuk pengorganisasian dokumen. Dengan jumlah dokumen yang sangat besar, untuk mencari sebuah dokumen akan lebih mudah apabila dokumen yang dimiliki terorganisir dan telah dikelompokkan sesuai kategorinya masing-masing.

2.1.3 *TextMining*

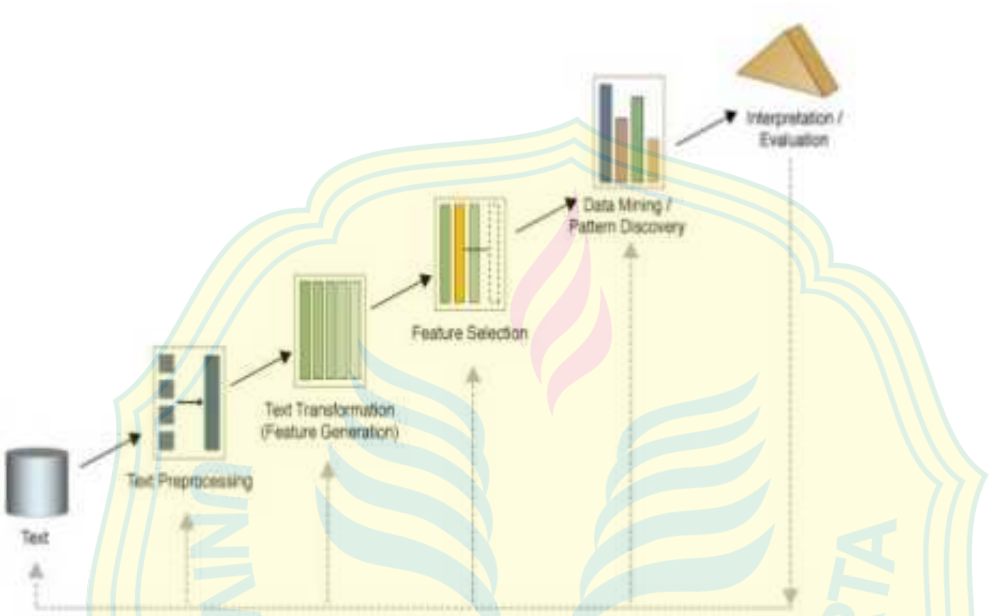
Text mining adalah salah satu bidang khusus dari *Data Mining*. *Text mining* dapat didefinisikan sebagai suatu proses menggali informasi dimana seorang user berinteraksi dengan sekumpulan dokumen menggunakan tools analisis yang merupakan komponen-komponen dalam *Data Mining* yang salah satunya adalah kategorisasi (Triawati, 2009).

Tujuan dari *Text mining* adalah untuk mendapatkan informasi yang berguna dari sekumpulan dokumen. Jadi, sumber data yang digunakan pada *Text mining* adalah kumpulan teks yang memiliki format yang tidak terstruktur atau minimal semi terstruktur. Adapun tugas khusus dari *text mining* antara lain yaitu pengkategorisasian teks (*text categorization*) dan pengelompokan teks (*text clustering*).

Permasalahan yang terdapat pada *text mining* sama dengan permasalahan yang terdapat pada *data mining*, yaitu jumlah data yang besar, dimensi yang tinggi, data dan struktur yang terus berubah, dan data noise. Perbedaan di antara keduanya adalah pada data yang digunakan. Pada *data mining*, data yang digunakan adalah

data yang terstruktur, sedangkan pada *text mining*, data yang digunakan pada umumnya adalah data yang tidak terstruktur, atau paling tidak semi terstruktur.

Untuk mengubah data pada *text mining*menjadi lebih terstruktur, diperlukan adanya beberapa proses text mining. Proses text mining dapat dijelaskan dengan gambar berikut.



Gambar 2.1. Proses Text Mining

Penjelasan proses text mining sesuai gambar 2.1 adalah sebagai berikut:

1. *Text*: Data pada *text mining* yang tidak terstruktur
2. *Text Preprocessing*:Tahap proses awal terhadap teks untuk mempersiapkan teks menjadi data yang akan diolah lebih lanjut. Sekumpulan karakter yang bersambungan (teks) harus dipecah-pecah menjadi unsur yang lebih berarti. Suatu dokumen dapat dipecah menjadi bab, sub-bab, paragraf, kalimat, bahkan kata. Parsing/Tokenizing adalah proses memecah teks menjadi kalimat atau kata (Feldman dan Sanger, 2007:60).

3. *Text Transformation*

   Tahapan yang dipergunakan untuk mengubah kata-kata ke dalam bentuk dasar, sekaligus untuk mengurangi jumlah kata-kata tersebut. Pendekatan yang dapat dilakukan itu dengan *stemming* dan *stopwords removal.*

4. *FeatureSelection*

   *FeatureSelection*adalah suatu kegiatan yang umumnya bisa dilakukan secara preprocessing dan bertujuan untuk memilih *feature* yang berpengaruh dan mengesampingkan *feature* yang tidak berpengaruh dalam suatu kegiatan pemodelan atau penganalisaan data. Secara garis besar ada dua kelompok besar dalam pelaksanaan *featureselection*: *RankingSelection* dan Subset *Selection*.

   1) *RankingSelection*

      Rankingselection secara khusus memberikan *ranking* pada setiap *feature* yang ada dan mengesampingkan *feature* yang tidak memenuhi standar tertentu. *Rankingselection* menentukan tingkat *ranking* secara independent antara satu *feature* dengan *feature* yang lainnya. *Feature* yang mempunyai *ranking* tinggi akan digunakan dan yang rendah akan dikesampingkan. *Rankingselection* ini biasanya menggunakan beberapa cara dalam memberikan nilai *ranking* pada setiap *feature* misalnya regression, correlation, mutual information dan lain-lain

   2) *Subset Selection*

      Subset selection adalah metode *selection* yang mencari suatu set dari *feature*s yang dianggap sebagai optimal *feature*. Ada tiga jenis metode

yang bisa digunakan yaitu *selection* dengan tipe *wrapper*, *selection* dengan tipe filter dan *selection* dengan tipe embedded.

1. *Feature Selection* Tipe*Wrapper.*

   Tipe ini melakukan *featureselection* dengan melakukan pemilihan bersamaan dengan pelaksanaan pemodelan. *Selection* tipe ini menggunakan suatu kriteria yang memanfaatkan rata-rata klasifikasi dari metode pengklasifikasian/pemodelan yang digunakan. Untuk mengurangi biaya komputasi, proses pemilihan umumnya dilakukan dengan memanfaatkan rata-rata klasifikasi dari metode pengklasifikasian/pemodelan untuk pemodelan dengan nilai terendah (misalnya dalam kNN, menggunakan nilai k terendah).

2. *Feature Selection* Tipe *Filter*:

   Featureselection dengan tipe filter hampir sama dengan *selection* tipe *wrapper,* yaitu dengan menggunakan *intrinsic statistical properties* dari data. Tipe filter berbeda dari tipe *wrapper* dalam hal pengkajian *feature* yang tidak dilakukan bersamaan dengan pemodelan yang dilakukan. *Featureselection*tipe *filter*ini dilakukan dengan memanfaatkan salah satu dari beberapa jenis filter yang ada.

3. *Feature Selection* Tipe *Embedded*:

   Featureselection jenis ini memanfaatkan suatu *learning machine* dalam proses *featureselection*. Dalam sistem *selection* ini, *feature*

secara natural dihilangkan, apabila *learning machine* menganggap *feature* tersebut tidak begitu berpengaruh.

Walaupun tahap sebelumnya sudah melakukan penghapusan kata-kata yang tidak deskriptif (*stopwords*), namun tidak semua kata-kata di dalam dokumen memiliki arti penting. Oleh karena itu, untuk mengurangi dimensi, pemilihan hanya dilakukan terhadap kata-kata yang relevan yang benar-benar merepresentasikan isi dari suatu dokumen.

5. *Pattern Discovery*

   Pattern Discovery(tahapan penemuan) adalah tahap terpenting dari keseluruhan proses *text mining,* yaitu penemuan pola atau pengetahuan dari keseluruhan teks.

6. *Interpretation/Evaluation*

   Hasil akhir dari keseluruhan proses untuk melihat apakah hasil tersebut relevan atau tidak.

2.1.4 Ekstraksi Dokumen

Ekstraksi dokumen dilakukan terlebih dahulu sebelum melakukan pembobotan kata. Dalam proses ekstraksi, dilakukan *pre-processing* yang umum dilakukan pada dokumen. Tujuan dari pre-processing adalah untuk melakukan pembuangan karakter dan kata yang tidak perlu dari suatu dokumen, yang dapat mempengaruhi kualitas pengelompokan. Tahap pre-processing yang digunakan dalam penelitian ini, yaitu: *Case folding, tokenizing, dan filtering*. Berikut adalah susunan pre-processing pada dokumen:

1. *Case folding*

   Case folding adalah tahapan yang mengubah semua huruf dalam dokumen menjadi huruf kecil. karakter selain huruf dihilangkan dan dianggap *delimiter*(pembatas).

2. *Tokenizing*

   Tahap *tokenizing* adalah tahap pemecahan kalimat yang ada di dalam sebuah file menjadi kata. Spasi digunakan untuk memisahkan antar kata tersebut.

3. *Filtering*

   Filtering adalah tahap mengambil kata-kata penting dari hasil *tokenizing*. Bisa menggunakan algoritma *stoplist* (membuang kata yang kurang penting) atau *wordlist* (menyimpan kata penting).

2.1.5 TF-IDF

Metode TF-IDF merupakan metode untuk menghitung bobot setiap kata yang paling umum digunakan pada *information retrieval*(Robertson, 2004). Metode ini juga terkenal efisien, simpel dan memiliki hasil yang akurat (Ramos, 2010). Metode ini akan menghitung nilai TF (Term Frequency) dan IDF (Inverse Document Frequency) pada setiap kata dalam setiap dokumen.

Metode TF-IDF yang akan digunakan dalam aplikasi dilakukan sedikit modifikasi. Istilah dokumen pada metode TF-IDFakan diganti menjadi kategori. Metode TF-IDF yang diterapkan bukan menghitung bobot setiap kata untuk information retrieval, melainkan untuk pengkategorian dokumen karya akhir.

Term Frequency(TF) adalah proses pertama yang dilakukan pada algoritma TF-IDF. TFyang diterapkan dalam sistem, merupakan pengukuran frekuensi

munculnya kata dalam suatu kategori. Kemudian TF dikombinasikan dengan *Inverse Document Frequency* untuk mencari kategori yang paling relevan dengan abstrak yang di-*input*.

*Inverse Document Frequency*pada sistem ini merupakan perhitungan untuk mencarinilai statistik frekuensi kemunculan suatu kata dalam keseluruhankategori. Perhitungan dilakukan dengan mengkalkulasi total kategoriyang ada, dibagi dengan jumlah kategori yang mengandung kata tertentu.

Pada algoritma TF-IDF yang digunakan dalam sistem ini, rumus untuk menghitung bobot kata abstrak yang di-*input*terhadap *database* tiap kategori adalah sebagai berikut :

$$W_{tk} = TF_{t,k} * IDF_t \quad \text{.....(1)}$$

Dimana rumus IDF:

$$IDF_t = \log\left(\frac{D}{DF_t}\right) \quad \text{.....(2)}$$

Keterangan:

W_{tk} = Bobot kata ke-t terhadap kategori ke-k

$TF_{t,k}$ = Jumlah kemunculan kata ke-t terhadap kategori ke-k

IDF_t = Nilai IDF kata ke-t

DF_t = Jumlah kategori yang memuat kata ke-t

D = Jumlah kategori pada aplikasi

k = Kategori ke-k

t = Kata ke-t dari abstrak yang di-*input*

Setelah mendapatkan bobot seluruh kata terhadap kategori, maka semua bobot TF-IDF tersebut akan diakumulasi pada masing-masing kategori yang menghasilkan frekuensi terhadap abstrak yang di-*input.* Kategori dengan

akumulasi nilai TF-IDF tertinggi merupakan kategori yang dihasilkan oleh aplikasi.

2.2. Kerangka Berpikir

Berdasarkan kajian teori yang telah dijelaskan diatas, permasalahan dalam sistem klasifikasi karya akhir di kalangan universitas belum sepenuhnya teratasi, untuk itu diperlukan aplikasi yang dapat membantu sistem klasifikasi karya akhir di Jurusan Teknik Elektro Universitas Negeri Jakarta.

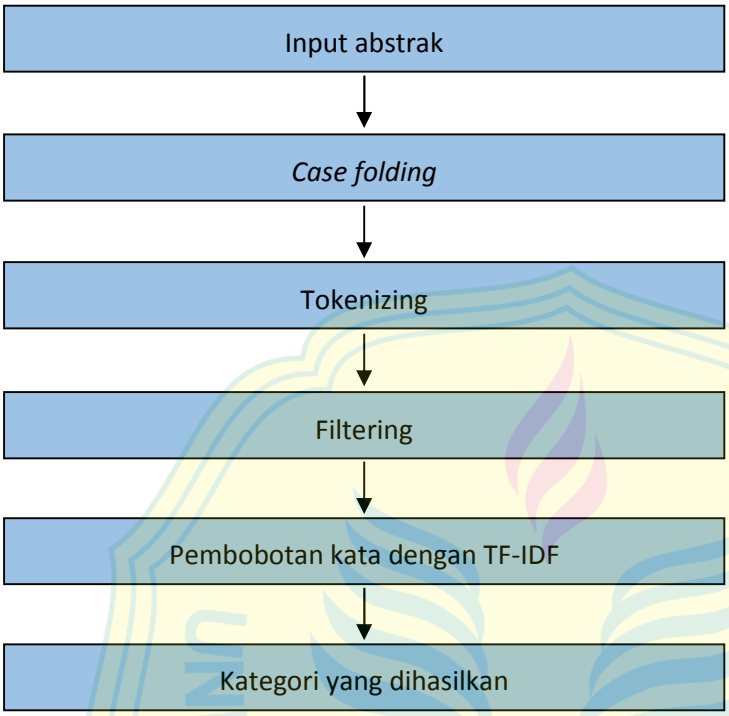
*Term Frequency Inversed Document Frequency*yang diterapkan pada aplikasi pengklasifikasi dokumen karya akhir ini, merupakan metode TF-IDF yang sudah dimodifikasi untuk menghitung nilai/bobot suatu kata terhadap kategori. Metode ini akan mengabaikan setiap kata-kata yang tergolong tidak penting. Oleh karena itu, sebelum mengimplementasi algoritma TF-IDF pada aplikasi, teks abstrak yang akan di klasifikasi harus diekstrak terlebih dahulu untuk menghasilkan token-token string dari abstrak yang di-*input*. Kemudian token-token string akan dihitung nilai TF-IDFnya terhadap kumpulan kata dalam kategori. Pada akhirnya bobot TF-IDF pada setiap kata terhadap kategori akan diakumulasi berdasarkan kategorinya masing-masing. Kategori dengan nilai tertinggi dari hasil akumulasi bobot kata TF-IDF per kategori, akan dijadikan dugaan kategori dari abstrak yang di-*input*.

Kategori yang digunakan pada aplikasi pengklasifikasi dokumen karya akhir di Jurusan Teknik Elektro Universitas Negeri Jakarta yaitu:

1. Elektro.
2. Elektronika.
3. Informatika.

4. Pendidikan.

Blok perangkat lunak aplikasi pengklasifikasi dokumen karya akhir yang mengimplementasi algoritma TF-IDF ditunjukkan oleh gambar 2.3:



Gambar 2.2Blok Perangkat Lunak

2.3. Hipotesis Penelitian

Hipotesis yang dapat diambil dalam penelitian ini adalah aplikasi diharapkan dapat membantu sistem pengklasifikasian karya akhir yang sesetiap tahunnya bertambah, dengan memanfaatkan algoritma TF-IDF dalam penerapan aplikasinya. Aplikasi nantinya akan menghasilkan kategori yang terkait dengan karya akhir.