DETEKSI PLAGIARISME DOKUMEN TEKS MENGGUNAKAN ALGORITMA SCAM

TEOFILUS JATI ASMARALOKA
Program Studi Teknik Informatika – S1, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro
Jl. Nakula 1 no 5-11 Semarang 5013, Telp. (024) 3517261, URL : http://dinus.ac.id/, email : jatiezx1@gmail.com

Abstract
Problems of plagiarism is one of the classic problems in education now includes a student dishonesty behavior has always been a picture for education in Indonesia. Cases of plagiarism in schools is common, it's become a benchmark for quality education. Copy and paste action seems to be required in fulfilling the ritual duties of teachers. The Act of plagiarism can make someone being lazy to become creative because they do not want to think. For that Act of plagiarism to be pressed, and one way is by detecting plagiarism of text documents. Based on the above method, it's been a lot made of plagiarism detection system on a text document. And one of them using an algorithm to detect plagiarism in SCAM a text document. The SCAM broke the algorithm a document into words. Then added into the sequence of the index and is used for comparison with the new document input. In addition to assist teachers in assessing tasks that are in the shape of a text document. Expected students begin to make the task of its own text documents without having to perform acts of plagiarism.

Key word = plagiarism, SCAM algorithm, text document

Abstrak
Problema plagiarisme merupakan salah satu persoalan klasik dalam dunia pendidikan sekarang ini termasuk perilaku ketidakjujuran siswa yang selalu menjadi momok bagi pendidikan di Indonesia. Tindakan plagiarisme tersebut dapat membuat seseorang menjadi malas berkreatifitas karena tidak mau berfikir. Untuk itu tindakan plagiarisme harus ditekan, dan salah satu caranya adalah

dengan mendeteksi plagiarisme dokumen teks. Berdasarkan metode di atas, sudah banyak dibuat sistem pendeteksi plagiarisme pada dokumen teks. Dan salah satunya menggunakan Algoritma SCAM untuk mendeteksi plagiarisme pada dokumen teks. Algoritma SCAM memecah sebuah dokumen menjadi potongan kata. Kemudian ditambahkan ke dalam urutan indeks dan digunakan untuk perbandingan dengan masukan dokumen yang baru. Selain untuk membantu pengajar dalam menilai tugas – tugas yang berbentuk dokumen teks. Diharapakan siswa mulai membuat tugas yang berupa dokumen teks sendiri tanpa harus melakukan tindak plagiarisme.

Kata Kunci = plagiarism, algoritma SCAM, dokumen teks

1. PENDAHULUAN

1.1 Latar Belakang

Problema plagiarisme merupakan salah satu persoalan klasik dalam dunia pendidikan sekarang ini. Menurut UU No 20 Tahun 2003 tersebut sangat jelas bahwa, pendidikan pada hakekatnya adalah mengembangkan potensi diri peserta didik dengan dilandasi oleh kekuatan spiritual keagamaan, pengendalian diri, kepribadian, kecerdasan, akhlak mulia, serta keterampilan. Dengan demikian, pendidikan mempunyai peran yang strategis dalam membangun karakter siswa. Perilaku ketidakjujuran siswa adalah fenomena plagiarisme yang selalu menjadi momok bagi pendidikan di Indonesia. Kasus plagiarisme di sekolah menjadi hal yang biasa, ini menjadi tolak ukur bagi kualitas pendidikan.

Tindakan plagiarisme tersebut dapat membuat seseorang menjadi malas berkreatifitas karena tidak mau berfikir. Untuk itu tindakan plagiarisme harus ditekan, dan salah satu caranya adalah dengan mendeteksi plagiarisme dokumen teks.

Berdasarkan uraian di atas, sudah banyak dibuat sistem pendeteksi plagiarisme pada dokumen teks . Dan salah satunya menggunakan Algoritma SCAM untuk mendeteksi plagiarisme pada dokumen teks. Algoritma SCAM memecah dokumen menjadi potongan kata. Kemudian ditambahkan ke dalam urutan indeks dan digunakan untuk perbandingan dengan masukan dokumen yang baru. Dan menurut uraian di atas peneliti tertarik untuk mengadakan penelitian mendeteksi plagiarisme pada dokumen teks menggunakan Algoritma.

1.2 Rumusan Masalah

Berdasarkan latar belakang pemilihan judul, maka yang menjadi permasalahan adalah bagaimana algoritma SCAM digunakan untuk mendeteksi plagiarisme pada dokumen teks secara efektif.

1.3 Batasan Masalah

Mengingat keterbatasan waktu dan tenaga, agar penulisan Penelitian Tugas Akhir ini dapat terfokuskan dan lebih terarah untuk memudahkan dalam pemahaman masalah yang ada, maka penulis membatasi masalah Mendeteksi Kemungkinan Plagiarisme Dokumen Teks menggunakan Algoritma SCAM adalah dokumen yang digunakan berformat *.txt, *.doc, *.docx, *.rtf.

1.4 Tujuan

Berdasarkan rumusan masalah dan batasan masalah yang ada, maka dapat dideskripsikan tujuan dari penelitian ini adalah untuk mengaplikasikan Algoritma SCAM yang digunakan untuk mendeteksi kemungkinan plagiarisme pada dokumen teks ,sehingga mampu memaksimalkan originalitas dalam penulisan.

1.5 Manfaat

Dengan adanya Penelitian Tugas Akhir ini penulis berharap dapat memberikan manfaat bagi beberapa pihak. Adapun manfaat yang dapat diperoleh adalah sebagai berikut:

1. Bagi Penulis
   - Menambah pengetahuan tentang apa dan bagaimana parameter dalam penentuan apakah suatu dokumen memiliki kesamaan atau tidak dengan dokumen lain yang dimiliki sebelumnya.
   - Menambah pengetahuan tentang bagaimana membangun webservice berbasis PHP yang memiliki enginependeteksi plagiarisme dokumen.
   - Menambah pengetahuan tentang bagaimana memanfaatkan service dari suatu webservice pada aplikasi client berbasis java dan PHP.
2. Bagi Akademik
   - Menambah ragam kepustakaan akademik, juga dapat memberikan informasi kepada pembaca sebagai bahan referensi bagi yang berminat mengembangkan sistem pendeteksi plagiarisme pada suatu teks dokumen.
   - Memberikan akses kepada siswa yang ingin memeriksa apakah tugas yang akan dibuat memiliki kesamaan dengan tugas akhir-tugas yang sebelumnya telah dibuat.

2. TINJAUAN PUSTAKA

2.1 Algoritma SCAM

SCAM adalah singkatan dari Stanford Copy Analysis Mechanism. dimana menjadi biasanya tolak ukur relatif untuk mendeteksi overlapping dengan membuat perbandingan pada satu set kata-kata yang umum antara dokumen tes dan dokumen terdaftar.

Secara general algoritma ini bekerja dengan memberikan pendekatan-pendekatan khusus untuk mendeteksi kesamaan antara dokumen. Dan langkah-langkah deteksi tersebut, dibagi menjadi empat langkah utama, yaitu:

Pengindeksan dataset: Dataset ini terdiri dari beberapa dokumen yang preproses dan di indeks.

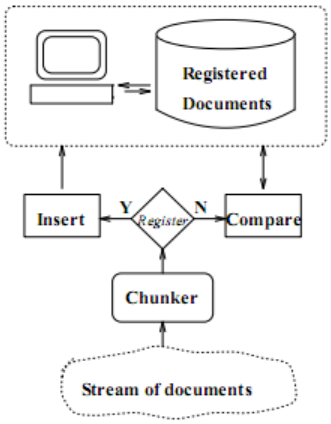
Pengolahan dokumen uji: Dokumen yang akan diuji diberikan dalam input diproses untuk proses tokenize yang akan dimasukkan dalam daftar kata yang akan diolah nantinya.

Pencarian pada indeks: Indeks akan digunakan untuk mengambil kecocokan antara dokumen uji dan dokumen milik dataset.

Mengevaluasi kesamaan: Menggunakan rumus SCAM. perbandingan kesamaan dokumen, nantinya akan berdasarkan dokumen uji dan dokumen milik dataset.

2.2 Copy Detection Preliminaries

Pada bagian ini menyajikan arsitektur dari Server deteksi copy generik dan memperkenalkan relevan terminologi. Pada bagian ini memberikan ringkasan singkat dari beberapa isu-isu yang perlu dipertimbangkan saat membangun copy deteksi server seperti struktur data, dan unit tekstual yang digunakan untuk perbandingan.



Gambar 2.3 Copy Generik Deteksi Server

2.3 Penyimpanan Indeks Inverted

Kami mengusulkan menggunakan struktur indeks terbalik (seperti dalam sistem IR tradisional) untuk menyimpan potongan dokumen terdaftar . Indeks dari potongan dalam kosa kata tersebut dibangun dan dipelihara pada saat regis - trasi . Setiap entri untuk menunjuk ke satu set posting yang menunjukkan dokumen mana yang dipotong . Setiap posting untuk sepotong wi diberikan memiliki dua atribut (docnum , frekuensi) , di mana docnum adalah identifikasi unik dari sebuah dokumen yang terdaftar , dan frekuensi adalah jumlah kemunculan wi dalam dokumen dengan ID docnum . Dalam Gambar 2 , kita mengilustrasikan struktur indeks dengan tiga dokumen terdaftar. Huruf " a " melalui " e " , yang mewakili potongan dalam dokumen , merupakan vocabulary dengan N = 5 . Misalnya , potongan " d " memiliki dua posting mewakili yang terjadi sekali dalam dokumen D2 dan D3 dua kali dalam dokumen .

Ketika D dokumen yang akan dibandingkan dengan dokumen terdaftar, potongan D mendongak dalam indeks dokumen terdaftar . Ini berarti bahwa hanya dokumen yang tumpang tindih di tingkat potongan akan dianggap menggunakan mekanisme indeks ini . Maka jumlah total pencarian pada indeks adalah jumlah potongan yang berbeda yang terjadi dalam dokumen D.

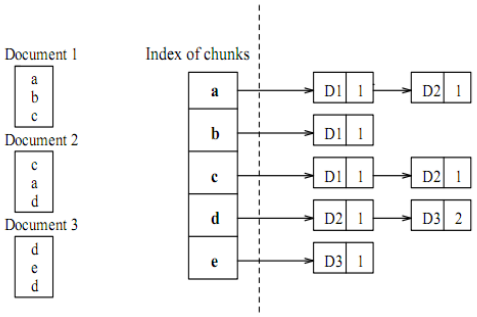


Figure 2: Inverted index storage mechanism.

Gambar 2.4 Inverted index storage mechanism

2.4 Unit Chunking

Seperti yang didefinisikan sebelumnya , chunking melibatkan putus dokumen ke unit yang lebih primitif seperti para- grafik , kalimat , kata atau kalimat yang tumpang tindih . Unit chunking dipilih untuk deteksi copy critical karena membentuk tumpang tindih dapat diuraikan di bawah .

- Similiarity Level : Semakin besar

unit chunking semakin rendah probabilitas dalam dokumen yang tidak terkait . Misalnya, dua hal dokumen-dokumen mungkin keduanya memiliki kalimat seperti " Ini ulang pencarian didanai oleh NSF " sebagai bagian dari ayat . Jika unit chunking adalah sebuah paragraf, dua dokumen mungkin tidak akan terdeteksi sebagai tumpang tindih, sementara mereka akan terdeteksi jika unit chunking adalah sebuah kalimat . Di sisi lain , semakin besar unit chunking , semakin tinggi kemungkinan hilang tumpang tindih yang sebenarnya . Sebagai contoh, mempertimbangkan dua paragraf yang berbagi 5 dari 6 kalimat identik. Dengan ayat chunking , tidak cocok akan terdeteksi , sedangkan dengan kalimat chunking, 5 dari kemungkinan 6 unit akan terdeteksi.

• Search Cost : Semakin besar potongan tersebut , semakin tinggi potensi jumlah potongan yang berbeda yang akan disimpan . Misalnya , sebagai koleksi doc - uments tumbuh , kami berharap jumlah kalimat yang berbeda yang akan disimpan lebih tinggi dari jumlah kata yang berbeda . Hal ini karena di luar titik tertentu jumlah kata-kata baru diperkenalkan ke kosakata akan rendah dibandingkan dengan pertumbuhan hampir linear kalimat / paragraf, maka kita melihat bahwa poteni ukuran ial dari indeks potongan lebih tinggi bila unit chunking dipilih adalah lebih besar .

2.5 Overlap Measure

Dalam skema IR tradisional , ketika permintaan datang dari pengguna , query " dibandingkan " terhadap dokumen , dan beberapa ukuran relevansi antara dokumen dan query diperoleh . Demikian pula , kita perlu membangun metrik yang mengukur tumpang tindih antara dokumen yang masuk dan dokumen pra-terdaftar. Pada bagian ini, kita mempertimbangkan model yang populer digunakan dalam sistem IR disebut Vector Space Model (VSM) yang berhubungan dokumen untuk pertanyaan , dan melihat mengapa hal ini tidak langsung diterapkan untuk deteksi copy . Kami kemudian mengusulkan Frekuensi Model Relatif (RFM) yang menyajikan suatu kerangka kerja yang lebih baik untuk mendeteksi tumpang tindih . Untuk diskusi kita, D merujuk ke dokumen generik (terdaftar atau baru) . Kami mendefinisikan vektor terjadinya 0 (D) menjadi daftar dari potongan di D. Misalkan F (D) (ukuran N) menjadi vektor frekuensi, di mana F1 (D) adalah jumlah kemunculan sepotong wi di D. Mari sim (di , D.)) menunjukkan ukuran kesamaan antara dokumen di dan 1 .) 2 , sebagaimana dihitung di bawah ini.

Untuk menggambarkan perhitungan kesamaan, pertimbangkan dokumen terdaftar R dan dokumen S1 ' baru yang akan dibandingkan dengan R. Mari O (R) = <a , b , c > , O (S1) = <a , b > , O (S2) = <a , b , c > , dan O (S3) = < ak > , di mana k > 1 menunjukkan jumlah dari dalam dokumen S3 (yaitu Fa (S3) = k) . Juga biarkan O (S4) = C a, b , c , d , c , f , g , h> Asumsikan bahwa kosakata 1.17 = {a , b , c , d , e , f , g , h.) . Kami harapkan skema deteksi copy baik untuk melaporkan 1 ? dan SI untuk menjadi " cukup " sama, R dan S2 menjadi replika yang tepat , O dan S3 untuk menjadi agak mirip pada nilai k rendah tetapi tidak mirip dengan k tinggi , dan S4 memiliki tumpang tindih yang signifikan dengan R.

Vector Space Model

Sebuah model populer dalam domain IR, adalah model VSNI. Mengingat permintaan dengan bobot yang sesuai, produk dot terjadinya vektor tertimbang dari query dengan dokumen yang disimpan dan dihitung: jika nilai produk dot melebihi batas tertentu, dokumen ditandai untuk mencocokkan query. Sebuah skema pembobotan common yang digunakan adalah normalisasi hitung frekuensi. Jika kita menerapkan ukuran ini untuk deteksi copy, vektor frekuensi normalisasi akan V(R) =<1/3, 1/3, 1/3, 0,>, dan V(S1) 1/2, 1/2, 0,0, ...> Dan sebagainya.Kesamaan antara II dan Si akan
sim(R.S1)=1/3*1/2±1/3*1/2=1/3.
Demikian sim(R.S2) =1/3, sim(R.S3) = 1/3 dan sim(R.S4) =1/8. Karena R tumpang dengan S2, dan S4l ebih signifikan dibandingkan dengan S1 atau S3, untuk mengukur bobot populer digunakan adalah ukuran kesamaan co-sinus yang mendefinisikan relevansi dokumen R untuk query Q menjadi

$$sim(R,Q) = \frac{\sum_{i=1}^{N} \alpha_i^2 * F_i(R) * F_i(Q)}{\sqrt{\sum_{i=1}^{N} \alpha_i^2 * F_i^2(R) * \sum_{i=1}^{N} \alpha_i^2 F_i^2(Q)}}$$

Semakin tinggi frekuensi kata, kurang kata kontribusi terhadap pencocokan kesamaan. Jika kita menggunakan ukuran ini untuk contoh kita dan menganggap bobot seragam untuk kata-kata (α =1), kita menemukan bahwa sim(R, S1) =(1*1+1*1) A/772=0,82 dan sim(R,S2)=1. (Dalam perhitungan menggunakan vektor frekuensi unnormalized, tapi hasilnya sama jika mereka normal.) Dalam hal ini, metrik muncul untuk bekerja dengan baik.

Setelah persamaan matematika yang telah ada, kemudian didefinisikan subset measure dari dokumen D1 dengan subset dari dokumen D2 menjadi,

$$\text{subset}(D_1, D_2) = \frac{\sum_{w_i \in c(D_1, D_2)} \alpha_i^2 * F_i(D_1) * F_i(D_2)}{\sum_{i=1}^{N} \alpha_i^2 F_i^2(D_1)}$$

Kemudian didefiniskan untuk mengukur kesamaan antara dua dokumen yaitu :

$$\text{sim}(R,S) = \max\{\text{subset}(R,S), \text{subset}(S,R)\}$$

Jika $\text{sim}(R,S) \geq 1$, maka nilai dari $\text{sim}(R,S)$ akan menjadi 1. Karena tidak ada informasi tambahan yang didapat ketika $\text{sim}(R,S)$ menghasilkan lebih besar dari 1. Nilai maximum dalam hal ini adalah 1 karena untuk mencerminkan tingkat kesamaan antara dua dokumen dengan range 0 sampai 100%.

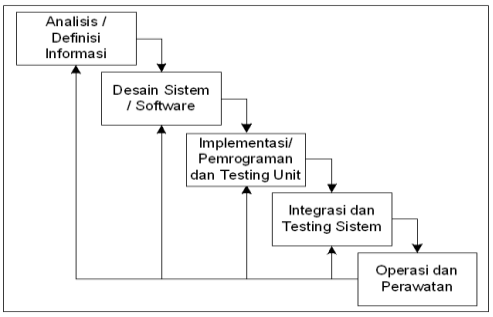
3. METODE PENELITIAN

3.1 Metode Pengumpulan Data

1. Oservasi
2. Studi Pustaka (*Library Research Method*)

3.2 Metode Pengembangan Sistem

Model waterfall yaitu suatu metodologi pengembangan perangkat lunak yang mengusulkan pendekatan kepada perangkat lunak sistematik dan sekuensial yang mulai pada tingkat kemajuan sistem pada seluruh analisis, design, kode, pengujian dan pemeliharaan. Jika telah memasuki tahap selanjutnya dalam project ini, maka anda tidak dapat kembali.

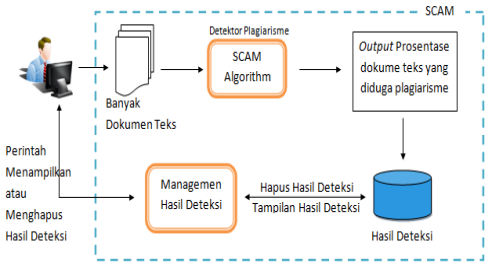


Gambar 2. Model *Waterfall*

4. IMPLEMENTASI DAN PENGUJIAN

Detektor plagiarisme yang dibangun akan diimplementasikan dan diberi nama ***Aplikasi SCAM***.

4.1 Arsitektur Aplikasi SCAM



Gambar 3. Arsitektur Aplikasi

4.2 Implementasi CM Algorithm

1. Scanner

Pada Proses ini dilakukan proses chunking, dimana fungsi scanner_scam melakukan proses penyederhanaan teks menjadi kata, dalam hal ini sesuai dengan kebutuhan SCAM Algorithm.

```
function scanner_scam (source : string) →array of string
{
  input  : kumpulan teks suatu dokumen teks
  proses : memotong-motong (chunking) teks menjadi kata
  output : mengembalikan sebuah array yang berisi kumpulan kata
}
KAMUS
  hsl   : array of string
  coba : string

ALGORITMA

  coba = strtolower(source)
  hsl ← preg_split("/[\s\,\-\.\"\:\/\(\)?]+/", coba)

→ hsl
```

Gambar 4. Algoritma Scanner SCAM

2. Parser

Hasil dari scanner yang berupa token-token dimasukkan kembali dalam proses parser. Dalam parser SCAM token-token hasil scanner dproses kembali apabila mengandung arti lain, misal tanda baca, kata dengan satu huruf, angka dan lain-lain.

```
function parser_scam (source : array of string) →array of string
{
  input  : sebuah array of kata yang utuh
  proses : menormalisasikan kata sesuai grammar SCAM
  output : mengembalikan sebuah array yang berisi kumpulan token/kata yang sudah ternormalisasi
}
KAMUS
  awal   : array of string
  hsl1   : array of string

ALGORITMA
  awal ← oneword(source)
  hsl1 ← normalisasi(awal)

→ $hsl1
```

Gambar 5. Algoritma Parser SCAM

3. Normalisasi

Proses ini digunakan untuk mengambil token/kata yang unik atau menghilangkan kata yang sama dalam array, sehingga token/kata yang tersimpan dalam array tidak ada redudansi token/kata.

```
function normalisasi(source : array of string) →array of string
{
  input  : token sequence hasil parser
  proses : menghilangkan token/kata yang sama pada array source
  output : token/kata dalam array unik
}
KAMUS
  hsl : array of token

ALGORITMA
  hsl ← array_unique(source)

→ hsl
```

Gambar 5. Algoritma Preparation

4. Mengambil Irisan Token

Proses ini digunakan untuk mencari irisan token/kata pada 2 buah kumpulan token dari dokumen teks yang dibandingkan, karena beberapa fungsi untuk memanipulasi array sudah terdapat pada library code igniter sehingga lebih memudahkan dalam penggunaannya.

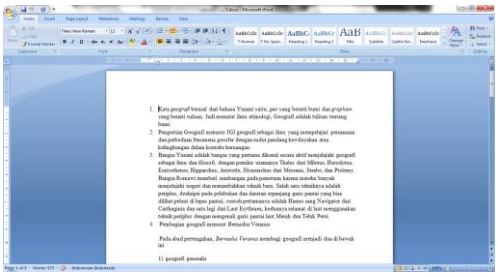
5. Hitung Prosentase

Menghitung nilai kesamaan dari 2 dokumen teks menggunakan rumus subset yang sudah dibahas pada bab tinjauan pustaka.

4.3 Pengujian

1. Proses Chunking / Tokenisasi

Simulasi chunking dilakukan dengan menginputkan alamat direktori tugas dokumen teks semua siswa pada tugas tertentu pada suatu kelas. Dengan asumsi pengumpulan tugas sudah dilakukan terlebih dahulu. Gambar 6 merupakan contoh dokumen teks yang akan dijadikan sebagai input data pada proses chunking.



Gambar 6. Contoh *dokumen teks*

Input data berupa dokumen teks buatan tiap siswa, satu per satu akan diproses dalam tahapan chunking pada SCAM algorithm. Proses yang pertama yaitu tokenisasi menggunakan kelas scanner dan parser, maka akan menghasilkan urutan token seperti yang terlihat pada gambar 7.

```
algoritma|pengantar|algoritmapendahuluanalgoritma|memegang|peranan|penting|dalam|bidang|
pemrograman|sebegitu|pentingnya|suatu|algoritma|sehingga|perlu|dipahami|konsep|dasar|alg
oritma|apalagi|untuk|seoran|programmer|tentu|diperlukan|suatu|algoritma|sehingga|dapat|m
embuat|program|yang|lebih|efektif|dan|efisien|bagi|kebanyakan|orang|algoritma|sangat|mem
bantu|dalam|memahami|konsep|logika|pemrograman|algoritma|adalah|kumpulan|instruksi|yang|
dibuat|secara|jelas|untuk|menunjukan|langkah|langkah|penyelesaian|suatu|masalah|pada|umu
mnya|algoritma|kurang|lebih|sama|dengan|suatu|prosedur|yang|sering|dilakukan|setiap|hari
|misalnya|prosedur|untuk|menggant|ban|bocor|perlah|prosedur|pemakaian|telepon|umum|prose
dur|membuat|kue|dan|lain|lain|dalam|bidang|komputer|misalnya|edp|elektronik|data|process
ing|atau|mis|management|information|system|algoritma|sering|dimanfaatkan|untuk|menyelesa
ikan|suatu|masalah|atau|untuk|proses|pengambilan|keputusan|seorang|sistem|analisis|anali
sist|system|tentunya|menggunakan|algoritma|untuk|merancang|suatu|sistem|bagi|seorang|pro
gramer|algoritma|digunakan|untuk|membuat|modul|modul|program|guna|memahami|suatu|algorit
ma|harus|dimiliki|pengetahuan|dasar|matematika|karena|pada|dasarnya|algoritma|lahir|dari
|konsep|logika|matematika|disini|yang|perlu|dilatih|adalah|kemampuan|logikanya|agar|bena
r|benar|bisa|menyusun|langkah|langkah|penyelesaian|masalah|dengan|baik|
```

Gambar 7. Hasil tokenisasi *source code*

Setelah semua file dokumen teks dilakukan tokenisasi melalui scanner dan parser, kemudian untuk perhitungan prosentase kesamaan antara 2 dokumen teks maka diperlukan irisan token dari 2 teks dokumen yang dibandingkan. Hasil irisan token antara dokumen “algoritma.docx” dan “deni.docx” lihat gambar 8.

```
suatu|untuk|dapat|yang|dan|adalah|secara|pada|dengan|sering|atau|proses|
sistem|menggunakan|digunakan|karena|dari|
```

Gambar 8. Hasil irisan token antara 2 dokumen teks

Kemudian mulai dilakukan perhitungan untuk menghitung prosentase kesamaan 2 dokumen teks. Dimulai dengan subset(algoritma.docx,deni.docx), 1 per 1 kata yang terdapat pada irisan token digunakan untuk menghitung ada berapa kata tersebut dalam 2 dokumen teks yang dibandingkan. Dimulai dengan kata “suatu” pada dokumen “algoritma.docx” terdapat 7 buah kata dan pada dokumen “deni.docx” terdapat 3 buah kata dan seterusnya.

$$\text{subset}(D_1,D_2) = \frac{\sum_{w_i \in c(D_1,D_2)} \propto_i^2 F_i(D_1) * F_i(D_2)}{\sum_{i=1}^{N} \propto_i^2 F_i^2(D_1)}$$

$$= \frac{(7*3)+(7*2)+(1*4)+(4*9)+(2*3)+(2*3)+(1*1)+(2*4)+(2*2)+(2*1)+(2*2)+(1*1)+(2*3)+(1*1)+(1*3)+(1*1)+(1*4)}{176}$$

$$= \frac{21+14+4+36+6+6+1+8+4+2+4+1+6+1+3+1+4}{176}$$

$$= \frac{122}{176}$$

$$= 0.693182$$

$$\text{subset}(D_2,D_1) = \frac{\sum_{w_i \in c(D_2,D_1)} \propto_i^2 F_i(D_1) * F_i(D_2)}{\sum_{i=1}^{N} \propto_i^2 F_i^2(D_2)}$$

$$= \frac{(7*3)+(7*2)+(1*4)+(4*9)+(2*3)+(2*3)+(1*1)+(2*4)+(2*2)+(2*1)+(2*2)+(1*1)+(2*3)+(1*1)+(1*3)+(1*1)+(1*4)}{185}$$

$$= \frac{21+14+4+36+6+6+1+8+4+2+4+1+6+1+3+1+4}{185}$$

$$= \frac{122}{185}$$

$$= 0.6594594$$

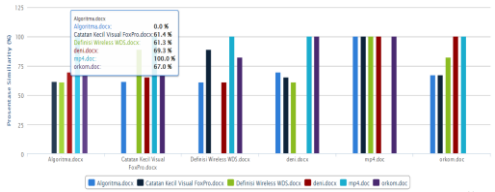
$$sim(D_1,D_2) = \max\{subset(D_1,D_2), subset(D_2,D_1)\}$$

$$= \max\{0.693182, 0.6594594\}$$

$$= 0.693182$$

Jadi Prosentase antara file dokumen “algoritma.docx” dan “deni.docx” adalah 0.693182 * 100 % yaitu 69.3182 %.

Perhitungan diatas merupakan contoh perhitungan yang hanya menghitung 2 dokumen saja. Lihat gambar 9 yang merupakan hasil deteksi plagiarisme dokumen teks secara keseluruhan.



Gambar 9 Hasil prosentase deteksi plagiarisme dokumen teks

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari kegiatan-kegiatan yang dilakukan terkait dengan pelaksanaan Tugas Akhir, dapat diambil kesimpulan sebagai berikut:

1. Aplikasi SCAM mampu mendeteksi kemungkinan plagiarisme pada dokumen teks, sehingga mampu memaksimalkan originalitas dalam penulisan.
2. Hasil deteksi plagiarisme SCAM sebaiknya tidak menjadi keputusan final mengenai kasus-kasus plagiarisme. Hasil deteksi hendaknya digunakan untuk membantu pengajar ketika melakukan investigasi lebih cermat, sehingga keputusan akhir tetap ditentukan oleh pengajar.
3. Dengan adanya upaya deteksi plagiarisme, diharapkan tingkat plagiarisme yang terjadi akan berkurang. Diharapkan dapat lebih berusaha dalam mengerjakan secara mandiri sehingga dapat meningkatkan prestasi akademik mahasiswa tersebut. Di sisi lain, munculnya detektor otomatis ini dapat mendorong dikembangkannya aplikasi penghindar deteksi plagiarisme yang lebih memudahkan mahasiswa dalam menyalin dokumen teks orang lain dan membuat modifikasi-modifikasi yang tidak mampu ditangani detektor. Walaupun begitu, studi mengenai deteksi plagiarisme akan terus berkembang sehingga berbagai upaya penyembunyian tindak plagiarisme akan dapat dideteksi. Penanaman norma dan etika akademik tetap merupakan bagian penting dari pengajaran.

5.2 Saran

Adapun saran terkait pelaksanaan Tugas Akhir ini adalah sebagai berikut:

1. Untuk ke depannya bisa dilakukan perubahan dalam hal pemanfaatan temporary data file di memory dalam penggunaan SCAM Algorithm, sehingga hanya memanggil dan menyimpan data yang diperlukan selama komputasi dari memory yang sudah disiapkan. Hal ini akan membantu kecepatan proses SCAM Algorithm karena load dan save data dilakukan pada saat proses dan tidak menggunakan database secara fisik. Dengan demikian SCAM Algorithm yang dikonstruksi dengan fungsi dan prosedur dapat menangani komputasi lebih cepat.
2. Aplikasi SCAM bisa diexplorasi lebih jauh, tidak hanya menghasilkan prosentase namun yang lebih mendetail lagi, seperti menampilkan bagian-bagian dokumen teks yang dianggap plagiat.

3. Dapat dikembangkan detektor plagiarisme dengan algoritma lain yang bisa dikombinasikan dengan algoritma SCAM.

DAFTAR PUSTAKA

[1] Edria Albert Varian W. (2007). Aplikasi Program Dinamis Dalam String Alignment Untuk Melakukan Pencocokan Dna, Makalah If2251 Strategi Algoritmik. Program Studi Teknik Informatika Institut Teknologi Bandung.

[2] Roger S. Pressman. (2005). Software Engineering : A Practitioner's Approach Sixth Edition.

[3] Mike Joy, Georgina Cosma, Jane Sinclair, Jane Yin-Kim Yau1. (2009). *A Taxonomy Of Plagiarism In Computer Science*. Proceedings of EDULEARN 09 Conference. 6th-8th July 2009, Barcelona, Spain.

[4] Ohno, A. and Murao, H. (2007). Measuring Source Code Similarity Using Reference Vectors, *International Journal Of Innovative Computing, Information And Control*, Vol.3, No.3, Pp.525-537.

[5] Ohno, A. and Murao, H. (2009). A new similarity measure for in-class source code plagiarism detection, *International Journal of Innovative Computing, Information and Control*, vol.5, no.11(B), pp.4237- 4247.

[6] Noersasongko, Edi. (2006). Peraturan Akademik Institut Universitas Dian Nuswantoro. Draf Keputusan Rektor No. .../KEP/UDN-01/XII/2006.

[7] Ohno, A. and Murao, H. (2011). A Two-Step In-Class Source Code Plagiarism Detection Method Utilizing Improved CM Algorithm And Sim, *International Journal of Innovative Computing, Information and Control*, vol.7, no.8, pp.4729- 4739.

[8] Prechelt, Lutz; Malpohl, Guido; Phlippsen, Michael (2005). *JPlag: Finding plagiarisms among a set of programs*. Fakultät für Informatik Technical Report 2000-1, Universität Kalrsruhe, Karlsruhe, Germany.

[9] Techterms.org(2006).Token. <http://www.techterms.org/definition/token> tanggal akses 1 Desember 2013