BAB II
LANDASAN TEORI

2.1 Plagiarisme

Menurut Kamus Bersar Bahasa Indonesia (KBBI) berarti pengambilan karangan (pendapat dan sebagainya) orang lain dan menjadikannya seolah-olah karangan (pendapat dan sebagainya) sendiri, misalnya menerbitkan karya tulis orang lain atas nama dirinya sendiri.

Menurut (Soelistyo, 2011) secara etimologis plagiat berasal dari bahasa Inggris *Plagiarism* yang apabila dirunut sebenarnya berasal dari bahasa Yunani yaitu *Plagiarius* berarti penculik atau pencuri karya tulis. Kemudian di kamus (Bauer, 1980), *plagiarism* didefinisikan sebagai pengambilan gagasan dari karya orang lain kemudian menggunakan gagasan tersebut dalam karyanya sendiri tanpa memberi penghargaan terhadap penulis aslinya.

Adapun beberapa contoh yang dianggap sebagai tindakan plagiarism sebagai berikut:

1. *Copy paste* (copas) pada artikel/tulisan/posting orang lain tanpa mencantum nama pemiliknya.
2. Mengganti nama pemilik karya tulis dengan nama sendiri.
3. Menyalin bersih tulisan orang lain ke dalam karya tulis yang dibuat tanpa ada perbedaan kata.
4. Mengunakan ide orang lain baik berupa tulisan, gambar, video dan lainnya tanpa mencantumkan sumbernya.
5. Membeli karya tulis orang lain lalu menyebarkan atas nama pribadi.
6. Menulis hasil karya orang lain dengan mengganti dengan kalimat sendiri tanpa mencantumkan sumber penelitian tersebut.
7. Mengubah hasil penelitian orang lain tanpa seizin dari pemililik karya ilmiah.

Suatu dokumen dapat dikatakan plagiat apabila:

Menurut (Sudigdo, 2007) mengklasifikasikan plagiat berdasarkan proporsi atau kadar plagiatnya, yaitu:

1. Plagiat Ringan

Plagiat ringan manakala dalam sebuah karya tulis ilmiah yang dibuat oleh seseorang kurang dari 30%.

2. Plagiat Sedang

Plagiat sedang mempunyai prosentasi 30%-70% dalam sebuah karya tulis yang dibuat.

3. Plagiat Total

Plagiat total berarti lebih dari 70% isi karya tulis ilmiahnya merupakan plagiat dari karya orang lain. Plagiat ini tidak bisa ditoleril dan karya tersebut harus direvisi ataupun ditak diakui.

2.2 Information Retrieval

Information Retrieval system (sistem temu kembali informasi) adalah suatu kajian ilmu yang mempelajari tentang bagaimana cara menemukan informasi dari berbagai dokumen (*corpus*) yang tersedia dalam penyimpanan sesuai dengan yang diinginkan pengguna atau pengguna.

Ada beberapa istilah penting yang berkaitan dengan *Information Retrieval* diantaranya adalah: (Ramadhany, 2008).

1. *Corpus* (korpus)

   Corpus adalah kumpulan teks dalam kumpulan dari potongan-potongan teks bahasa dalam bentuk elektronik, dipilih sesuai dengan kriteria eksternal untuk mewakili suatu ragam bahasa sebagai sumber data penelitian linguistik.

2. Kueri (*query*)

   Kueri adalah inputan informasi yang diinginkan pengguna. Kueri bisa berupa kata kunci yang diberikan terhadap sistem untuk menperoleh hasil sesuai dengan yang diinginkan pengguna. Defenisi kueri secara formal adalah kumpulan spesifikasi yang digunakan untuk menggali data yang ada pada database.

3. Relevansi

   Relevansi adalah tingkat kesesuaian dari sebuah dokumen dengan kueri pengguna. Relevansi dihitung berdasarkan metode yang digunakan pada penelitian sistem temu kembali informasi.

4. *Rangking* (peringkat)

   Rangking adalah peringkat yang diperoleh oleh berbagai dokumen yang mengacu pada kueri pengguna.

5. *Term* (kata)

   Kata adalah kata yang memiliki arti yang terdapat pada dokumen dan kueri. Dari semua kata-kata yang merupakan kata umum akan dihilangkan sehingga yang tersisa benar-benar kata yang berhubungan dengan isi dokumen.

2.3 Penerapan Pengukuran Kemiripan Dokumen

Menurut (Kurniawati, Sekarwati, dan Wicaksana, 2012) pengukuran kemiripan dokumen dapat digunakan untuk mengukur kemiripan dokumen resmi, dokumen standar seperti *Standar Operasional Procedure* (SOP), peraturan perundangan, hasil penelitian dan lain-lain. Pengukuran dokumen dapat diterapkan di berbagai instansi seperti pemerintahan, perusahaan, akademik dan lain-lain. Menurut (Vamplew dan Dermoudy, 2005), plagiarisme dalam bidang akademik dapat dibagi menjadi dua yaitu:

1. *Content-based file comparison*

   Pendekatan *content-based file* merupakan pendekatan untuk mengukur kemiripan dokumen.

2. *Content-based comparison of source code*

   Pendekatan ini digunakan untuk mengukur kemiripan *source code* pemrograman.

2.3.1 Pengukuran Kemiripan

Pada sub-bab ini dijelaskan tentang jenis pengukuran kemiripan, pendekatan pengukuran kemiripan, penelitian-penelitian yang telah dilakukan terhadap pendekatan pengukuran kemiripan.

a. Jenis Pengukuran Kemiripan

Menurut (Novanta, 2009) berdasarkan batasan ruang lingkup pemeriksaan lokasi dokumen, pengukuran kemiripan dapat dibagi menjadi dua jenis, yaitu:

1. *Intra-Corporal Detection*

   Jenis pengukuran ini dilakukan secara *offline*, yaitu dokumen teks yang diidentifikasi plagiat (*copy documents*) diperiksa dengan dokumen teks yang dianggap asli (*source documents*) dengan dibatasi pada sebuah lokasi (*folder*) tertentu yang terdiri dari beberapa dokumen (*corpus*) yang akan dibandingkan, dimana proses pengumpulan koleksi dokumen dilakukan secara manual.

2. *Internet-based Detection*

   Jenis pengukuran ini dilakukan secara *online*, yaitu dokumen teks yang diidentifikasi plagiat (*copy documents*) diperiksa dengan dokumen teks yang dianggap asli (*source documents*) yang berada tersebar pada jaringan *World Wide Web*. Proses kerja *internet-based detection*.

b. Pendekatan Pengukuran Kemiripan

Menurut (Kurniawati et al., 2012) pendekatan pengukuran kemiripan dapat dikategorikan menjadi tiga kategori yaitu *string matching, keyword similarity* dan *fingerprint* analysis. Penjelasan pendekatan pengukuran kemiripan adalah sebagai berikut:

1. String Matching

Pendekatan *substring matching* merupakan pendekatan untuk mengidentifikasi string yang sama yang digunakan sebagai indikator kemiripan. Prinsip dari pendekatan ini adalah membandingkan semua kata dari dokumen dengan semua kata pada dokumen yang lain. Algoritma pada pendekatan ini adalah Brute Force, Edit Distance, Smith dan Karp Rabin.

2. Keyword Similarity

Prinsip dari pendekatan ini adalah mengekstraksi kata kunci dari dokumen dan kemudian dibandingkan dengan kata kunci padadokumen lain. Jika kemiripan melebihi ambang batas, dokumen dibagi menjadi bagian yang lebih kecil, yang kemudian dibandingkan secara rekursif.

3. ***Fingerprint Analysis***

Pendekatan yang paling populer untuk mengukur kemiripan adalah mengukur urutan teksyang tumpang tindih dengan cara *fingerprint*. Dokumen dibagi ke dalam urutan yang disebut dengan *chunks*, dari pembacaan dokumen dihitung pola dokumennya. Algoritma pada pendekatan ini adalah algoritma *winnowing* dan algoritma *chuncking*.

2.4 *Latent Semantic Analysis* (LSA)

Latent Semantic Analysis (LSA) adalah sebuah teori dan algoritma untuk menggali dan merepresentasikan konteks yang digunakan sebagai sebuah arti kata dengan memanfaatkan komputasi statik untuk sejumlah *corpus* yang besar. *Corpus* adalah kumpulan teks yang memiliki kesamaan subjek atau tema.

Menurut (Landauer, Folt, dan Laham, 1998) *Latent Semantic Analysis* (LSA) adalah algoritma matematika dan statistika untuk menentukan hubungan kontekstual arti kata pada bagian teks yang dibutuhkan.

Metode *Latent Semantic Analysis* (LSA) menerima masukan berupa dokumen teks pada proses awal sebelumnya. Pada proses perbandingan dengan metode LSA kata-kata yang unik pada setiap dokumen akan direpresentasikan sebagai kolom matriks. Nilai dari matriks tersebut adalah banyaknya kemunculan di sebuah kata di setiap dokumen yang akan dibandingkan. Pada LSA dilakukan beberapa proses, yaitu:

1. *Singuler Value Decomposition* adalah representasi komponen kata dan dokumen ke dalam bentuk matrik.
2. *Cosine Similarity* dikenal sebagai rumus umum untuk mengukur kemiripan kata atau dokumen.

2.4.1 *Singuler Value Decomposition*

Singuler Value Decomposition (SVD) merupakan representasi komponen kata dan dokumen ke dalam bentuk matrik, sehingga dari hasil dekomposisi SVD terdapat tiga jenis operasi perbandingan yang dapat dilakukan, diantaranya yaitu:

1. Membandingkan dua kata (*terms*)

   Koordinat dari suatu kata pada s*emantic space* direpresentasikan oleh vektor baris dari matrik U x S yang bersesuaian dengan kata tersebut. Untuk menentukan kemiripan antara dua kata menggunakan metode *cosine similarity* dengan menghitung sudut antara kedua koordinat kata. Operasi inilah yang digunakan pada tugas akhir ini.
2. Membandingkan dua dokumen (*documents*)

   Koordinat dari suatu dokumen pada s*emantic space* direpresentasikan oleh vektor baris dari matrik V x S yang bersesuaian dengan dokumen tersebut. Untuk menentukan kemiripan antara dua dokumen menggunakan metode *cosine similarity* dengan menghitung sudut antara kedua koordinat dokumen.

3. Membandingkan kata (*term*) dengan dokumen (*document*)

   Berbeda dari dua operasi di atas yang memerlukan *cosine similarity* untuk menghitung kesamaan kata atau dokumen, untuk membandingkan kata *i* dengan dokumen *j* dapat diketahui dari nilai cell (*i* ,*j*) dari matrik aproksimasikata-dokumen yang didapat dari perhitungan SVD.

Menurut (Aji, Baisal, dan Firdaus, 2011) SVD akan menguraikan sebuah matriks menjadi tiga buah matriks baru yaitu matriks vektor singular kiri, matriks nilai singular dan matriks vektor singular kanan, SVD dari sebuah matriks X berdimensi m*n adalah sebagai berikut:

$$\boldsymbol{X}_{mxn} = \boldsymbol{U}_{mxm} \cdot \boldsymbol{S}_{mxn} \cdot \boldsymbol{V}^{T}_{nxn} \quad \text{..... (2.1)}$$

Keterangan:

X : matriks berdimensi m*n

U : matriks vektor singular kiri berdimensi m*m

S : matriks nilai singular berdimensi m*n dengan nilai terurut menurun

V : matriks vektor singular kanan berdimensi n*n

V^T : matrik V transpose.

2.4.2 *Cosine Similarity*

Cosine Similarity dikenal sebagai rumus yang umumnya digunakan untuk pengukuran similaritas, dengan menentukan sudut antara vektor dokumen dengan vektor kueri dalam dimensi V ruang Euclidean, dengan V adalah ukuran *input*, (Lee, Song, dan Kim, 2010). Hasil *cosine similarity* bernilai antara 0 sampai 1. Nilai 0 menunjukkan bahwa dokumen tidak terkait atau tidak berhubungan dengan kueri. Dan jika hasil *cosine similarity* bernilai 1, berarti keterhubungan antara dokumen dengan kueri tinggi. Dengan rumus sebagai berikut:

Vektor kueri:

$$\bar{q} = q^T \, U_k \, \sum_k 1 \quad \text{..... (2.2)}$$

Keterangan:

$\bar{q}$: Vektor kueri

q^T : Transpose matrik vektor kueri

U_k : Matrik Singular kiri dalam ruang k

$\sum_k 1$: Invers Matrik Singular dalam ruang *k*

Vektor dokumen:

Pembentukan vektor dokumen ($\bar{d}$) dilakukan setelah proses SVD. vektor dokumen sama dengan setiap kolom pada matrik *VT*.

Setelah vektor dokumen dan vektor kueri dibentuk maka dihitunglah *similarity* antara kueri dan dokumen dengan cara menghitung nilai cosinus sudut yang dibentuk oleh vektor kueri dan vektor dokumen. *Similarity* vektor kueri dan vektor dokumen didefinisikan sebagai berikut (Aji et al., 2011):

$$\cos \alpha = \frac{\bar{q}.\bar{d}}{\|\bar{q}\|\|\bar{d}\|} = \frac{\sum_{i=1}^{n} \bar{q}_i \text{ x } \bar{d}_i}{\sqrt{\sum_{i=1}^{n} (\bar{q}_1)^2} \text{ x} \sqrt{\sum_{i=1}^{n} (\bar{d}_1)^2}} \text{ ..... (2.3)}$$

Keterangan:

$\bar{q}$: Vektor kueri

$\bar{d}$: Vektor dokumen

$\bar{q} . \bar{d}$: Dot produk antara vektor kueri dan dokumen

$\|\bar{q}\|$: Panjang vektor kueri

$\|\bar{d}\|$: Panjang vektor dokumen

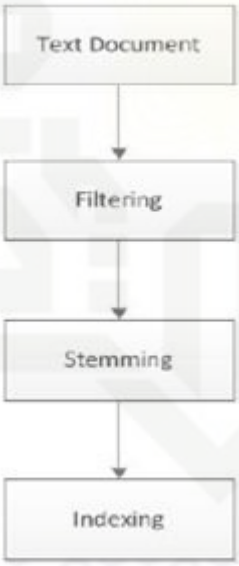
$\|\bar{q}\| \, \|\bar{d}\|$: Cross produk antara $\|\bar{q}\|$ dan $\|\bar{d}\|$

Menurut (Deerwester, Dumais, dan Harshman, 1990), ada beberapa keunggulan dari teori ini:

1. Sinonim, LSA ini mampu mengenali kata yang berbeda tapi memiliki kesamaan arti.
2. Kata yang memiliki banyak makna (Polisemi) akan mengurangi keakuratan dalam pencarian dokumen, teknik *Reduced Representation* pada LSA diharapkan akan dapat menghilangkan *noise* pada kata.
3. Keterkaitan kata, LSA memperhatikan keterkaitan kata dalam dokumen.

2.5 Metodologi *Indexing Text*

Menurut (Karyono, Utomo, Sistem, dan Balik, 2012), proses indexing terdiri dalam beberapa tahap, yaitu



Gambar 2.1 Metodologi Indexing Text

2.5.1 Filtering

Tahap filtering adalah tahap pengambilan kata-kata yang penting dari hasil tokenizing. Tahap filtering ini menggunakan daftar stoplist atau wordlist. Stoplist yaitu penyaringan (*filtering*) terhadap kata-kata yang tidak layak untuk dijadikan

sebagai pembeda atau sebagai kata kunci dalam pencarian dokumen sehingga kata-kata tersebut dapat dihilangkan dari dokumen. Sedangkan wordlist adalah daftar kata yang mungkin digunakan sebagai kata kunci dalam pencarian dokumen, dengan demikian maka tentu jumlah kata yang termasuk dalam wordlist akan lebih banyak daripada *stoplist*.

2.5.2 *Stemming*

Stemming merupakan proses dalam mengubah kata yang berimbuhan menjadi kata dasar. Imbuhan pada Bahasa Indonesia terdiri dari kombinasi :

Prefiks 1 + Prefiks 2 + Kata Dasar + Sufiks 3 + Sufiks 2 + Sufiks 1

Contohnya adalah Mem+per+main+kan yang merupakan kata dasar dari kata "main".

Pada penelitian ini, Proses *stemming* menggunakan Algoritma Nazief dan Andriani yang memiliki presentase keakuratan (presisi) yang tinggi dibandingkan dengan algoritma *stemming* lainnya.

Algoritma Nazief dan Adriani yang dibuat oleh Bobby Nazief dan Mirna Adriani ini memiliki tahap-tahap sebagai berikut:

1. Pertama cari kata yang akan di stem dalam kamus kata dasar. Jika ditemukan maka diasumsikan kata adalah kata dasar. Maka algoritma berhenti.
2. *Inflection Suffixes* ("-lah", "-kah", "-ku", "-mu", atau "-nya") dibuang. Jika berupa particles ("-lah", "-kah", "-tah" atau "-pun") maka langkah ini diulangi lagi untuk menghapus *Possesive Pronouns* ("-ku", "-mu", atau "-nya"), jika ada.
3. Hapus *Derivation Suffixes* ("-i", "-an" atau "-kan"). Jika kata ditemukan di kamus, maka algoritma berhenti. Jika tidak maka ke langkah 3a
   a. Jika "-an" telah dihapus dan huruf terakhir dari kata tersebut adalah "-k", maka "-k" juga ikut dihapus. Jika kata tersebut ditemukan dalam kamus maka algoritma berhenti. Jika tidak ditemukan maka lakukan langkah 3b.
   b. Akhiran yang dihapus ("-i", "-an" atau "-kan") dikembalikan, lanjut ke langkah 4.

4. Hapus *Derivation Prefix*. Jika pada langkah 3 ada sufiks yang dihapus maka pergi ke langkah 4a, jika tidak pergi ke langkah 4b.
   a. Periksa tabel kombinasi awalan-akhiran yang tidak diijinkan. Jika ditemukan maka algoritma berhenti, jika tidakpergi ke langkah 4b.
   b. *For* i = 1 to 3, tentukan tipe awalan kemudian hapus awalan. Jika root word belum juga ditemukan lakukan langkah 5, jika sudah maka algoritma berhenti. Catatan: jika awalan kedua sama dengan awalan pertama algoritma berhenti.
5. Jika semua langkah telah selesai tetapi tidak juga berhasil maka kata awal diasumsikan sebagai kata dasar. Proses selesai.

2.6 Pembobotan

Menurut (Sucipto, 2014) setiap kata yang telah diindeks diberikan bobot sesuai dengan skema pembobotan yang dipilih, apakah pembobotan lokal, global atau kombinasi keduanya. Jika menggunakan pembobotan lokal maka, pembobotan kata diekspresikan sebagai tf (*term frequency*). Namun, jika pembobotan global yang digunakan maka, pembobotan kata didapatkan melalui nilai idf (*inverse document frequency*). Beberapa aplikasi juga ada yang menerapkan pembobotan kombinasi keduanya yaitu, dengan mengalikan bobot lokal dan global (tf-idf).

1. *Term Frequency*

Empat cara yang dapat digunakan untuk memperoleh nilai *term frequency* (tf), yaitu:

a. *Raw term frequency*.

   Nilai tf sebuah kata diperoleh berdasarkan jumlah kemunculan kata tersebut dalam dokumen. Contohnya, jika suatu kata muncul sebanyak tiga kali dalam suatu dokumen maka, nilai tf kata tersebut adalah 3.

b. *Logarithm term frequency*.

   Hal ini untuk menghindari dominasi dokumen yang mengandung sedikit kata dalam kueri, namun mempunyai frekuensi yang tinggi. Cara ini menggunakan fungsi logaritmik matematika untuk memperoleh nilai tf.

tf = 1 + log(tf)

c. *Binary term frequency.*

Hanya memperhatikan apakah suatu kata ada atau tidak dalam dokumen. Jika ada, maka tf diberi nilai 1, jika tidak ada diberi nilai 0. Pada cara ini jumlah kemunculan kata dalam dokumen tidak berpengaruh.

d. *Augmented term frequency.*

tf = 0,5 + 0,5 x tf / *max*(tf)

Nilai tf adalah jumlah kemunculan suatu kata pada sebuah dokumen, sedangkan nilai *max*(tf) adalah jumlah kemunculan terbanyak sebuah kata pada dokumen yang sama.

2. *Inverse Document Frequency*

Inverse document frequency (idf) digunakan untuk memberikan tekanan terhadap dominasi kata yang sering muncul di berbagai dokumen. Hal ini diperlukan karena kata yang banyak muncul di berbagai dokumen, dapat dianggap sebagai kata umum (*common term*) sehingga tidak penting nilainya. Pembobotan akan memperhitungkan faktor kebalikan frekuensi dokumen yang mengandung suatu kata (*inverse document frequency*).

$$idf_j = log\left(\frac{N}{df_j}\right) \quad \text{..... (2.4)}$$

Keterangan:

N : Jumlah dokumen dalam *corpus*.

Df(t) : *Document frequency* atau jumlah dokumen dalam *corpus* yang mengandung kata t.

Menurut (Robertson, 2004) jenis formula yang akan digunakan untuk perhitungan *term frequency* (tf) yaitu tf murni (*raw* tf). Dengan demikian rumus umum untuk TF-IDF adalah penggabungan dari formula perhitungan *raw* tf dengan formula idf (rumus 2.4) dengan cara mengalikan nilai *term frequency* (tf) dengan nilai *inverse document frequency* (idf):

$$W_{ij} = tf_{ij}\ x\ idf_j \quad \text{..... (2.5)}$$

$$W_{ij} = tf_{ij}\ x\ log\left(\frac{N}{df_j}\right) \quad \text{..... (2.6)}$$

Keterangan :

W_{ij} : Bobot *term* t_j terhadap dokumen d_i

tf_{ij} : Jumlah kemunculan *term* t_j dalam dokumen *d*i

N : Jumlah semua dokumen yang ada

df_j : Jumlah dokumen yang mengandung term *t*j (minimal ada satu kata yaitu *term* t_j)