Deteksi Plagiat Dokumen Tugas Daring Laporan Praktikum Mata Kuliah Desain Web Menggunakan Metode Naive Bayes

Dwi Susanto[1], Achmad Basuki[2] dan Prada Duanda[3]
Prodi Teknologi Multimedia Broadcasting, Departemen Teknologi Multimedia Kreatif, Politeknik Elektronika Negeri Surabaya

dwi@pens.ac.id[1], basuki@pens.ac.id[2], pradaduanda@gmail.com[3]

ABSTRAK

Website Dosenjaga merupakan sebuah Learning Management System(LMS) untuk program Pendidikan Jarak Jauh(PJJ) yang diterapkan di Politeknik Elektronika Negeri Surabaya (PENS). Pada website ini terdapat berbagai fasilitas yang digunakan untuk memudahkan mahasiswa dan dosen PJJ dalam proses pembelajaran. Salah satu fasilitasnya yakni pengumpulan tugas dalam jaringan(daring). Fasilitas ini memberikan akses bagi mahasiswa untuk mengumpulkan tugas dalam bentuk dokumen dengan mengunggahnya ke Dosenjaga. Pengumpulan tugas dalam bentuk dokumen ini membuka peluang terjadinya penjiplakan atau plagiat yang dilakukan oleh mahasiswa. Sementara itu, proses pengecekan dokumen unggahan mahasiswa masih dilakukan secara manual oleh dosen. Berdasarkan permasalahan tersebut, maka dibuatlah sistem pendeteksi plagiat yang diimplementasikan pada Dosenjaga dalam bentuk fitur. Naive Bayes digunakan untuk mendeteksi kesamaan antar dokumen tugas mahasiswa. Hasil yang didapatkan dari sistem pendeteksi plagiat ini adalah persentase kesamaan antara dokumen yang dibandingkan.

Kata Kunci— elearning , naive bayes ,plagiarisme, tugas daring .

ABSTRACT

Website Dosenjaga is a Learning Management System (LMS) for Distance Learning (ODL) in Electronic Engineering Polytechnic Institute of Surabaya (PENS). There are various facilities that are used to facilitate students and teachers in the learning process on this website. One of them is the online assignment. This facility provides access for students to upload assignments to Dosenjaga. In this feature there is a possibility of plagiarism committed by students. Meanwhile, the document checking process is still done manually by the lecturer. Based on these problems, plagiarism detection system is created. Naive Bayes is used to detect a similarity between the assignment document. Results obtained from this plagiarism detection system is the percentage of similarity between the compared documents.

Keywords : elearning, naive bayes, plagiarism, online assignment

I. PENDAHULUAN

Sebuah salinan elektronik yang dapat Berdasarkan Undang-Undang Perguruan Tinggi nomer 12 tahun 2012, pasal 31 tentang Pendidikan Jarak Jauh (PJJ) menjelaskan bahwa PJJ merupakan proses belajar mengajar yang dilakukan secara jarak jauh melalui penggunaan berbagai media komunikasi. PJJ adalah suatu sistem pendidikan yang memiliki karakteristik terbuka, belajar mandiri, dan belajar tuntas dengan memanfaatkan TIK dan/atau menggunakan teknologi lainnya. Untuk mendukung terlaksananya program PJJ, maka LMS dengan nama Dosenjaga dibuat. LMS ini digunakan untuk mengelola pembelajaran PJJ secara online. Salah satu fitur yang ada pada Dosenjaga adalah upload tugas oleh mahasiswa. Selanjutnya dosen akan memeriksa hasil tugas mahasiswa secara manual. Upload tugas berupa dokumen ini membuka peluang munculnya plagiarisme. Saat ini pengecekan terkait plagiarisme pada Dosenjaga masih dilakukan secara manual.

Berdasarkan permasalahan tersebut, maka dibuatlah fasilitas tambahan yang dapat digunakan untuk mengecek plagiarisme dari dokumen tugas yang diupload oleh mahasiswa. Pengecekan plagiarisme dilakukan dengan menggunakan metode Naive Bayes. Tugas mahasiswa yang dijadikan sebagai sample adalah tugas laporan praktikum mata kuliah Desain dan Pemrograman Web. Proses yang dilakukan adalah dengan memilih salah satu dokumen tugas mahasiswa sebagai rujukan. Selanjutnya semua dokumen tugas mahasiswa yang lain dibandingkan dengan dokumen rujukan. Hasil prosentase yang diperoleh menentukan apakah dokumen tersebut merupakan plagiat atau

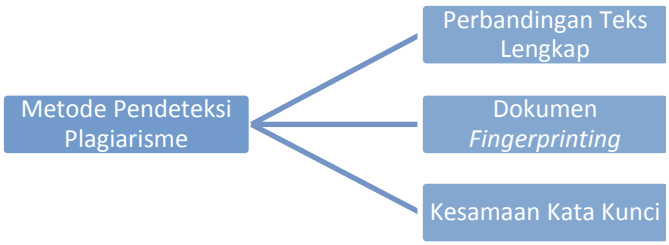
bukan.

1.1. Plagiarisme

Berdasarkan Kamus Besar Bahasa Indonesia(KKBI)[1] plagiarisme adalah perbuatan mengambil karangan atau pendapat orang lain dan menjadikannya seolah-olah karangan atau pendapat itu milik sendiri. Menurut Paul dan Jamal[2], plagiarisme adalah mengambil sebagian tulisan karya orang lain tanpa melalui proses citasi yang benar, sehingga hal ini dianggap sebagai sebuah pencurian dengan menganggap karya orang lain sebagai miliknya.

Sastroasmoro[4] membagi plagiarisme menjadi beberapa jenis yakni plagiarisme berdasarkan aspek yang dicuri, berdasarkan sengaja atau tidak, berdasarkan proporsi kata atau kalimat yang dibajak, dan juga plagiarisme berpola.

Metode pendeteksi plagiarisme dibagi menjadi tiga macam cara yaitu metode perbandingan teks lengkap, metode dokumen *fingerprinting*, dan metode kesamaan kata kunci.



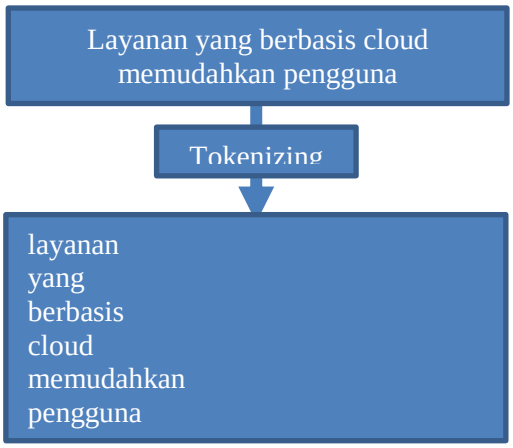
Gambar 1. Klasifikasi Metode Pendeteksi Plagiarisme

Pada Gambar 1. menunjukkan klasifikasi metode yang digunakan dalam mendeteksi plagiarisme. Pada metode Perbandingan Teks Lengkap, pencarian kesamaan antar dokumen hanya dapat dilakukan pada dokumen – dokumen yang berada pada penyimpanan lokal. Pencarian kesamaan menggunakan metode ini dapat diterapkan untuk jumlah dokumen yang banyak. Akan tetapi, akan dibutuhkan waktu yang lama dalam proses pencariannya yang seimbang dengan hasil pencarian kesamaan yang cukup efektif. Algoritma yang menggunakan metode ini adalah algoritma *brute force*, algoritma *edit distance*, algoritma *boyer moore* dan algoritma *lavenshtein distance.*

Metode Dokumen Fingerprinting melakukan pencarian kesamaan antar dokumen dapat dilakukan pada seluruh atau sebagian teks dalam dokumen dengan menggunakan teknik *hashing*.Teknik *hashing* sendiri merupakan sebuah fungsi yang bekerja dengan mengkonversi setiap *string* atau kata dalam dokumen menjadi kedalam bentuk bilangan. Sedangkan metode Kesamaan Kata Kunci, pencarian kesamaan antar dokumen dilakukan dengan mencari kata kunci dari dokumen yang dicek untuk kemudian dibandingkan dengan dokumen lain. Pendekatan yang digunakan pada metode ini adalah teknik dot.

1.2.Tahap *Preprocessing*

Informasi yang akan digunakan sebagai sumber seringkali tidak tersusun dengan rapi. Susunan informasi yang tidak rapi akan mempengaruhi kecepatan sistem dalam melakukan proses komputasi. Agar informasi dapat tersusun secara rapi, dibutuhkan sebuah proses awal agar data dapat terstruktur dengan baik. Salah satu yang dilakukan dalam langkah ini adalah proses *tokenizing*.



Gambar 2. Ilustrasi proses *tokenizing* dengan memotong perkata pada tiap dokumen

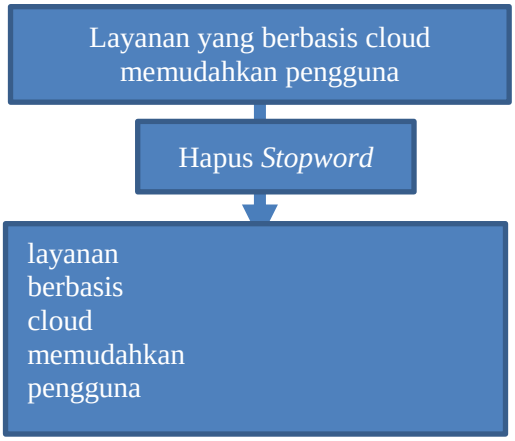
Tokenizing merupakan proses pemotongan kalimat menjadi per kata tanpa memperhatikan struktur kalimatnya. Pada Gambar 1 proses *tokenizing* dilakukan dengan memotong setiap kata dan menyusunnya sesuai dengan urutan pada kalimat aslinya. Proses ini berguna untuk dilanjutkan pada proses selanjutnya yakni menghilangkan kata-kata tertentu, misalnya kata sandang "yang".

1.3. Ekstrak

Proses ini digunakan untuk mengambil isi dari dokumen pdf untuk kemudian dipecah menjadi kalimat – kalimat dan dipecah kembali menjadi kata. Pada tahapan ini juga dilakukan penghapusan karakter - karakter tertentu seperti tanda baca atau tanda dalam pemrograman dan mengubah semua kata ke bentuk huruf kecil (*lower case*).[10]

1.4. Penghapusan Stopword

Stopword merupakan term yang tidak berhubungan dengan subyek utama dari database. Meskipun kata-kata tersebut sering muncul di dalam dokumen. Beberapa contoh stopword misalnya ada, adalah, adapun, agak, yang, dan lain-lain.



Gambar 3. Ilustrasi penghapusan kata-kata yang termasuk *stopword*

Gambar 2 menunjukkan contoh ilustasi penghapusan kata-kata yang termasuk dalam *stopword.* Pada contoh Gambar 2. kata "yang" dihapus karena termasuk dalam *stopword.*

1.5. *Stemming*

Proses berikutnya yang dilakukan adalah *stemming*, dimana dalam proses ini, kata-kata akan dirubah menjadi akar kata nya. Algoritma yang digunakan untuk proses *stemming* pada penelitian ini adalah algoritma Nazief dan Adriani. Algoritma ini memiliki beberapa tahapan untuk mengembalikan sebuah kata ke akar atau asal kata nya. Tahapan yang pertama adalah dengan mengecek kata pada koleksi kata yang dimiliki, apabila ditemukan, berarti kata tersebut merupakan akar kata. Tahapan berikutnya dengan mengidentifikasi imbuhan baik awalan dan akhiran. Apabila dikenali sebagai imbuhan dan akhiran, maka akan dilakukan penghapusan[6].

1.6. Algoritma Term Frequency (TF)

Algoritma Term Frequency merupakan suatu cara untuk memberikan bobot hubungan suatu kata (Term) terhadap dokumen.Term Frequency (TF) merupakan frekuensi kemunculan sebuah kata di dalam sebuah dokumen tertentu. Frekuensi kemunculan kata di dalam dokumen yang diberikan menunjukan seberapa penting kata tersebut di dalam dokumen. Sehingga bobot hubungan antara sebuah kata dan sebuah dokumen akan tinggi apabila frekuensi kata tinggi di dalam dan frekuensi keseluruhan dokumen yang mengandung kata tersebut akan rendah pada kumpulan dokumen[5].

Pemberian bobot atau *weighting* dilakukan pada tiap *term* atau kata dari dokumen. Pembobotan dapat dilakukan dalam bentuk pembobotan lokal, global atau kombinasi keduanya. Bobot lokal seperti *term frequency* (tf) dan pembobotan global seperti *global inverse document frequency* (idf). Kombinasi keduanya ditulis dengan tf.idf. Proses pemberian bobot dilakukan setelah *query* melewati proses *tokenizing*, penghapusan *stopword*, dan *stemming*. Setiap kata dari dokumen akan dihitung jumlah kemunculannya dalam dokumen yang sama dan dalam keseluruhan dokumen.

Pada algoritma TF/IDF digunakan rumus untuk menghitung bobot (W) masing-masing dokumen terhadap kata kunci dengan menggunakan rumus 1.

$$W_{dt} = tf_{dt} * IDF_t \quad (1)$$

Dimana d merupakan dokumen ke d dan t merupakan kata ke t dari kata kunci. W_{dt} merupakan nilai bobot kata ke t dari dokumen ke d. Tf merupakan frekuensi kemunculan atau banyaknya kata dalam sebuah dokumen. IDF merupakan *Inverse Document Frequncey* yang diperoleh dengan membagi jumlah total dokumen dengan jumlah dokumen yang memiliki kata yang dicari dan kemudian melakukan log.

Setelah bobot (W) masing-masing dokumen diketahui, dilakukan proses *sorting*/pengurutan dimana semakin besar nilai W, semakin besar tingkat similaritas dokumen tersebut terhadap kata kunci, demikian sebaliknya. Perhitungan pertama kali dilakukan pada nilai TF yakni nilai frekuensi kemunculan kata.

1.7. Naive Bayes

Algoritma klasifikasi yang digunakan adalah Naive Bayes. Naive Bayes (NB) digunakan sebagai aturan dasar dalam klasifikasi kelas pertanyaan, karena kecepatannya, mudah diimplementasikan dan walaupun ketidak bergantungan biasanya merupakan asumsi yang kurang, pada aplikasinya NB sering mampu bersaing dengan teknik klasifikasi yang lebih canggih. NB yang cukup banyak digunakan untuk klasifikasi sebenarnya berdasarkan teori probabilitas sederhana yang dikenal dengan aturan Bayes.

$$P(C_k|X) = P(C_k)\frac{P(X|C_k)}{P(X)} \quad (2)$$

Saat $P(C_k|X)$ diketahui sebagai permasalahan klasifikasi, klasifikasi dapat dilakukan secara optimal, misalnya jumlah kesalahan klasifikasi yang diharapkan dapat dikurangi dengan menambahkan sebuah dokumen dengan fitur vektor X ke kelas C_k dimana $P(C_k|X)$ tertinggi. Seringkali $P(C_k|X)$ tidak diketahui dan harus diestimasi dari data, yang sulit apabila dilakukan secara langsung. Cara yang umum untuk mengatasi kesulitan ini adalah dengan mengasumsikan distribusi X berdasarkan pada C_k dan bisa diuraikan untuk semua C_k dan dapat ditulis seperti pada formula 3.

$$P(X|C_k) = \prod_{j=1}^{d} P(X_j|C_k) \quad (3)$$

Apabila kita mengasumsikan ketidakbergantungan kemunculan (itulah mengapa metode ini diberi nama Naive), yang merupakan kemunculan nilai tertentu dari X_j, secara statistik tidak tergantung kemunculan dari X_j yang lain. Misalnya sebuah dokumen dengan tipe C_k , maka dapat dirumuskan seperti rumus 4

$$P(C_k|X) = P(C_k)\frac{\prod_{j=1}^{d} P(X_j|C_k)}{P(X)} \quad (4)$$

Apabila semua nilai dari sisi kanan diestimasikan maka kita akan memiliki estimasi untuk $P(C_k|X)$.

$$P(C_k|X) = \frac{P(C_k)\prod_{j=1}^{d} P(X_j|C_k)}{P(X)} \quad (5)$$

Apabila tujuan dari klasifikasi adalah untuk menimialisir jumlah kesalahan, maka sebuah dokumen dengan fitur vektor X dapat dimasukkan pada tiap C_k seperti $P(C_k|X)$ yang tertinggi[6].

1.8. Penelitian Terkait

Deteksi terhadap plagiarisme telah banyak dilakukan dengan menggunakan metode yang beragam. Untuk pengecekan plagiarisme pada dokumen berbahasa Indonesia berbeda dengan pengecekan dokumen berbahasa asing. Misalnya yang dilakukan oleh Isa dan Abidin[3]. Pada penelitiannya mereka menggunakan Vector Space Model(VSM) untuk mengecek kesamaan paragraf dengan menggunakan data set dari beberapa kampus yang ada di Indonesia. Proses yang dilakukan adalah dengan memecah dokumen menjadi paragraf. Hasil yang diperoleh adalah VSM dapat mendeteksi plagiarisme dengan menemukan tingkat kesamaan dokumen pada pasangan paragraf dengan tingkat similaritas yang tinggi.

Sedangkan Amalia, Budi, dan Antonius[8] melakukan klasifikasi dan pencarian jurnal dengan menggunakan Naive Bayes dan Vector Space Model. Data set yang digunakan berupa jurnal berbahasa Inggris yang diambil dari Proquest dan dikategorikan dalam lima kategori. Pada penelitian ini Naive Bayes digunakan untuk klasifikasi label kategori dari tiap jurnal. Hasil yang diperoleh mampu menghasilkan nilai akurasi sebesar 64%.

Pada penelitian yang dilakukan oleh Danang, Irawan, dan Rukmi[9] Bayesian dikombinasikan dengan *Latent Semantic Analysis* untuk mendeteksi kemiripan antar dokumen teks. Bayesian digunakan untuk menjaga dan memperhatikan pola *term* dalam mendeteksi kemiripan antar dokumen. Hasil yang diperoleh bisa lebih baik karena deteksi kemiripan tidak hanya mengacu pada frekuensi *term*, akan tetapi juga makna yang tergantung dalam dokumen yang digunakan sebagai pembanding.

II. METODE PENELITIAN

2.1 Data Set

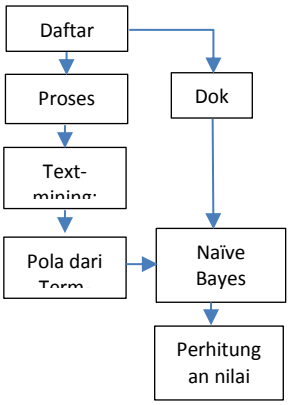
Data set yang digunakan dalam penelitian ini adalah berupa kumpulan dokumen pdf yang berisi laporan praktikum dari mata kuliah Desain dan Pemrograman Web. Jumlah dokumen yang digunakan sebanyak 120 buah dokumen. Isi dari tiap dokumen memiliki struktur yang sama yaitu berisi dasar teori dari praktikum, analisa hasil praktikum dan *screenshot* hasil praktikum.

2.2. Algoritma

Proses deteksi plagiarisme terhadap hasil tugas yang diunggah oleh mahasiswa ke LMS Dosenjaga melewati beberapa tahapan. Tahapan yang pertama adalah menentukan dokumen yang dijadikan sebagai sebuah referensi dari sekumpulan dokumen pada tugas yang sama. Penentuan referensi dapat dilakukan dengan dua cara. Cara pertama adalah sistem merekomendasikan sebuah dokumen untuk dijadikan referensi dengan berdasarkan waktu pengumpulan yang paling awal. Cara kedua yakni dilakukan secara manual dengan memilih salah satu dokumen tugas untuk dijadikan referensi.

Gambar 3 menunjukkan proses yang dilewati dalam mendeteksi plagiarisme. Dokumen dalam format pdf dirubah menjadi bentuk teks agar proses ekstraksi dokumen dapat dilakukan. Setelah melewati tahap preprocessing, maka akan dilakukan pembobotan dengan menggunakan TF/IDF. Selanjutnya Naive Bayes digunakan mengecek kemiripan antar dokumen. Hasil yang diperoleh berupa nilai prosentase seberapa besar tingkat plagiat dari dokumen yang diperiksa.

Penentuan sebuah dokumen merupakan plagiat atau tidak mengacu pada *Role Of Librarian In Quality Sustenance In Research Publications Through Plagiarism Checker Prevention, Detection And Response*[7] yakni apabila prosentase hasilnya sebesar 0% maka tidak termasuk plagiat. Prosentase dengan nilai 1%-5% termasuk kategori plagiat ringan. Sedangkan 6%-16% termasuk plagiat sedang. Dokumen dikatakan plagiat total apabila hasil pengecekan menghasilkan nilai prosentase sebesar 17%-100%.



Gambar 4. Rancangan sistem dari deteksi plagiarisme yang dibuat.

2.3. Rancangan Sistem

Tahap ini merupakan tahap perencanaan tampilan sistem pendeteksi plagiat yang meliputi pembuatan desain wireframe dan pembuatan prototype tampilan yang akan digunakan sebagai panduan dalam membuat sistem pembuat deteksi plagiat.

A. Desain Wirefrrame

Pada tahap ini, dilakukan pembuatan rancangan tampilan tata letak, yakni wireframe. Wireframe ini akan digunakan sebagai panduan untuk membuat tampilan untuk sistem pendeteksi plagiat. Pembuatan wireframe ini didasarkan pada tampilan halaman koreksi tugas pada *user* dosen di Dosenjaga, yakni tata letak elemen – elemen website tersebut seperti letak tombol, tabel dan lain sebagainya.

Gambar 5. Rancangan tampilan awal

Berdasarkan Gambar 4, tahap pertama dalam proses pengecekan plagiat adalah pemilihan dokumen referensi dari tampilan daftar dokumen seperti pada Gambar 3.4. Gambar 5 merupakan gambar rancangan tampilan tata letak tombol yang akan dibuat pada sistem pendeteksi plagiat pada halaman pengecekan tugas. Pada tampilan tersebut ditambahkan tombol "jadikan referensi" dan "cek plagiat" serta radio button di masing – masing dokumen yang ada di dalam tabel dari tampilan semestinya pada Dosenjaga.

Tombol "jadikan referensi" digunakan untuk menyimpan dokumen sebagai referensi setelah memilih dokumen dengan memberi tanda pada radio button di dokumen yag diinginkan untuk menjadi dokumen referensi. Setelah tombol "referensi" tersebut ditekan, tombol tersebut akan digantikan dengan tombol "pilih semua" dan tampilan list daftar dokumen akan berubah, dimana dokumen referensi akan ditandai dan tidak dapat dihilangkan tandanya. Sementara itu pada daftar dokumen dalam tabel, akan diimbuhi dengan checkbox seperti pada Gambar 6.

Gambar 6. Rancangan tampilan setelah pemilihan dokumen yang akan dicek

setelah pengecekan selesai dilakukan, akan tampil hasil pengecekan berupa keterangan persentase kesamaan dengan dokumen referensi pada masing – masing kolom catatan untuk tiap dokumen pada tabel. Kemudian dokumen – dokumen lain yang berada di urutan pengumpul kedua setelah dokumen referensi hingga yang terakhir, akan langsung menjadi dokumen yang dicek dan dibandingkan dengan dokumen referensi.

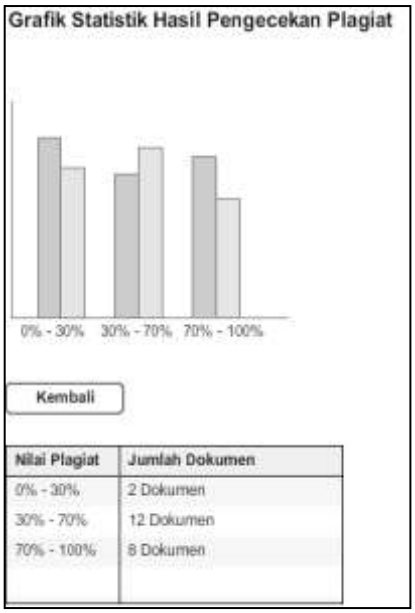
Gambar 7. Rancangan hasil pengecekan dengan satu referensi

Pada Gambar 7. setelah tombol "cek plagiat" dipilih, tombol tersebut akan digantikan dengan tombol "lihat grafik". Tombol tersebut berfungsi untuk menampilkan grafik batang yang dihasilkan dari persentase hasil pengecekan pada keseluruhan dokumen yang telah dicek seperti pada Gambar 8.

Tampilan grafik batang dari hasil pengecekan menampilkan hasil pengelompokkan dokumen yang telah dicek ke dalam 4 kriteria didasarkan dari penggolongan dokumen berdasarkan plagiat yang ditemukan dalam dokumen pada "*ROLE OF LIBRARIAN IN QUALITY SUSTENANCE IN RESEARCH PUBLICATIONS THROUGH*

PLAGIARISM CHECKER PREVENTION, DETECTION AND RESPONSE", yaitu [12] :

0% : Tidak ditemukan plagiat
1% - 5% : Plagiat ringan (dapat diterima)
6% - 16% : Plagiat sedang
17% -100%: Plagiat total



Gambar 8. Rancangan tampilan grafik batang hasil pengecekan plagiat

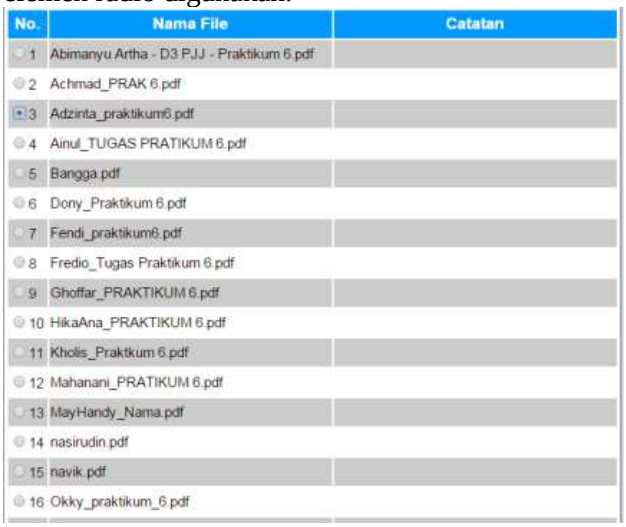
Grafik pada Gambar 8 mewakili hasil penjumlahan total dokumen yang termasuk dalam tiap kriteria. Sehingga didapatkan 4 grafik batang yang mewakili total jumlah dokumen dalam tiap kriteria dalam warna yang berbeda - beda dan diikuti dengan tabel yang berisi jumlah dokumen dalam tiap kriteria.

B. Pembuatan Sistem

Pada pembuatan sistem pendeteksi plagiat berdasarkan rancangan alur sistem yang telah dibuat, dibutuhkan beberapa langkah hingga mencapai hasil akhir pengerjaan, yaitu pembuatan tampilan awal sistem, penerapan ekstraktor dokumen, penghapusan stopword, penerapan stemming, penerapan term frequency (TF), serta penggolongan dokumen dengan naive bayes dan penampilan hasil pengecekan melalui grafik.

Tahap yang pertama adalah menampikan data dokumen tugas yang tersimpan dalam folder. Pada tahap ini, sistem harus mampu menampilkan daftar dokumen pdf pada satu pengumpulan tugas dalam bentuk tabel sebagai tampilan awal sistem.

Tahap berikutnya adalah membuat mekanisme untuk pemilihan dokumen referensi. Pada tahap ini, dari tabel daftar dokumen yang telah dibuat, dikembangkan kembali untuk ditambahkan dengan radio button pada masing – masing nama dokumen tersebut untuk dijadikan sebagai tampilan awal pada sistem. Masing-masing dokumen dapat dijadikan sebagai dokumen referensi. Untuk memastikan hanya satu dokumen yang dijadikan rujukan, maka elemen radio digunakan.



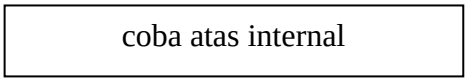
No.	Nama File	Catatan
1	Abimanyu Artha - D3 PJJ - Praktikum 6.pdf	
2	Achmad_PRAK 6.pdf	
3	Adzinta_praktikum6.pdf	
4	Ainul_TUGAS PRATIKUM 6.pdf	
5	Bangga.pdf	
6	Dony_Praktikum 6.pdf	
7	Fendi_praktikum6.pdf	
8	Fredio_Tugas Praktikum 6.pdf	
9	Ghoffar_PRAKTIKUM 6.pdf	
10	HikaAna_PRAKTIKUM 6.pdf	
11	Kholis_Praktkum 6.pdf	
12	Mahanani_PRATIKUM 6.pdf	
13	MayHandy_Nama.pdf	
14	nasirudin.pdf	
15	navik.pdf	
16	Okky_praktikum_6.pdf	

Gambar 9. Tampilan data dokumen untuk dijadikan referensi

Proses yang dilewati oleh dokumen referensi adalah ekstraksi dokumen, penghapusan stopword, proses stemming, perhitungan term frequency (TF) dan penyimpanan hasil perhitungan term frequency. Sehingga pada tahap ini dilakukan penerapan ekstraktor dokumen pdf, penghapusan stopword, penerapan stemming, perhitungan term frequency dan penyimpanan hasil perhitungan term frequency.

Ektraksi dokumen pdf dilakukan dengan menggunakan program yang telah ada yakni pdf to text. Program tersebut akan mengekstrak teks yang terdapat pada dokumen untuk ditampilkan dalam versi html setelah mencantumkan nama file dan lokasi file pdf pada saat memanggil program tersebut mealui skrip program yang dibuat.

Selanjutnya, kalimat yang telah didapatkan akan dipecah kembali menjadi kata. Namun, struktur kalimat masih dipertahankan sehingga kata yang dipecah masih berada pada posisinya dalam kalimat. Untuk itu, dilakukan penyimpanan kedalam array multidimensional, dimana pada array tingkat pertama akan diisi dengan kalimat dalam teks dan pada array tingkat kedua akan diisi dengan kata. Sehingga struktur kata dan kalimat terjaga. Langkah berikutnya adalah penghapusan *stopword* dan proses stemming.



Gambar 10. Contoh kalimat setelah melalui proses penghapusan stopword dan stemming

Penggunaan hubungan antar kata atau relasi kata dalam penelitian ini ditujukan untuk memperbesar kemampuan sistem pendeteksi plagiat untuk mendeteksi kesamaan kata dengan memperhatikan posisi kata tersebut. Karena jika

sistem hanya memperhatikan kata yang sama saja, tanpa memperhatikan posisinya, maka akan timbul kesamaan yang sangat besar karena sistem tidak mempedulikan makna dari kata tersebut. Namun hanya mempedulikan ada atau tidaknya kata tersebut.

```
--coba
coba-atas
atas-internal
```

Gambar 11. Hasil penerapan hubungan kata

Penerapan relasi antar kata yang dilakukan dengan tujuan agar perhitungan kesamaan tidak hanya terpaku pada kata yang ada dalam suatu dokumen melainkan juga mempertimbangkan posisi kata tersebut dalam suatu dokumen yakni pada kalimat dimana kata tersebut berada. Sehingga pada perhitungan *term frequency* dengan relasi antar kata ini didapatkan jumlah kata dengan relasinya atau kata sebelumnya dalam suatu dokumen.

Hasil perhitungan term frequency ini, akan disimpan dalam variabel yang kemudian akan disimpan ke dalam file txt. Data variabel tersebut akan menyimpan kata dan juga jumlah kata dalam satu dokumen referensi.

Proses pengekstrakan dokumen, penghapusan stopword, proses stemming, dan perhitungan term frequency yang dialami oleh dokumen yang dipilih untuk dicek sama seperti yang dialami oleh dokumen referensi.

Tahap berikutnya adalah penggolongan dengan naive bayes. Pada tahap ini, dilakukan dua tahap, yang pertama yakni perhitungan kesamaan isi dokumen yang dicek jika dibandingkan dengan dokumen referensi dan penggolongan dokumen yang dicek berdasarkan kriteria yang telah ditetapkan sesuai dengan hasil pengecekan yang diperoleh.

Untuk melakukan perbandingan kesamaan isi dokumen yang dicek dengan dokumen referensi, data hasil perhitungan *term frequency* dari dokumen referensi yang telah disimpan dalam file data.txt, akan diambil kembali. Kemudian, untuk perhitungan kesamaan isi dokumen, dapat dilihat pada Gambar 12.

Hasil yang diinginkan dalam pengerjaan penelitian ini adalah berupa persentase kesamaan dokumen yang dicek dengan dokumen referensi. Karena data hasil perhitungan yang dibutuhkan harus dalam bentuk persentase, maka hasil perhitungan kesamaan ini dikalikan dengan 100.

No.	Nama File	Catatan
1	Adzinta_praktikum6.pdf	1 % sama dengan referensi
2	Fredio_Tugas Praktikum 6.pdf	16 % sama dengan referensi
3	Suherman_praktikum 6.pdf	55 % sama dengan referensi

Gambar 12. Tampilan hasil pengecekan plagiat

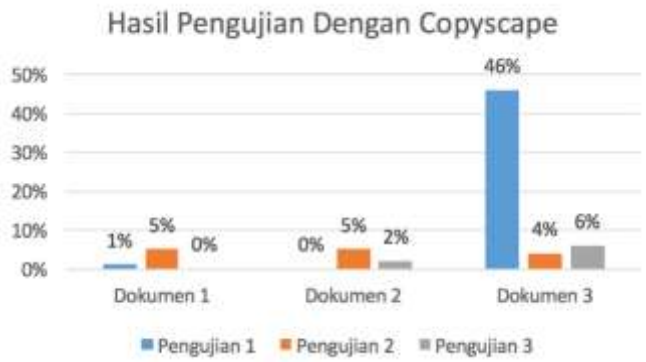
Persentase hasil perhitungan akan dibawa ke fungsi pengelompokan. Kriteria yang digunakan dalam penggolongan dokumen ini dibagi menjadi 4 kelompok, yakni bukan plagiat dengan persentase kesamaan 0%, yang kedua yakni plagiat ringan dengan pesentase kesamaan > 0% dan < 5%, plagiat sedang dengan persentase kesamaan > 5% dan < 17%, dan plagiat berat / total dengan persentase kesamaan > 17%. Dari kriteria yang telah ditetapkan ini, dapat diketahui jika persentase kesamaan diatas 0% sudah dianggap menjadi plagiat.

III. HASIL DAN PEMBAHASAN

3.1. Hasil Pengujian Algoritma

Pengujian deteksi plagiarisme ini dilakukan dengan membandingkannya dengan *software* deteksi plagiat yang sudah ada. Salah satu perangkat lunak yang dipilih sebagai pembanding adalah *Copyscape*. Dokumen akan dicek terlebih dahulu menggunakan Copyscape, selanjutnya dokumen dicek dengan menggunakan deteksi plagiarisme yang dibuat. Pengujian dilakukan dengan memilih satu berkas tugas sebagai file referensi dari sekumpulan berkas lainnya pada tugas yang sama. Selanjutnya dokumen referensi ini akan dibandingkan dengan dokumen-dokumen lainnya pada tugas yang sama.

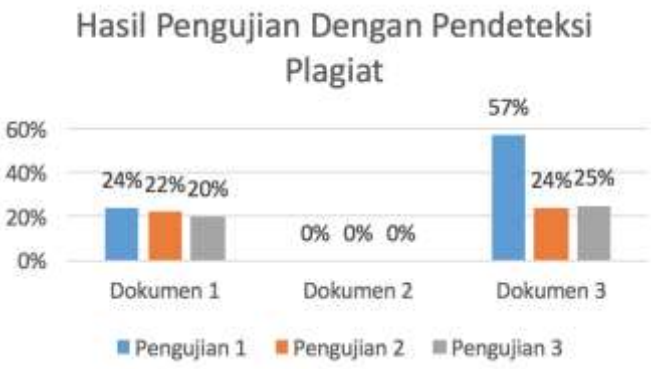
Pengujian dengan Copyscape dilakukan dengan mengunggah dokumen pada website Copyscape. Setiap kali pengecekan, Copyscape hanya mampu membandingkan dua buah dokumen. Pada pengecekan untuk teks, kesamaan yang dideteksi diantara dua teks, merupakan kesamaan beberapa kata pada dokumen yang letaknya berdampingan dan sama seperti pada referensi, baik posisi maupun struktur katanya. Sehingga tidak setiap kata yang sama akan dinilai sebagai plagiat. Akan tetapi, pengecekan plagiat pada Copyscape ini tidak memperhatikan penggunaan stopword maupun karakter pada isi teks. Copyscape hanya memperhatikan kesamaan kata yang berurutan sama seperti pada referensi meskipun pada kesamaan tersebut terselip karakter maupun kata stopword. Selain itu, pada Copyscape ini, tidak dilakukan proses stemming, sehingga masih terdapat banyak kata berimbuhan pada hasil output pengecekan.



Gambar 13. Grafik hasil pengujian dengan Copyscape

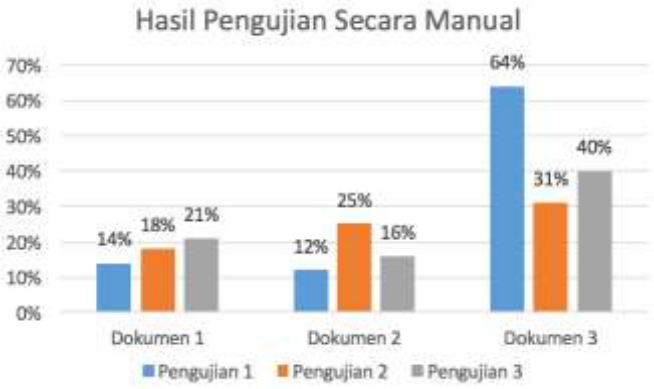
Pada Gambar 4. menunjukkan hasil pengujian dengan menggunakan Copyscape. Pada pengujian pertama, dokumen satu memiliki kesamaan sebesar 1% dengan dokumen referensi. Pada pengujian kedua dan ketiga masing-masing memiliki kesamaan sebesar 5% dan 0%. Sedangkan dokumen 2 memiliki kesamaan sebesar 0% pada pengujian kesatu dan memiliki kesamaan sebesar 5% dan 2% untuk pengujian kedua dan ketiga. Dokumen ketiga memiliki nilai kesamaan yang cukup tinggi pada pengujian kesatu yakni sebesar 46% dan memiliki nilai sebesar 4% dan 6% pada pengujian kedua dan ketiga.

Pengujian berikutnya dilakukan dengan menggunakan dokumen yang sama, akan tetapi pengecekan dilakukan dengan menggunakan aplikasi pendeteksi plagiat yang dibuat. Pada Gambar 5. Menunjukkan hasil pengujian dengan pendeteksi plagiarisme yang dibuat. Pada dokumen 1, hasil pengujian pertama menunjukkan nilai prosentase sebesar 24% dan pada pengujian kedua dan ketiga memperoleh hasil sebesar 22% dan 20%. Sedangkan untuk dokumen 2, pengujian 1, 2 dan 3 semuanya memperoleh hasil 0%. Untuk dokumen 3, hasil pengujian pertama menunjukkan nilai sebesar 57%. Pada pengujian kedua dan ketiga memperoleh hasil 24% dan 25%.



Gambar 14. Hasil pengujian dengan pendeteksi plagiarisme yang dibuat.

Pengujian berikutnya dilakukan secara manual dengan menggunakan naive bayes. Dokumen yang dicek merupakan dokumen yang sama dengan pengecekan sebelumnya. Hasil pengecekan secara manual ini terlihat seperti Gambar 6. Pada hasil pengujian secara manual ini diperoleh hasil untuk dokumen 1 sebesar 14% untuk pengujian pertama. Pada pengujian kedua diperoleh hasil sebesar 18% dan 21% untuk pengujian ketiga. Untuk dokumen 2 diperoleh hasil sebesar 12% pada pengujian pertama. Pada pengujian kedua diperoleh hasil sebesar 25% dan pada pengujian ketiga nilai prosentasenya sebesar 16%.



Gambar 15. Hasil pengujian secara manual

Pada pengujian untuk dokumen 3, diperoleh hasil sebesar 64% untuk pengujian pertama. Pada pengujian kedua dan ketiga diperoleh hasil sebesar 31% dan 40%.

Dari pengujian menggunakan Copyscape, aplikasi plagiat yang dibuat, dan secara manual, terdapat perbedaan hasil yang diperoleh. Misalnya pada percobaan pertama untuk dokumen 1. Dengan menggunakan Copyscape diperoleh hasil sebesar 1%, sedangkan dengan menggunakan pendeteksi plagiat yang dibuat diperoleh hasil sebesar 24%. Pada saat dilakukan deteksi secara manual, diperoleh hasil sebesar 14%.

Dari pengujian satu ini, didapatkan hasil jika pengecekan dengan menggunakan pendeteksi plagiat masih belum bisa mendekati atau bahkan sama dengan hasil pengecekan yang didapatkan dari pengecekan manual. Bahkan terdapat hasil pengecekan yang melebihi hasil pengecekan dengan menggunakan pengecekan manual. Hasil yang melebihi pengecekan manual ini, yakni pada dokumen satu, disebabkan karena pendeteksi plagiat menghasilkan total jumlah kata pada dokumen referensi yang hampir mendekati separuh dari jumlah total kata dokumen referensi yang didapatkan dari pengecekan manual yakni 1079 untuk total kata dari pengecekan manual dan 419 untuk pengecekan dengan pendeteksi plagiat. Sementara untuk total kata yang sama sudah hampir mendekati total kata yang sama dari pengecekan manual yakni dengan selisih 38 kata. Sehingga pengecekan dengan pendeteksi plagiat memperolah hasil yang melebihi persentase dari pengecekan manual.

Kemudian, untuk hasil yang kurang dari persentase yang didapatkan dari pengecekan manual, yakni pada

dokumen tiga, terjadi karena total kata sama yang diperoleh jauh berbeda dengan total kata sama yang diperoleh dari pengecekan manual. Sedangkan untuk total kata pada dokumen referensi sama dengan pada pengujian untuk dokumen satu. Selisih dari total kata sama yang diperoleh dengan total kata sama dari pengecekan manual adalah 454 kata. Sehingga hasil yang didapatkan pun masih belum bisa mendekati hasil dari pengecekan manual.

Kemudian, pada pengujian ketiga, didapatkan hasil pengecekan pada dokumen dua yang selalu menghasilkan persentase kesamaan 0% yang menunjukkan jika dokumen tersebut tidak sama dengan dokumen referensi. Dari hasil yang diperoleh tersebut, diketahui jika sistem tidak dapat mengekstrak isi dokumen tersebut. Hal ini dapat disebabkan karena sistem hanya mampu mengekstrak isi dokumen pdf yang dihasilkan dari beberapa converter tertentu. Pada pengujian ini, diketahui jika sistem tidak dapat mengekstrak isi dokumen pdf yang dihasilkan dari Nitro Reader 3 dan bekerja sangat baik untuk dokumen pdf dari Microsoft Word.

IV. KESIMPULAN

Dari hasil pengujian, aplikasi pendeteksi plagiarisme yang dibuat dapat mendeteksi kesamaan antar dokumen dan menghasilkan nilai prosentase plagiarisme. Beberapa hasil pengujian, sistem tidak dapat menghasilkan nilai yang akurat karena dipengaruhi oleh kemampuan sistem dalam mengekstrak teks dari dokumen pdf. Hal ini disebabkan dokumen pdf tugas mahasiswa merupakan hasil *convert* dari beragam software pengolah kata. Untuk dokumen-dokumen yang dieksport dari Miscrosoft Word dapat diekstrak dengan baik.

REFERENSI

[1] Departemen Pendidikan dan Kebudayaan., *Kamus Besar Bahasa Indonesia*, Balai Pustaka, Jakarta, 1990.

[2] Paul, M., & Jamal, S., *An Improved SRL Based Plagiarism Detection Technique using Sentence Rangking*, International Conference On Information and Communication Technologies, Procedia Computer Science 46, 223-230 (2015).

[3] Isa, T.A, & Abidin, T.F., *Mengukur Tingkat Kesamaan Paragraf Menggunakan Vector Space Model untuk Mendeteksi Plagiarisme*, Seminar Nasional dan Expo Teknik Elektro 2013.

[4] Herqutanto., 2013, Plagiarisme, Runtuhnya Tembok Kejujuran Akademik. Plagiarisme. Volume 1, No. 1. http://journal.ui.ac.id/index.php/eJKI/article/download/1589/1335, 13 Januari 2015.

[5] Indranandita, Amalia., Budi, Susanto., & Antonius, Rahmat C, November, 2008, Sistem Klasifikasi dan Pencarian Jurnal dengan Menggunakan Metode Naive Bayes dan Vector Space Model. Jurnal Informatika. Volume 4, No. 2

[6] Susanto, Dwi. *Autonomous Intelligent Agent Question Answer System Using Minimal Differentiator Expressions*. Tesis S2 Jaringan Cerdas Multimedia, Teknik Elektro ITS. 2013.

[7] Sridhar, K., Selvan, R. Saravana Subbu., Prabhu, V., *Role of Librarian in Quality Sustenance in Research Publications Through Plagiarism Checker Prevention, Detection and Response*, International Journal of Library and Information Science (IJLIS), Volume 3, 2014.

[8] Indranandita, Amalia., Susanto, B., & Rahmat, A., Sistem Klasifikasi dan Pencairan Jurnal Dengan Menggunakan Metode Naive Bayes dan Vector Space Model, Jurnal Informatika, Volume 4 No.2, 2008.

[9] Wicaksono, D.W., Irawan, M.I., & Rukmi, A.M., *Sistem Deteksi Kemiripan Antar Dokumen Teks Menggunakan Model Bayesian Pada Term Latent Semantic Analysis*, Jurnal Sains dan Seni POMITS, Volume 3, No. 2, 2014.

[10] Indriani, Aida., 2014, "Klasifikasi Data Forum dengan Menggunakan Metode *Naïve Bayes Classifier*", Seminar Nasional Aplikasi Teknologi Informasi (SNATI), Yogyakarta