JURNAL ILMIAH FLASH
Vol. 5 No. 1, Halaman: 29 - 32
Juni 2019

SISTEM DETEKSI PLAGIARISME LINTAS BAHASA MENGGUNAKAN ALGORITMA TF-IDF

Stefania N. Lolyta[1], Rocky Y. Dillak[2], dan Folkes E. Laumal [3]
Politeknik Negeri Kupang
Jl. Adisucipto – Penfui Kupang
*E-mail: lithaloly@mail.com

Abstrak

Dalam penulisan karya ilmiah tidak sedikit orang yang karya tulisnya adalah hasil plagiarisme. Plagiarisme merupakan tindakan mengambil ide orang lain, mengambil tulisan orang lain dan mengambil isi teks secara keseluruhan dan mengakuinya sebagai miliknya sendiri. Penelitian ini dilakukan untuk mengurangi tindakan plagiarisme. Sistem ini dibuat dengan menggunakan algoritma TF-IDF melalui beberapa proses yaitu proses tralsnlasi, tokenizing, eliminasi, stemming dan menghitung cosines similarity untuk mendapatkan hasil presentasi kesamaan dari dokumen yang di uji. Berdasarkan hasil pengujian yang telah dilakukan menunjukan bahwa sistem dapat mendeteksi kesamaam dokumen yang diuji.

Kata kunci: deteksi plagiarisme, lintas Bahasa, algoritma, TD-IDF

PENDAHULUAN

Karya tulis adalah sebuah hasil karangan dalam bentuk tulisan yang merupakan hasil dari sebuah penelitan, pengamatan, tinjauan dalam bidang tertentu yang disusun secara sistematis. Dalam penulisan karya ilmiah tidak sedikit orang yang karya tulisnya adalah hasil plagiarisme. Plagiarisme merupakan tindakan mengambil ide orang lain, mengambil tulisan orang lain dan mengambil isi teks secara keseluruhan dan mengakuinya sebagai miliknya sendiri.

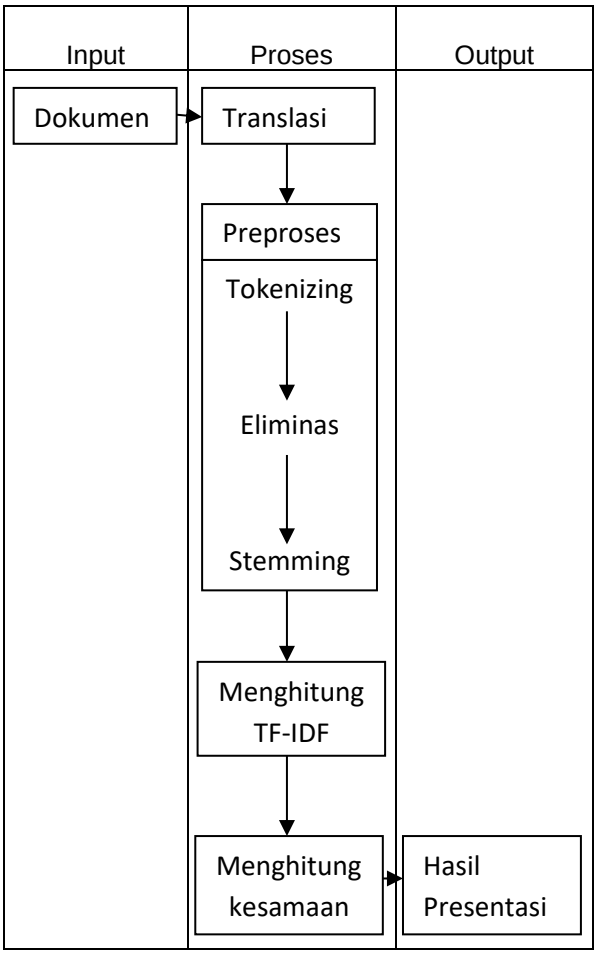
Pada penelitian sebelumnya yang dilakukan oleh Kadja, Jen, Ledy, dkk (2016, h. 2-3) dengan judul penelitian "*Deteksi Dini Plagiarisme pada Konten Teks Digital Tugas Akhir Mahasiswa Jurusan Teknik Elektro Politeknik Negeri Kupang Menggunakan Algoritma N-gram dan Winnowing* ", pendeteksian plagiarisme ditujukan untuk Tugas Akhir mahasiswa dengan penerapan algoritma N-gram dan Winnowing. Proses pendeteksian ini dilakukan dengan membandingkan dua buah file yang berbeda dan hanya berformat txt (.txt). Kesimpulannya adalah pendeteksian ini masih kurang efektif disebabkan oleh semakin banyaknya isi sebuah file yang dideteksi, waktu prosesnya akan semakin lama (running time) dan waktu pendeteksian file paling lambat 3.600 jika melewati batas maka akan terjadi error.

Algoritma yang akan digunakan dalam sistem ini adalah algoritma TF-IDF (*Term frequency-inverse document frequency*). Algoritma TF-IDF (*Term frequency-inverse document frequency*) yaitu algoritma yang digunakan untuk menghitung bobot setiap kata yang paling umum digunakan pada information retrieval. Diharapkan dengan algoritma ini dapat menambah tingkat keefektifan hasil kemiripan dan waktu yang dibutuhkan untuk melakukan proses pendeteksian. Kemudian input yang dimasukan oleh user tidak hanya berformat txt (.txt) tetapi berformat pdf (.pdf).

Berdasarkan permasalahan diatas maka penulis memberikan judul "**Sistem deteksi plagiarisme lintas bahasa menggunakan algoritma TF-IDF**". Dengan harapan sistem ini dapat memberikan solusi untuk mengurangi terjadinya tindakan plagiarisme.

METODE PENELITIAN

Metode penelitian yang digunakan pada penelitian ini dapat dilihat pada gambar arsitektur sistem berikut ini :



Gambar 1 Arsitektur Sistem

Berdasarkan gambar 1, dapat dijelaskan proses pendeteksian plagiarisme sebagai berikut :

1. Translasi

Translasi yaitu proses menerjemahkan isi dokumen dari bahasa inggris ke bahasa Indonesia jika dokumen yang diuji merupakan dokumen bahasa inggris. Proses penerjemahan ini menggunakan Yandex API sebagai media penerjemah. Berikut merupakan potongan kode program *translasi* :

```
$AW6_text=htmlentities(ucfirst(
trim(!empty($_POST['aw6_transla
te_from']) ?
$_POST['aw6_translate_from'] :
$AW6_text)));

$AW6_from$=htmlentities(!empty(
$_POST['aw6_langpairFROM']) ?
$_POST['aw6_langpairFROM'] :
'');
$AW6_to=htmlentities(!empty($_P
OST['aw6_langpairTO']) ?
$_POST['aw6_langpairTO'] : '');
if (!empty($_POST)):

$AW6_translation=translate($AW6
_text, $AW6_from, $AW6_to);
endif;
   }
```

2. Tokenizing

Tokenizing yaitu proses mengubah semua huruf besar menjadi huruf kecil (a sampai z) dan menghilangkan karakter-karakter huruf. Berikut merupakan potongan kode program dari *tokenizing* :

```
//tokenizing => ubah ke huruf
kecil
$teks =
strtolower(trim($teks));
//hilangkan tanda baca
$teks = str_replace("'", " ",
$teks);
$teks = str_replace("-", " ",
$teks);
$teks = str_replace(")", " ",
$teks);
$teks = str_replace("(", " ",
$teks);
```

3. Eliminasi

Eliminasi stopwords yaitu proses membuang kata yang tidak relevan atau kata umum yang biasanya muncul dalam jumlah besar. Berikut merupakan potongan kode program dari *eliminasi stopwords* :

```
foreach ($astoplist as $i =>
$value) {
$teks =
preg_replace("/\b{$value}\b/i",
"", $teks);
}
$teks = trim($teks);
return $teks;
} //end function
```

4. Stemming

Stemming yaitu proses mengubah semua kata ke bentuk kata dasar. Tujuan dari proses ini adalah mencari root kata dari tiap kata hasil *eliminasi stopwords*. Berikut merupakan potongan kode program dari proses *stemming*:

```
for($i=0;$i<count($d);$i++){
  $x="";
  $h=array();
  $h=explode(" ",$d[$i]);
      for($j=0;$j<count($h);$j++){

      $h[$j]=stemming($h[$j]);
      $x .=$h[$j]." ";
```

```
    }
    $dok[$i]=$x;
```

5. Menghitung TF-IDF

TF-IDF (*Term Frequency - Inverse Dokumen Frequency*) yaitu proses meghitung bobot setiap kata yang paling umum digunakan. Berikut merupakan potongan program dan hasil perhitungan dari *Term Frequency - Inverse Dokumen Frequency* :

```
public function cari_idf($dok)
{
    $n=count($dok);
    for($i=0;$i<count($this-
>df);$i++)
    {
          $this->idf[$this-
>semua_kata[$i]]=round((log($n/
($this->df[$this-
>semua_kata[$i]]))),4)+1;
    }
}
//tf-idf
public function cari_w($d)
{
    for($i=0;$i<count($this-
>semua_kata);$i++)
    {

    for($j=0;$j<count($d);$j++)
          {
          $this->bobot_w[$this-
>semua_kata[$i]][$j]=$this-
>idf[$this-
>semua_kata[$i]]*$this-
>tfd[$this-
>semua_kata[$i]][$j];
          }
          }
```

6. Menghitung Cosinus Similarity

Perhitungan kesamaan dokumen ini menggunakan *cosine similarity* dari hasil teks yang telah diproses pada perhitungan TF-IDF. Tujuan dari proses ini yaitu untuk mengetahui tingkat kemiripan dari dokumen yang diuji. Berikut adalah potongan kode program dan hasil perhitungan dari *cosine similarity* :

```
for($j=0;$j<count($dok)-1;$j++)
    {
          $this-
>magnitude_w[$j]=$mag[$q]*$m
ag[$j];
          $this-
>cosinus[$j]=$this-
>sum_inner_w[$j]/$this-
>magnitude_w[$j];
          $this-
>cosinus[$j]=round((100 *
$this->cosinus[$j]),2);
    }
```

HASIL DAN PEMBAHASAN

Pengujian sistem ini bertujuan untuk mengetahui apakah sistem yang sebelumnya didesain dan dibuat dapat berjalan dengan lancar atau tidak. Pengujian ini dilakukan dengan membandingkan satu dokumen dengan salinan dari dokumen itu sendiri. Setiap dokumen yang disalin telah ditentukan tingkat kesamaanya masing-masing. Tingkat kesamaan yang ditentukan tersebut adalah tingkat kesamaan antara dokumen salinan dengan dokumen asli, untuk mendapatkan tingkat kesamaan tersebut, setiap salinan telah mengalami pengurangan kata sesuai dengan tingkat kesamaannya. Berikut merupakan tabel hasil pengujian antara dokumen asli dengan dokumen salinan.

Tabel 1 Tabel pengujian sistem

No. Dok	Perhitungan Plagiarisme		Selisih
	Manual	Sistem	
1.	100%	100%	0%
2.	91,73%	91,73	0%
3.	83,68%	83,68%	0%
4.	72,57%	72,57%	0%
5.	43,52%	43,52%	0%
6.	31,12%	31,12%	0%
7	0,71%	0,71%	0%

Berdasarkan Tabel 1 dapat dilihat bahwa perhitungan manual dengan perhitungan sistem memiliki selisih yang tidak jauh berbeda. Misalnya pada dokumen 3 perhitungan secara manual 81,04% sedangkan perhitungan melalui sistem 83,68% maka selisih dari kedua perhitungan tersebut adalah 2,64%.

Berdasarkan pengujian yang telah dilakukan maka dapat ditentukan ukuran nilai *similarity* antar dokumen-dokumen yang diuji menjadi 5 jenis penilaian presentase *similarity* sebagai berikut :

1. Hasil uji 0% berarti kedua dokumen tersebut benar-benar berbeda baik dari segi isi dan kalimat secara keseluruhan.
2. Hasil uji kurang dari 15% berarti kedua dokumen tersebut hanya mempunyai sedikit kesamaan.

3. Hasil uji 15% sampai 50% berarti dapat dikatakan bahwa dokumen tersebut mendekati plagiarisme.
4. Hasil uji lebih dari 50% berarti menandakan dokumen tersebut termasuk paliat tingkat sedang.

KESIMPULAN

Berdasarkan hasil pengujian sistem yang telah dilakukan, dapat disimpulkan bahwa :

1. Sistem membandingkan satu dokumen asli dengan beberapa dokumen salinan dari dokumen asli dan menampilkan hasil berupa presentasi kesamaan.
2. Semakin banyak dokumen pembanding yang disimpan pada server maka waktu yang dibutuhkan untuk memproses dokumen yang ada semakin lama.

DAFTAR PUSTAKA

[1]. Andayani, Sri; Riansyah, *Ady. Implementasi Algoritma TF-IDF Pada Pengukuran Kesamaan Dokumen*. http://ojs.ukmc.ac.id/index.php/JUTSI/article/view/218. Diakses tanggal 15 Maret 2018.

[2]. Brianorman, Yulrio. 2016. *Aplikasi Pendeteksi Plagiat Terhadap Karya Tulis Berbasis WEB Menggunakan Natural Language Processing dan Algoritma Knuth-Morris-Pratt*. http://jurnal.untan.ac.id/index.php/jcskommipa/article/view/13332. Diakses pada 15 Maret 2018.

[3]. Dillak, R., Laumal, F., & Kadja, L. (2016). SISTEM DETEKSI DINI PLAGIARISME TUGAS AKHIR MAHASISWA MENGGUNAKAN ALGORITMA NGRAMS DAN WINNOWING. *Jurnal Ilmiah Flash, 2*(1),12-18. doi:10.32511/jiflash.v2i1.19

[4]. Khairunnisa, Nova, dkk. 2012. *Aplikasi pendeteksi plagiat dengan menggunakan metode Latent semantic Analisys (studi kasus : Laporan TA PCR).* https://jurnal.pcr.ac.id/index.php/jakt/article/view/550. Diakses tanggal 10 Mei 2018.

[5]. Maarif, Azis, Abdul. *Penerapan algoritma TF-IDF untuk Pencarian Karya Ilmiah.* http://mahasiswa.dinus.ac.id/docs/skripsi/jurnal/15309.pdf. Diakses tanggal 15 Maret 2018.

[6]. Salmuasih, dkk. 2013. *Perancangan sistem deteksi plagiat pada dokumen teks dengan konsep similarity menggunakan algoritma Tabin Karpp.* https://edoc.site/perancangan-sistem-deteksi-plagiat-pada-dokumen-teks-pdf-free.html. Diakses tanggal 10 Mei 2018.

[7]. Supraptono, Eko. 2017. *Penerapan Algoritma Cosine Similarity dan Pembobotan TF-IDF pada Sistem Klasifikasi Dokumen Skripsi.* https://journal.unnes.ac.id/artikel_nju/jte/10955. Diakses tanggal 15 Maret 2018.

[8]. Suryana, Emis. 2016. *Self efficacy dan Plagiarisme di perguruan tinggi.* http://jurnal.radenfatah.ac.id/index.php/Tadrib/article/download/1169/988. Diakses tanggal 14 april 2018.

[9]. Tudesman, dkk. 2010. *Sistem deteksi plagiarisme dokumen bahasa Indonesia menggunakan metode vector space model.* http://eprints.mdp.ac.id/998/1/21tudesmanJurnal.pdf. Diakses tanggal 10 Mei 2018.

[10]. https://play.google.com/store/apps/details?id=ru.yandex.translate&hl=en. Diakses tanggal 08 Mei 2019.

[11]. https://yandex-translate-free-offline-dictionary-and-translator-acr-ios.soft112.com/. Diakses tanggal 08 Mei 2019.