Deteksi Plagiat Tugas Akhir dengan Metode *Jaccard Similarity*

Sapto Utomo, Imam Much Ibnu Subroto, Andi Riansyah
Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Sultan Agung

Correspondence Author: imam@unissula.ac.id

Abstrak

Syarat menyelesaikan Tugas Akhir sebelum sidang Tugas Akhir harus membuat Laporan Tugas Akhir dimana mahasiswa dalam pembuatan Laporan Tugas Akhir bisa melihat Laporan Tugas Akhir dari kakak tingkat yang sudah menyelesaikan Laporan Tugas Akhir yang sudah disetujui oleh Dosen Pembimbing atau juga bisa mencari referensi lain diinternet baik jurnal maupun paper. Untuk mencegah adanya unsur plagiarisme dalam pembuatan Laporan Tugas Akhir mahasiswa tidak boleh meniru persis kata atau kalimat yang akan dijadikan sebagai acuan dari pembuatan Laporan Tugas Akhir. Oleh sebab itu dibuatlah sebuah sistem "Deteksi Plagiat Tugas Akhir Dengan Metode Jaccard Similarity Pada Program Studi Teknik Informatika Universitas Islam Sultan Agung". Pada proses pendeteksian dalam sistem melewati proses preprocessing yang terdiri dari tahap case folding dimana sebuah kalimat dalam Laporan Tugas Akhir disetarakan hurufnya menjadi kecil semua dari huruf kapital menjadi huruf kecil. Tahap filtering dimana didalam kalimat spasi dihapus ataupun spesial karakter lain selain huruf. Tahap stemming dimana dari proses filtering diambil kata-kata dasar, kemudian tahap stopword dimana dari hasil stemming dipecah atau dijadikan token. Setelah itu masuk proses Metode Jaccard Similarity yaitu proses penghitungan token dari dokumen uji dan dokumen asli atau pembanding hasilnya adalah nilai similarity dari dokumen uji dan dokumen asli sehingga menampilkan hasil presentase similarity Laporan Tugas Akhir.

***Kata kunci** : Laporan Tugas Akhir, Plagiarisme, Jaccard Similarity.*

1. PENDAHULUAN

Perguruan Tinggi memiliki tanggung jawab yang besar untuk memberikan edukasi dan sosialisasi terkait dengan pencegahan tindakan plagiarisme. Hal ini mengingat perguruan tinggi merupakan salah satu produsen ilmu pengetahuan. Melalui tulisan ini diharapkan anggota civitas academica (mahasiswa, dosen dan staf kependidikan) mampu menghasilkan karya tulis yang berkualitas dan terhindar dari unsur plagiarisme. Menurut Peraturan Menteri Pendidikan RI Nomor 17 Tahun 2010 menyakatakan:

"Plagiat adalah perbuatan sengaja atau tidak sengaja dalam memperoleh atau mencoba memperoleh kredit atau nilai untuk suatu karya ilmiah, dengan mengutip sebagian atau seluruh karya dan atau karya ilmiah pihak lain yang diakui sebagai karya ilmiahnya, tanpa menyatakan sumber secara tepat dan memadai" [1].

Harus diakui, ciptaan karya ilmiah tidak bisa lepas dari produsen utamanya, yaitu kampus dan kalangan intelektual. Salah satu kegiatan atau aktivitas terkait dengan nilai kejujuran dalam kehidupan ilmiah adalah sikap terbuka dan fair dalam penulisan karya tulis ilmiah. Dalam konteks penulisan dan penelitian ini, Karya tulis yang dianggap menjadi *master piece* mahasiswa strata satu kususnya adalah Tugas Akhir Skripsi (TAS). Karya tulis ilmiah inilah yang perlu mendapat tekanan dan perhatian serius. TAS harus dikelola dan diteliti secara serius terkait dengan gejala plagiat yang terjadi di dalamnya [2]

Maka diperlukan suatu metode atau perangkat yang dapat digunakan untuk mendeteksi kemungkinan tingkat kesamaan dalam sebuah Tugas Akhir. Metode yang digunakan untuk menghitung kesamaan adalah metode *Jaccard Similarity* atau *Jaccard Coefficien. Jaccard Similarity* adalah salah satu metode yang dipakai untuk menghitung *Similarity* antara dua objek (*items*) [3]. Seperti halnya *cosine distance* dan *matching coefficient*, secara umum perhitungan metode ini didasarkan pada *vector space similarity measure*. Masing-masing dokumen akan dihitung kata yang sama antara dokumen yang satu dengan dokumen yang lain. Hasil dari perhitungan akan dihasilkan nilai similaritas dokumen. Nilai similaritas dokumen yang tertinggi dapat dianggap bahwa dokumen tersebut paling similar, dengan kata lain memiliki banyak kesamaan.

Penelitin yang dilakukan oleh peniliti Ro'is, M. A dengan judul implementasi metode *Jaccard Similarity* pada aplikasi pencarian lirik lagu, pada penelitian ini menggunkan lirik lagu sebagai kata kunci untuk mendeteksi kemiripan lagu yang sesuai pencarian dengan menentukan nilai terbesar. tahapan-tahapan dalam penelitian tersebut yaitu pengumpulan data lirik lagu, *Tokenizing* atau tahap pemecahan perkata pada tiap–tiap lirik lagu, *Stemming* pengubahan kata asli ke kata dasar, perhitungan menggunakan metode *Jaccard Similarity* untuk mencari kemiripan dari lirik lagu yang dicari dengan menentukan nilai terbesar. untuk penerapan metode *Jaccard Similarity* sebagai penghitung dengan hasil berdasarkan judul lagu, *singer*/penyanyi dan lirik lagu [4].

Dari rujukan yang kedua kasus yang diteliti oleh peneliti Fadelillah dkk, Sistem Rekomendasi Hasil Pencarian Artikel Menggunakan Metode *Jaccard's Coefficient*. pada penelitian ini bertujuan mendesain sistem rekomendasi dengan metode *Jaccard's Coefficient*, dan menguji kinerja temu kembali menggunakan *recall* dan *precision*. Selanjutnya dibuat aplikasi dengan algoritma yang efektif dalam pencarian dokumen dengan metode *Jaccard's Coefficient* pada Portal Garuda IPI (*Indonesian Publication Index*), sehingga pengguna mudah mencari dokumen yang diinginkan. Secara umum sistem yang akan dibangun adalah menggunakan metode *Jaccard Coefficient*, yaitu suatu metode yang bertujuan untuk menentukan bobot dari suatu dokumen dan mengurutkannya berdasarkan nilai persamaannya. Sistem ini akan dibangun menjadi sebuah alat bantu yang berkaitan dengan masalah pencarian artikel jurnal pada Portal Garuda IPI. Sistem ini dapat mempermudah proses pencarian artikel yang diinginkan serta untuk mempermudah pengelompokkan dokumen berdasarkan nilai indeksnya. penerapan metode *jaccard's Coefficient* dalam penelitian tersebut ialah setiap dokumen harus dikorelasikan dengan subyek dengan relasi *many to many*, artinya satu subyek bisa memiliki beberapa dokumen, sebaliknya satu dokumen bisa juga memiliki beberapa subyek. Untuk dapat melakukan pengelompokan dokumen terhadap subyek dapat dilakukan dengan 2 cara, yaitu :

1. Memasukkan setiap dokumen secara langsung kedalam subyek.
2. Memasukkan dokumen secara tidak langsung kedalam suatu subyek dengan menggunakan bantuan term. Untuk dokumen dalam jumlah yang sangat banyak, tidak dilakukan pengelompokan dengan cara memasukkan satu persatu dokumen kedalam subyek, yaitu dengan memperhitungkan frekuensi kemunculan term dalam dokumen tersebut dan jumlah dokumen yang mengandung term tersebut [5].

Pada rujukan ketiga yang dibuat oleh peneliti Samodra dkk. Deteksi Kemiripan Halaman pada Al-Qur'an dengan Menggunakan Algoritma *Rabin Karp* dan *Jaccard Similarity*. pada penelitian ini diperlukan beberapa tahapan penting yaitu, *Hashing* dan *Jaccard Coeffisient*. Selain itu sistem ini menggunakan Algoritma *Rabin Karp* untuk proses *fingerprint*. Algoritma *Rabin-Karp* tidak bertujuan menemukan *string* yang cocok dengan *string* masukan, melainkan menemukan pola (*pattern*) yang sekiranya sesuai dengan teks masukan. Kunci utama penggunaan algoritma *Rabin Karp* adalah perhitungan yang efisien terhadap nilai hash substring. Data input yang dipakai harus melewati peroses *preprocessing* terlebih dahulu. *Preprocessing* digunakan untuk menghilangkan *punctuation*, *case folding*, dan *whitespace insentivity*. Lalu data akan di *Hashing* untuk mendapatkan nilai *ascii* dari data masukkan. *Ascii* merupakan kode standar amerika yang biasanya digunakan untuk pertukaran informasi berupa angka. *Hashing* adalah metode yang menggunakan fungsi hash untuk mengubah suatu jenis data menjadi beberapa bilangan bulat sederhana. Pada tahapan akhir dari sistem ini merupakan proses medapatkan nilai antara 2 dokumen yang dibandingkan, yaitu halaman pertama dan halaman kedua. Hasil dari perbandingan tersebut merupakan nilai *similarity*. Proses pencarian nilai *similarity* menggunakan *Jaccard Coeffisient*. Untuk penerapan metode *Jaccard Similarity Coeffisient* ialah menghitung nilai yang dihasilkan oleh *fingerprinting*, nilai yang dikeluarkan oleh sistem merupakan jumlah nilai 0 dan 1. Dimana nilai 1 mengartikan sebuah ayat tersebut memiliki kemiripan tinggi dengan ayat yang lain. Begitu juga degan nilai 0 yang mengartikan sebuah ayat tersebut memiliki kemiripan rendah dengan ayat yang lain [6].

Pada rujukan keempat yang dibuat oleh peneliti Annisa dkk. Diagnosis Kerusakan Komputer Menggunakan Metode *Similarity Jaccard Coefficient*. dalam penelitian ini ialah membangun sistem berbasis web, Sistem ini tidak untuk menggantikan peran seorang teknisi komputer, tetapi lebih kepada memberikan kemungkinan hasil diagnosis awal beserta solusi berdasarkan gejala-gejala kerusakan yang terjadi pada komputer *user*. dalam penelitian ini menghitung *similarity* antara dua objek A (kasus lama) dan B (kasus baru). Untuk penerapan metode *Similarity Jaccard Coefficient* menghitung *similarity* dengan cara membandingkan hasil diagnosis kasus baru yang dilakukan oleh pakar (teknisi) dengan hasil diagnosis oleh sistem [7].

Kemudian rujukan yang kelima yang dibuat oleh peneliti Rinartha, K. *Simple query suggestion* untuk pencarian artikel menggunakan *Jaccard Similarity*. pada penelitian ini membangun aplikasi berbasis web *Simple query suggestion* dengan menggunakan *Jaccard algorithm* dimulai dari pengguna mengetikkan sesuatu pada bagian *text field*, kemudian system akan mencari artikel dari *database* dengan menghitung nilai *Jaccard Similarity* dan *system* akan menampilkan hasilnya. Hasil dari *query suggestion* mungkin satu atau lebih sesuai dengan data yang terdapat di *database* yang berhubungan dengan *keyword* yang dimasukkan oleh pengguna. *System* akan berhenti melakukan pencarian ketika pengguna berhenti mengetik atau ketika pengguna memilih salah satu *query suggestion* yang diberikan. dalam penerapan *Jaccard Similarity* ialah Bentuk matematis *Jaccard Similarity* diubah menjadi bentuk SQL untuk membaca data dari *database*. Ketika pengguna memasukkan "signal indera pro", *system* akan mencari dan menghitung nilai dari *Jaccard Index* [8].

2. METODE PENELITIAN

Metode penelitian yang digunakan adalah *Jaccard Similarity* dimana nilai kesamaan yang didadapat dan atau dihitung dari nilai persamaan token dari dokumen uji dan dokumen asli atau pembanding. Tahapan-tahapan dalam penelitian ini adalah sebagai berikut ;

2.1. *Pengumpulan Data*

Perancangan sistem deteksi kemiripan tugas akhir dengan metode *Jaccard Similarity*, dalam proses deteksi kemiripan tugas akhir terdapat beberapa tahapan yang terdiri diantaranya;

a. Pengumpula data TA uji dan data TA pembanding
   Pengampulan data TA baik data TA uji maupun data TA asli mengambil data dari *web repository* teknik informatika universitas islam sultan agung.
b. Studi literatur
   Studi literatur yaitu pencarian informasi mengenai sistem deteksi plagiat atau sistem deteksi kesamaan, yang mana informasi yang didapat diperlukan untuk pengumpulan data kebutuhan desain sistem yang akan dibuat.

2.2. *Preprocessing*

a. *Stemming*

Pengembalian kata dasar dari kata kata yang telah mengalami imbuhan, sisipan dan awalan. Merupakan proses pemetaan dan penguraian berbagai bentuk (*variants*) dari suatu kata menjadi kata dasarnya (*stem*). Proses ini disebut juga sebagai *conflation*. Proses *stemming* secara luas sudah digunakan di dalam *information retrieval* (pencarian informasi) untuk meningkatkan kualtas informasi yang didapatkan [4].

b. *Tokenizing*

Tokenizing adalah proses memisahkan deretan kata di dalam kalimat, paragraf atau halaman menjadi token atau potongan kata tunggal atau *termmed word* yang berdiri sendiri.[9] atau dengan kata lain yaitu proses dimana isi dari dokumen yang berisikan kumpulan kalimat dipecah menjadi kata per kata. Disini tanda baca dan spesial karakter lainnya dihilangkan dari kumpulan kata yang ada [10].

c. *Stopword Removal*

Stopword removal dokumen, yaitu kata-kata yang sering muncul dalam dokumen namun artinya tidak deskriptif dan tidak memiliki keterkaitan dengan tema tertentu. Pada bahasa Indonesia, *stopword* disebut juga sebagai kata yang tidak penting, misalnya “di”, “oleh”, “pada”, “sebuah”, “karena” dan lain sebagainya [11].

2.3. *Jaccard Similarity*

Jaccard Algorithm atau dikenal dengan *Jaccard Coefficient* dan atau *Jaccard Similarity* adalah salah satu metode yang dipakai untuk menghitung *similarity* antara dua objek (*items*) [12]. Masing-masing dokumen akan dihitung kata yang sama antara dokumen yang satu dengan dokumen yang lain. Hasil dari perhitungan akan dihasilkan nilai similaritas dokumen. Nilai similaritas dokumen yang tertinggi dapat dianggap bahwa dokumen tersebut paling similar, atau memiliki banyak kesamaan [13]. Rumus *Jaccard Coefficient* atau *Jaccard Similarity*

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A|+|B|-|A \cap B|} \quad (1)$$

Pada perhitungan *Jaccard*, nilai |A∩B| merupakan jumlah *fingerprint* yang sama antara dokumen A dengan dokumen B. Untuk dapat mengetahui nilai dokumen A sama dengan dokumen B dilakukan penyimpanan setiap nilai pada dokumen A kemudian dibandingkan dengan setiap nilai pada dokumen B, apabila sesuai maka nilai irisan ditambahkan dan disimpan. Bila semua nilai sudah dibandingkan maka proses berhenti dan nilai |A∩B| sudah diketahui. Pada proses |AUB| bisa dihitung dengan mencari nilai jumlah *fingerprint* pada dokumen A kemudian ditambah dengan jumlah *fingerprint* pada dokumen B dan dikurangi oleh nilai |A∩B| [14].

Misalkan terdapat dua buah dokumen yang dilakukan pembobotan dan pengindeksan. Isi dari kedua dokumen yang digunakan tersebut dapat dirincikan sebagai berikut:

Dokumen 1 = sarjana teknik informatika unissula
Dokumen 2 = sarjana teknik sipil unissula

Kemudian memisahkan dua dokumen di atas menjadi *array*; (1)sarjana, (2)teknik, (3)informatika, (4)sipil , (5)unissula. Berarti memiliki dua set yang berbeda yaitu Dokumen 1 dan Dokumen 2.

Dokumen1 = A,
Dokumen2 = B
A= {1,2,3,5} dan B= {1,2,4,5} .

Kemudian mencari *Union* dari kedua dokumen tersebut. *Union* adalah jumlah kata secara keseluruhan dari dua dokumen yang sedang dihitung. Dari *array* diatas bisa lihat bahwa jumlah kata secara keseluruhan adalah 5 kata. *Union* dari Dokumen 1 dan 2 adalah sebagai berikut :

AUB = {1,2,3,4,5}
Keterangan :
U = *Union*
A = Dokumen 1

B = Dokumen 2

Setelah berhasil mendapatkan hasil *Union*, selanjutnya adalah mencari *Intersection* diantara dua dokumen tersebut. *Intersection* adalah jumlah kata yang sama dari dua dokumen yang sedang dihitung. Jika dilihat dari Dokumen A dan Dokumen B, ada beberapa kata yang sama dari kedua dokumen tersebut, antara lain : [1]sarjana, [2]teknik , [5]unissula. Intersection dari Dokumen A dan B adalah :

A ∩ B ={1,2,5}

Keterangan :

∩ = *Intersection*

A = Dokumen 1

B = Dokumen 2

Union = 1,2,3,4,5 = 5 Kata

Intersection =1,2,5 = 3 Kata

Langkah selanjutnya adalah menghitung kemiripan dari kedua dokumen tersebut dengan rumus sebagai berikut ;

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{3}{5} = 0{,}6$$

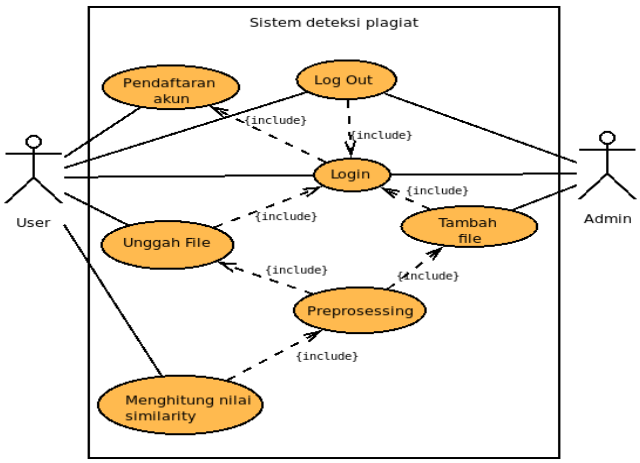
Berdasarkan nilai persamaan yang diperoleh maka dapat ditetapkan bahwa nilai kemiripan dari dokumen 1 dan dokumen 2 adalah 0,6 [5].

1. Algoritma *Jaccard Similarity*

Algoritma *Jaccard Similarity* digunakan untuk menghitung nilai similaritas dari dokumen uji yang melakukan pembandingan dengan dokumen pembanding, algoritma *Jaccard Similarity* menerima nilai dari dokumen uji dan nilai dari dokumen pembanding, setelah mendapatkan nilai kemudian melakukan proses perhitungan, dari proses perhitungan nilai dari dokumen uji dan dokumen pembanding tersebut kemudian didapat nilai *similarity* dari dokumen uji.

2. Diagram *Use Case*

Diagram *Use case* analisa kebutuhan sistem merupakan *use case* diagram dari sistem deteksi plagiat tugas akhir, dalam *use case* tersebut terdapat 2 (dua) aktor yang terlibat yaitu *user* dan admin, aktor *user* dapat sebagai mahasiswa, dosen dan pengguna lain, sedangkan aktor admin adalah *steckholder* yang dipercaya dalam mengelola sistem. aktor *user* dapat melakukan pendaftaran akun, masuk dan keluar dengan akun dari sistem, unggah *file* dokumen uji dan melihat nilai *similarity* dokumen yang diujikan. aktor admin dapat masuk dan keluar dengan akun dari sistem, melakukan *Create Read Update Delete* (CRUD) dokumen pembanding. Gambar 1 adalah use case diagram deteksi plagiat pada penelitian ini.

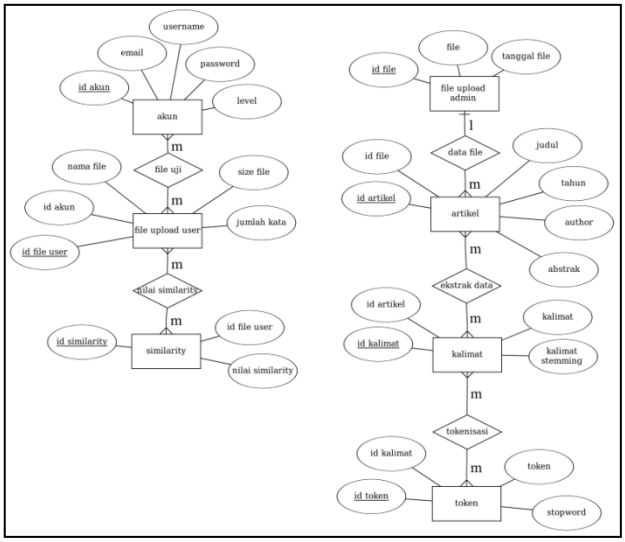


Gambar 1 *Use Case diagram* sistem deteksi plagiat

3. *Entity Relationship* Diagram (ERD)

Entity Relationship Diagram kebutuhan sistem deteksi plagiat tugas akhir dengan menggunakan metode *Jaccard Similarity* terdapat beberpa Entitas diantaranya, Entitas akun yaitu entitas yang menyimpan data akun aktor baik data akun aktor admin maupun data aktor *user*, Entitas *file* uji yaitu entitas inputan *file* dimana entitas ini hanya dapat di akses oleh aktor *user*, Entitas nilai *similarity* dimana entitas ini menampung data nilai *similarity* atau nilai kesamaan setelah melalui proses deteksi kesamaan dokumen, Entitas *file upload* admin dimana entitas ini sebagai input *file* dokumen asli entitas ini hanya dapat di akses oleh aktor admin, Entitas artikel dimana entitas ini berfungsi

menyimpan bagian data *file* yang di-*upload*, Entitas kalimat dimana entitas ini menyimpan data hasil proses *stemming* dari hasil *file upload* admin, dan kemudian terdapat Entitas token dimana entitas ini menyimpan data token dari kalimat yang telah ter-*stemming*. Untuk lebih jelas tentang *Entity Relationship* Diagram kebutuhan sistem dapat dilihat pada Gambar 2 *Entity Relationship* Diagram



Gambar 2 *Entity Relationship Diagram* sistem deteksi plagiat

4. Perancangan Sistem

Sistem yang akan dikembangkan penulis merupakan sebuah sistem pendeteksi plagiat tugas akhir dengan menggunakan metode *Jaccard Similarity* , tugas akhir yang dibuat sebagai sampel yaitu terkhusus pada tugas akhir mahasiswa/i teknik informatika unissula. Dalam prosesnya, sistem menerima input *file* data yang dideteksi yang ber-ekstensi (pdf), kemudian sistem menghasilkan nilai kemiripan dari *file* yang diinputkan, dan dari nilai yang dihasilkan terdeteksi apakah *file* yang diujikan termasuk dalam kategori plagiat atau tidak.

3. HASIL DAN PEMBAHASAN

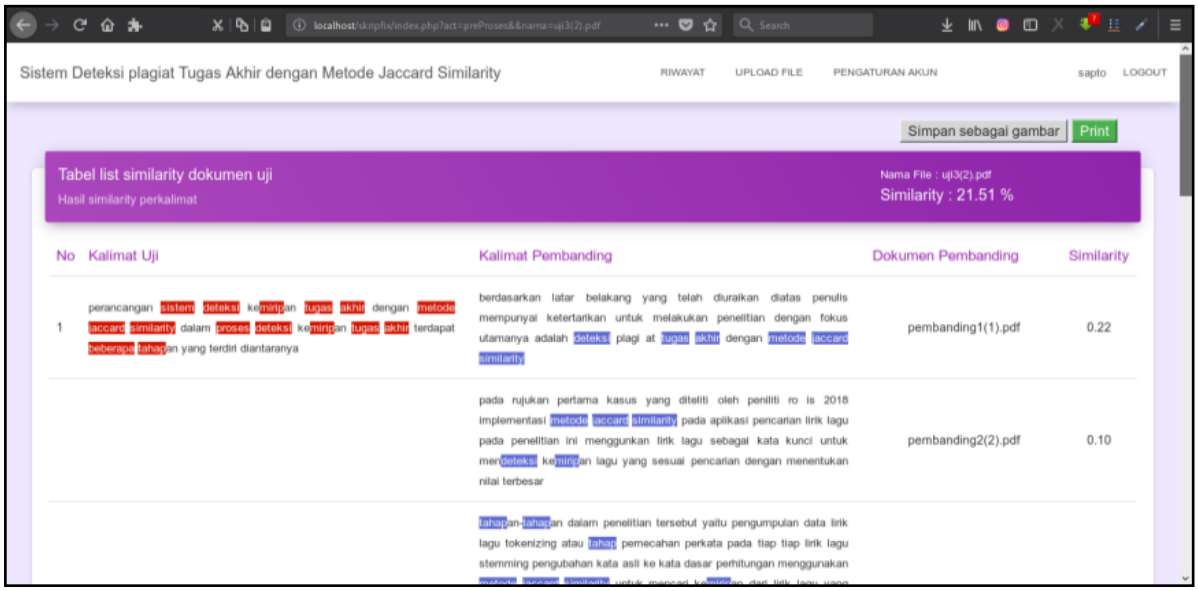
3.1. Implementasi Rumus *Jaccard Similarity*

Berikut adalah potongan program dari rumus *jaccard similarity* yang peneliti terapkan pada sistem.

```
//proses pembandingan kata ter-stopword yang sama
//$fileDataTokenInput = token ter-stopword dukumen input
//$tokenKalimatDB = token ter-stopword database
$outputData_array_intersections = array_intersect(explode(' ', $fileDataTokenInput),explode(' ',
$tokenKalimatDB));
 //size of intersection
$intersection__sizeof=
sizeof(array_unique($outputData_array_intersections));
//token db stopword
$kalimatDatabasesDB = explode(' ',$tokenKalimatDB);
//token dokumen input stopword
$kalimatInputXIN = explode(' ',$fileDataTokenInput);
//size of union
$union_sizeof = sizeof(array_unique(array_merge($kalimatInputXIN,
$kalimatDatabasesDB)));
//jaccard similarity rumus
$jaccard = $intersection_sizeof / $union_sizeof;
```

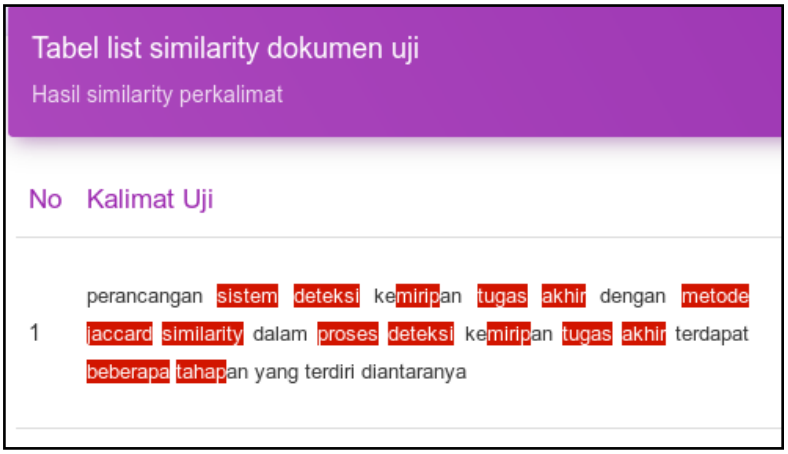
3.2. Implementasi *User Interface*

Halaman *user* proses deteksi *similarity* Gambar 4.1 Halaman *user* proses deteksi *similarity* merupakan tampilan proses pendeteksian *similarity* atau kesamaan kalimat dokumen uji dengan dokumen pembanding.



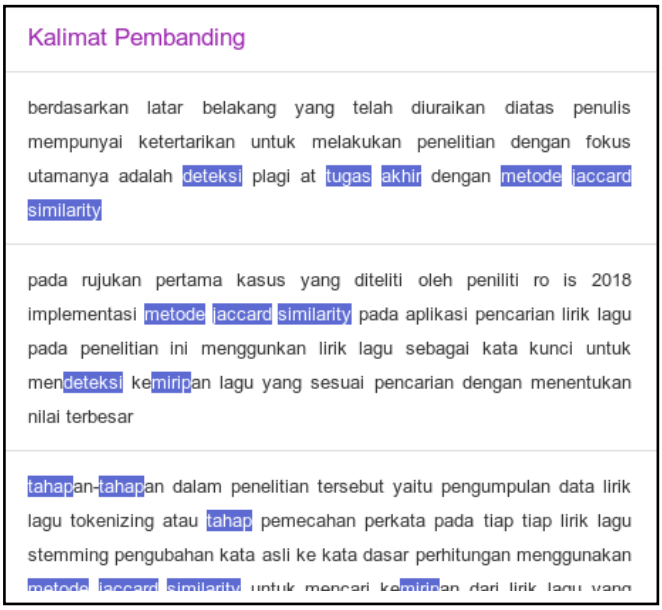
Gambar 3 Halaman *user* proses deteksi similarity

Tampilan pada Gambar 3, menunjukkan tampilan kalimat uji yang ter-*highlight* warna merah dan kalimat pembanding yang ter-*highlight* warna biru, serta terdapat nama file pembanding dan hasil *similarity* dari perbandingan.



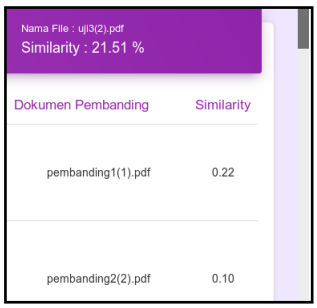
Gambar 4 Halaman user proses deteksi similarity (2*)*

Tampilan pada Gambar 4, menunjukkan kalimat uji yang ter-*highlight* warna merah menandakan kata tersebut terdeteksi kesamaan dengan kalimat pada dokumen pembanding.



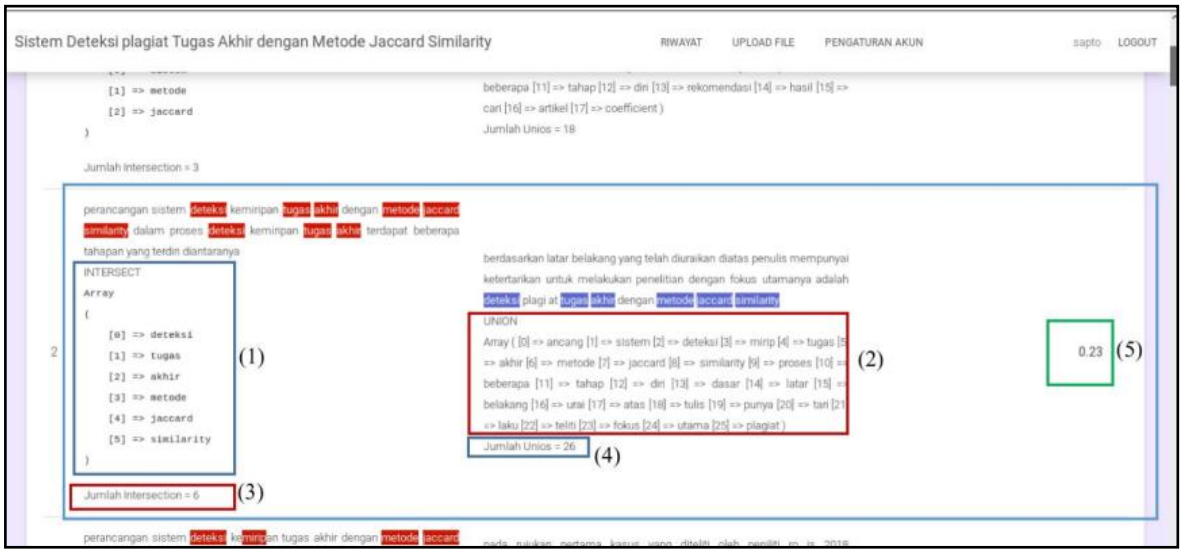
Gambar 5 Halaman user proses deteksi similarity (3)

Tampilan pada Gambar 5, menunjukkan kalimat pembanding yang ter-*highlight* warna biru menandakan kata tersebut terdeteksi kesamaan dengan kalimat pada dokumen uji.



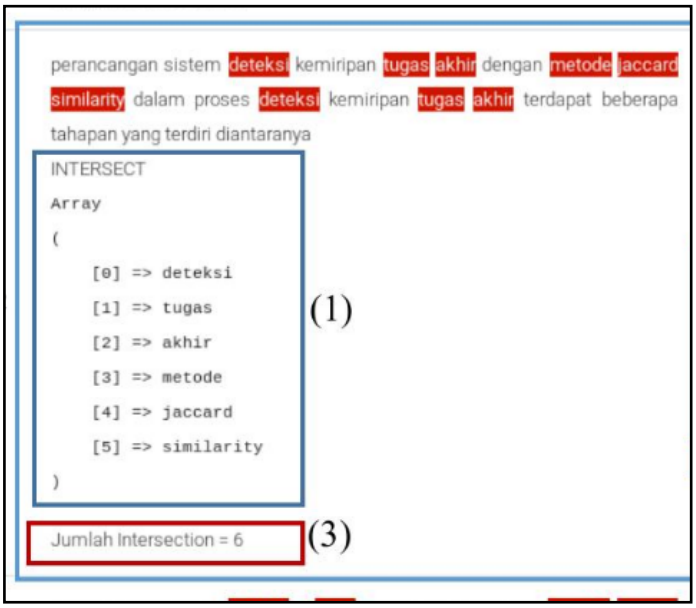
Gambar 6 Halaman user proses deteksi similarity (4)

Tampilan pada Gambar 6, menunjukkan hasil perkalimat yang terdeteksi kesamaan dengan kalimat pada dokumen uji. Proses perhitungan deteksi plagiat tugas akhir pada Gambar 7, Ketika *user upload file* uji, sistem memproses dengan cara mengekstrak dokumen PDF dalam bentuk teks kemudian teks dibuat menjadi beberapa kalimat, pengambilan kalimat dalam teks yang terekstrak yaitu dengan cara pembatasan teks yang diakhiri titik yang tersambung spasi (.) dan atau titik dan disambung *Enter* (.↵), setelah kalimat terbuat, maka kalimat di *stemming* setelah kalimat ter-*stemming* kemudiansistem melakukan *stopword* kalimat, setelah itu kalimat dipecah menjadi token-token, proses *stemming*, *stopword* dan tokenisasi peneliti menggunakan library sastrawi, selanjutnya token-token inilah yang di bandingkan dengan token yang terdapat dalam *database*, setelah token dibandingkan dan mendapati token yang sama, sistem mencari kalimat asli atau pembanding yang tokennya terdeteksi sama dengan token kalimat uji, kemudian sistem menghitung jumlah token yang sama, token yang sama diidentifikasikan sebagi *Intersection*, setelah didapat nilai dari perhitungan token yang sama, sistem juga menghitung token dari kalimat uji dan kalimat pembanding yang terbandingkan dari token yang sama. token yang tidak terdeteksi kesamaannya dalam satu kalimat perbandingan diidentifikasikan sebagi *Union.* Kemudian nilai dari *intersection* dan *union* dihitung oleh sistem hingga didapati nilai *similarity* dari kalimat tersebut, pada Gambar 4.5 nomor (2), sistem hanya menampilkan kalimat pembanding yang tokennya paling banyak atau paling besar terdeteksi kesamaannya pada setiap kalimat dokumen pembanding, setelah didapatkan yang paling besar pada setiap kalimat, sistem menseleksi kembali dengan hanya menampilkan kalimat pembanding yang terdapat minimal 5 (lima) token yang terdeteksi sama, sehingga meminimalkan tampilan pada sistem, walapun kalimat yang tampil dalam sistem yang terbesar kesamaannya, akan tetapi perhitungan nilai tetap berdasarkan banyaknya kalimat yang terdeteksi kesamaannya. Untuk lebih detail dapat dilihat pada Gambar 4.5 Contoh perhitungan sistem deteksi *similarity* terdapat *array* yang sengaja peneliti tampilkan untuk memudahkan dalam perhitungan manual deteksi kesamaan.



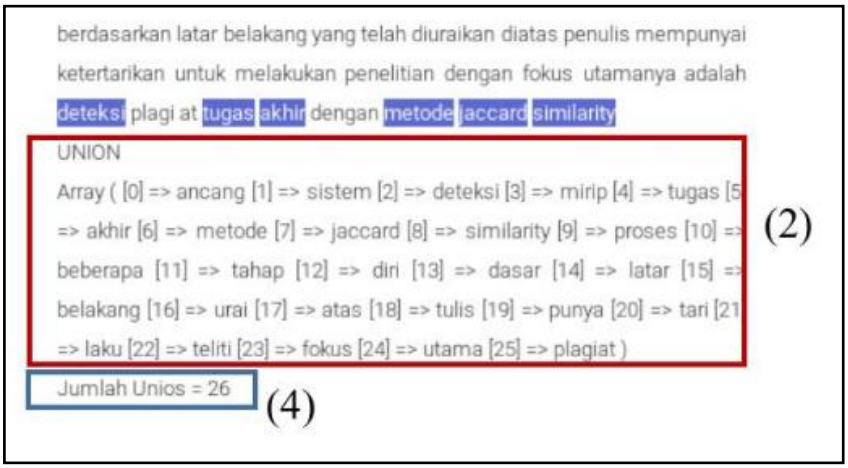
Gambar 7 Contoh perhitungan sistem deteksi similarity

Pada Gambar 7, terdapat dua tampilan yaitu tampilan kalimat pembanding berserta token pada nomor (1) dan jumlah token dalam kalimat uji pada nomor (3) dan kalimat pembanding pada nomor (2) dan jumlah token dalam kalimat pembanding pada nomor (4) , kemudian hasil dari perbandingan pada nomor (5).



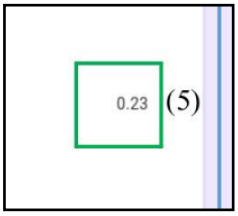
Gambar 8 Contoh perhitungan sistem deteksi similarity (2)

Tampilan konten pada Gambar 8, menunjukkan kalimat uji yang ter-*highlight* warna merah menandakan kalimat tersebut terdeteksi kesamaan dengan kalimat pada dokumen pembanding, nomor (1) adalah token kalimat uji yang terdeteksi kesamaanya dengan kalimat pembanding yang diidentifikasikan sebagai *intersection*, dan nomor (3) adalah jumlah token yang terdeteksi kesamaanya .



Gambar 9 Contoh perhitungan sistem deteksi similarity (3)

Tampilan konten pada Gambar 9, menunjukkan kalimat pembanding yang ter-*highlight* warna biru menandakan kata tersebut terdeteksi kesamaan dengan kalimat pada dokumen uji, nomor (2) adalah penggabungan token kalimat pembanding dan token kalimat uji yang diidentifikasikan sebagai *union*, dan nomor (4) adalah jumlah token yang terdeteksi .



Gambar 10 Contoh perhitungan sistem deteksi similarity (4)

Pada nomor 1 (satu) adalah *array* token perbandingan yang terdeteksi sama dari kalimat uji dan kalimat pembanding, tokenya yaitu "deteksi", "tugas", "akhir", "metode", "*jaccard*", "*similarity*", total token yang terdeteksi kesamaannya adalah 6 (enam), yang diindentifikasikan sebagai *Intersection,* kemudian pada nomor 2 (dua) adalah *array* token penggabungan dari dokumen uji dan dokumen pembanding yang tidak terdeteksi kesamaannya dalam satu kalimat perbandingan, keterangan token dapat dilihap pada Gambar 4.6 dan Gambar 4.7, total token penggabungan pada nomor 4 (empat) adalah 26 (dua puluh enam), token penggabungan yang diidentifikasikan sebagai *Union.* Setelah diketahui nilai dari *intersection* dan *union* dari kalimat yang diperbandingkan, kemudian dimasukkan dalam rumus *Jaccard Similarity*. Keterangan sebagai berikut;

A = Kalimat uji
B = kalimat pembanding
Rumus *Jaccard Similarity*
|A∩B| = *Intersection*
|AUB| = *Union*

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{6}{26} = 0{,}23$$

Nilai 13 (tiga belas) adalah jumlah kata *uniq* dalam kalimat uji, dan nilai 19 (sembilan belas) adalah jumlah kata *uniq* dalam kalimat pembanding. Maka didapat hasil 0,23 seperti pada nomor 5 (lima), hasil 0,23 adalah hasil pembulatan dari hasil asli yaitu 0,2307692308. hasil 0,23 adalah hasil perkalimat perbandingan, untuk hasil *similarity* keseluruhan atau global yaitu penjumlahan dari nilai perbandingan perkalimat dibagi total jumlah kalimat uji yang terbanding atau yang terdeteksi kesamaannya kemudian dikalikan seratus (100) untuk merubah kedalam nominal persen., maka jika nilai 0,23 dijadikan persen hasilnya 23% (dua puluh tiga persen).

4. KESIMPULAN

Setelah melakukan pembahasan dapat ditarik sebuat kesimpulan penerapan Algoritma *Jaccard Similarity* dalam sistem deteksi kemiripan tugas akhir dapat menghitung nilai perbandingan dari dokumen uji dengan dokumen asli, nilai perbandingan didapat dari jumlah besaran token yang sama dan jumlah besaran token yang berbeda dalam satu kalimat perbandingan, penggabunagn token yang sama dalam satu kalimat perbandingan di sebut sebagai *intersection* sedangkan penggabunag token yang berbeda dalam satu kalimat perbandingan disebut sebagi *Union*, nilai

Intersection dan nilai *Union* inilah yang dihitung sebagai nilai perbandingan dan kemudian didapat nilai presentase nilai kemiripan atau nilai *similarity*, setiap dokumen yang dibandingkan menampilkan nilai presentase kemiripannya yang berbeda-beda.

DAFTAR PUSTAKA

[1] P. Istiana and Purwoko, "Panduan Anti Plagiarism," *Perpust. Univ. Gajah Mada*, pp. 1–14, 2016.

[2] M. A. A. Widiantoko, "Plagiat pada Tugas Akhir Skripsi Mahasiswa Fakultas Ilmu Sosial Universitas Negri Yogyakarta," Universitas Negeri Yogyakarta, 2014.

[3] O. Nurdiana, J. Jumadi, and D. Nursantika, "Perbandingan Metode Cosine Similarity Dengan Metode Jaccard Similarity Pada Aplikasi Pencarian Terjemah Al-Qur'an Dalam Bahasa Indonesia," *J. Online Inform.*, vol. 1, p. 59, 2016.

[4] M. A. Ro'is, "Implemenyasi Metode Jaccard Similarity Pada Aplikasi Pencarian Lirik Lagu," *Artik. Skripsi Univ. Nusant. PGRI Kediri*, vol. 6, 2018.

[5] M. Fadelillah, I. Much, I. Subroto, and D. Kurniadi, "Sistem Rekomendasi Hasil Pencarian Artikel Menggunakan Metode Jaccard ' s Coefficient," *J. Elektro dan Inform. Unissula*, vol. 2, no. 1, pp. 1–14, 2017.

[6] W. E. Samodra and M. A. Bijaksana, "Deteksi Kemiripan Halaman pada Al- Qur ' an dengan Menggunakan Algoritma Rabin Karp dan Jaccard Similarity," *e-Proceeding Eng.*, vol. 5, no. 3, pp. 7658–7664, 2018.

[7] A. Annisa, T. Tursina, and H. S. Pratiwi, "Diagnosis Kerusakan Komputer Menggunakan Metode Similarity Jaccard Coefficient," *J. Sist. dan Teknol. Inf.*, vol. 5, no. 2, pp. 104–108, 2017.

[8] K. Rinartha, "Simple Query Suggestion Untuk Pencarian Artikel Menggunakan Jaccard Similarity," *J. Ilm. Rekayasa dan Manaj. Sist. Inf.*, vol. 3, no. 1, pp. 30–34, 2017.

[9] L. Robinson, "Implementasi Metode Generalized Vector Space Model Pada Aplikasi Information Retrieval untuk Pencarian Informasi Pada Kumpulan Dokumen Teknik Elektro Di UPT BPI LIPI," *J. Ilm. Komput. dan Inform. (KOMPUTA)*, 2014.

[10] P. F. Ariyani, A. Rahmala, and N. Juliasari, "Implementasi Metode Stemming Tala Dan Fungsi Jaccard Pada Aplikasi Katalog Perpustakaan," *Semin. Nas. Inov. dan Apl. Teknol. di Ind. 2019*, pp. 128–133, 2019.

[11] A. . Fallis, "Implementasi Metode Tersm Frequency-Inverse Document Frequency (TF-IDF) dan Maximum Marginal Relevance untuk Monitoring Diskusi Online," *J. Chem. Inf. Model.*, vol. 53, no. UIN SUSKA RIAU, pp. 1689–1699, 2015.

[12] O. Nurdiana, J. Jumadi, and D. Nursantika, "Perbandingan Metode Cosine Similarity Dengan Metode Jaccard Similarity Pada Aplikasi Pencarian Terjemah Al-Qur'an Dalam Bahasa Indonesia," *J. Online Inform.*, vol. 1, no. 1, p. 59, 2016.

[13] S. Sunardi, A. Yudhana, and I. A. Mukaromah, "Implementasi Deteksi Plagiarisme Menggunakan Metode N-Gram Dan Jaccard Similarity Terhadap Algoritma Winnowing," *Transmisi*, vol. 20, no. 3, p. 105, 2018.

[14] J. Evan Harya Chandra, V. Christiani M, and D. S.Naga, "Plagiarisme Abstrak Menggunakan Algoritma Winnowing dan Synsets," *J. Ilmu Komput. dan Sist. Inf.*, no. Universitas Tarumanagara Jakarta, pp. 121–129, 2016.