Deteksi Kemiripan Dokumen Publikasi Skripsi Mahasiswa Menggunakan Algoritma Modifikasi Cosine Similarity

Yisti Vita Via[1], Retno Mumpuni[2]

[1,2] Program Studi Teknik Infomatika Universitas Pembangunan Nasional "Veteran" Jawa Timur
[1]yistivia.if@upnjatim.ac.id
[3]retno_mumpuni.if@upnjatim.ac.id

***Abstrak*— Pada perguruan tinggi, kasus plagiarisme menjadi satu hal yang paling disoroti khususnya pada kasus publikasi karya ilmiah. Di lingkungan perguruan tinggi, pencegahan terhadap kasus plagiarisme sudah dilakukan dengan melampirkan halaman surat pernyataan anti plagiarisme pada laporan karya ilmiah. Namun tindakan ini belum cukup menjamin tingkat originalitas suatu karya. Cara lain untuk mencegah tindakan plagiarisme adalah dengan mendeteksi adanya kemiripan dokumen sebelum dokumen tersebut diakui pihak kedua. Penelitian ini merancang dan membangun sistem untuk deteksi kemiripan isi teks dokumen pada jurnal publikasi skripsi karya mahasiswa dengan menggunakan metode Levenshtein Distance dan Cosine Similarity. Aplikasi ini berbasis website dengan keluaran sistem berupa hasil prosentase kemiripan dokumen yang diunggah, yang mana akan ditampilkan dalam sebuah file lampiran pernyataan. File lampiran ini yang nantinya dapat dicetak dan digunakan sebagai surat jaminan originalitas karya ilmiah menggantikan surat pernyataan anti plagiarisme yang digunakan sebelumnya.**

***Kata Kunci*— Plagiarisme, Levenshtein Distanee, Cosine Similarity,Text Mining.**

I. PENDAHULUAN

Pada era modern seperti saat ini, pemanfaatan teknologi digital telah menjadi kebutuhan primer. Salah satu komponen yang ada di dalam dunia digital adalah dokumen yang berupa file teks. Dokumen atau file dalam bentuk digital memiliki banyak kelebihan salah satunya kemudahan dalam hal penyimpanan dan pencarian. Selain kemudahan yang positif, ternyata tidak menutup kemungkinan adanya kemudahan untuk menjiplak atau menyalin dokumen yang dalam hal ini tentu sangat bersifat negatif dan merugikan pihak tertentu.

Praktek penjiplakan atau istilahnya plagiarisme sering terjadi di dunia akademik, mulai dari tingkat sekolah dasar hingga perguruan tinggi. Contoh sederhana saja adalah praktek salin-tempel terhadap tugas-tugas yang diberikan dari sekolah atau institusi. Pada perguruan tinggi, kasus plagiarisme menjadi satu hal yang paling disoroti. Terutama kasus plagiarisme pada publikasi karya ilmiah. Di lingkungan perguruan tinggi, pencegahan terhadap kasus plagiarisme sudah dilakukan. Salah satunya dengan melampirkan halaman surat pernyataan anti plagiarisme pada laporan karya ilmiahnya mahasiswa maupun dosen. Namun tindakan ini belum cukup menjamin tingkat originalitas suatu karya.

Cara lain untuk mencegah tindakan plagiarisme yaitu dengan mendeteksi adanya kemiripan antar dokumen sebelum dokumen tersebut diakui pihak kedua. Memang cara paling sederhana mendeteksi dokumen yang mirip adalah dengan membandingkan secara manual beberapa dokumen, namun cara tersebut tentu sangat tidak efektif. Oleh karena itu seringkali lembaga atau institusi menggunakan perangkat lunak atau aplikasi yang bisa mendeteksi plagiarisme, namun harus keberatan dengan berlangganan dan membayar dengan biaya yang sangat mahal. Untuk memberikan solusi dari beberapa kendala tersebut, maka pada penelitian ini diajukan pembuatan suatu sistem deteksi kemiripan isi teks dokumen pada database jurnal dengan menggunakan metode Levenshtein Distance dan Cosine Similarity.

Salah satu metode yang tepat dalam melakukan deteksi kemiripan isi teks dokumen adalah dengan menggunakan metode Levenshtein Distance. Levenshtein Distance memperhatikan tiga operasi dalam menentukan jarak diff, yaitu (1) operasi penyisipan (insertion), (2) operasi penghapusan (deletion), (3) operasi penggatian (subtitution), sebuah huruf yang berdekatan. Sedangkan metode Cosine Similarity merupakan metode yang digunakan untuk menghitung tingkat kesamaan antar dua dokumen.

Penelitian ini bertujuan untuk membuat suatu sistem yang mampu mendeteksi tingkat kemiripan isi teks dari dokumen jurnal publikasi skripsi karya mahasiswa dengan menggunakan metode hibrida Levenshtein Distance dan Cosine Similarity.

Dengan adanya sistem ini diharapkan pencegahan terhadap permasalahan plagiarisme dapat diatasi. Selain itu mampu mengurangi beban biaya institusi karena tidak lagi berlangganan aplikasi cek plagiat berbayar.

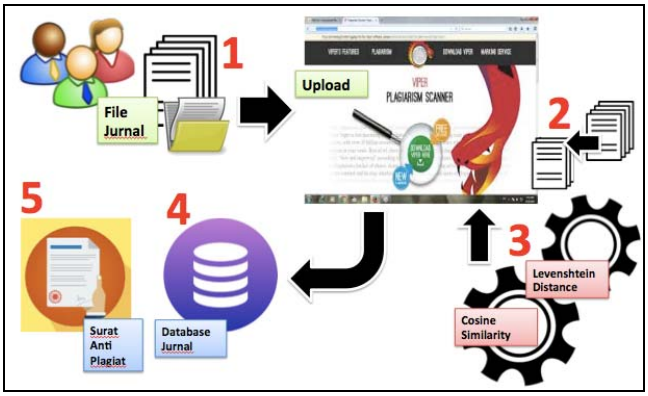
II. METODOLOGI

Pada bab ini bagian tahapan metodologi akan dijelaskan secara rinci.

A. *Desain Sistem*

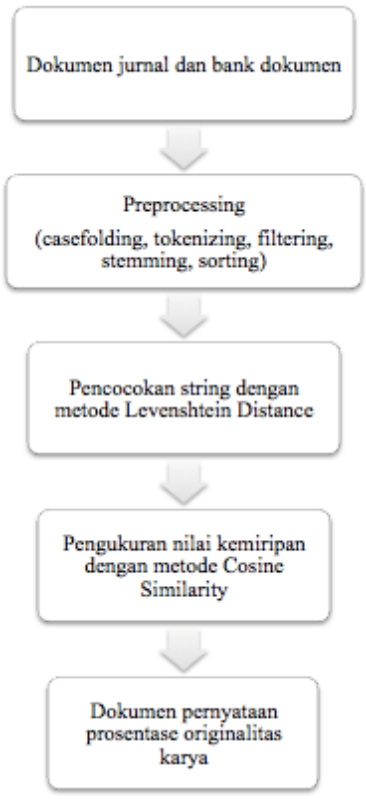
Keseluruhan sistem pada aplikasi diilustrasikan pada Gambar 1. Alur kerja sistem dijelaskan terurut mulai dari input, proses, kemudian output. Input atau masukan dari sistem adalah berupa file dokumen jurnal publikasi mahasiswa yang akan yudisium dengan format dokumen pdf. Pada blok diagram, dokumen masukan digambarkan dengan gambar folder (nomor 1). Dokumen yang diupload ini akan dibandingkan kemiripannya dengan dokumen-dokumen jurnal publikasi mahasiswa lainnya yang sebelumnya sudah dipreprocessing dan tersimpan di database dokumen. Database dokumen ini digambarkan dengan gambar tumpukan dokumen (nomor 2) pada blok diagram. Proses deteksi kemiripan ini

menggunakan dua algoritma yaitu Levenshtein Distance dan Cosine similarity. Dua algoritma ini digambarkan di blok diagram dengan dua roda gerigi (nomor 3).



Gbr. 1 Desain Sistem

Setelah dibandingkan dan dideteksi kemiripannya dengan database dokumen, hasil preprocessing dari dokumen masukan ini akan disimpan oleh database dokumen. Proses ini diberi nomor 4 pada blok diagram. selanjutnya akan memberikan output atau luaran berupa file dokumen yang berisi prosentase hasil kemiripan dan pernyataan plagiat atau bukan diukur dari nilai batas ambang 30 persen. File dokumen luaran ini digambarkan dengan nomor 5 pada blok diagram.



Gbr. 2 Alur Tahapan Sistem

Secara flowchart input, proses, hingga output sistem dijelaskan pada Gambar 2. Masukan sistem berupa dokumen jurnal publikasi mahasiswa skripsi dan kumpulan hasil preprocessing dari dokumen jurnal publikasi yang sudah tersimpan di database. Kemudian proses sistem terdiri dari tiga tahapan yaitu tahap preprocessing, tahap pencocokan string dengan Levenshtein Distance, dan tahap pengukuran kemiripan dengan Cosine Similarity. Sedangkan untuk luaran sistem yaitu penyimpanan hasil preprocessing dokumen masukan ke dalam database dan file dokumen pernyataan prosentase hasil kemiripan dokumen.

B. *Levenshtein Distance*

Levenshtein Distance merupakan algoritma yang digunakan untuk mengukur kemiripan antara 2 string yaitu disebut string awal (s) dengan string target (t). Semisal dijelaskan pada kalimat berikut :

Jika (s) = “asal”, dan (t) = “asal” ==> maka nilai Levenshtein(s,t) = 0, hal ini dikarenakan antara kedua string tersebut tidak memiliki perbedaan atau dikatakan sama.

Jika (s) =”asal”, dan (t) = “awal” ==> maka nilai Levenstein(s,t) =1 , hal ini dikarenakan adanya perbedaan antara kedua string tersebut yaitu perbedaan di huruf yang kedua yakni ‘s’ dan ‘w’ (Ijalandhika, 2015).

Levenshtein Distance pertama kali dikenalkan pada tahun 1965 oleh Vladimir Levenshtein. Dalam melakukan perhitungan edit distance digunakan matriks untuk menghitung jumlah perbedaan string dari dua string.

Ada 3 jenis operasi utama pada algoritma ini yaitu :

1. Operasi Pengubahan Karakter

Operasi pengubahan karakter adalah operasi untuk menukar sebuah karakter dengan karakter lain. Contohnya string “yamg” diubah menjadi “yang”, ini berarti karakter “m” diganti dengan huruf “n”.

2. Operasi Penambahan Karakter

Operasi penambahan karakter adalah operasi untuk menambahkan karakter ke dalam suatu string. Contohnya string “kepad” ditambah karakternya menjadi “kepada”, ini berarti karakter “a” ditambahkan pada string "kepad" di akhir string. Penambahan karakter bisa dilakukan di awal, di akhir, maupun disisipkan di tengah string.

3. Operasi Penghapusan Karakter

Operasi penghapusan karakter adalah operasi yang dilakukan untuk menghilangkan karakter dari suatu string. Contohnya menghilangkan karakter "r" pada string “barur” sehingga menjadi string “baru” (Andriyani, 2010).

Tiga tahap algoritma ini dilakukan mulai dari pojok kiri atas suatu array dua dimensi yang telah berisi sejumlah karakter sring asal dan string target dan diberikan nilai harga atau cost. Nilai harga pada ujung kanan bawah menjadi nilai edit-distance yang merupakan nilai dari jumlah perbedaan dua string.

C. *Cosine Similarity*

Cosine similarity merupakan algoritma yang digunakan untuk menghitung tingkat kesamaan atau similarity antara dua buah obyek. Algoritma ini biasanya digunakan untuk tujuan klastering dokumen. Fungsi Cosine Similarity menggunakan rumus berikut ini:

$$Similarity(X,Y) = \frac{|X \cap Y|}{|X|^{1/2}|Y|^{1/2}}$$

Dimana:
|X∩Y| adalah jumlah term yang terdapat pada dokumen X dan dokumen Y
|X| adalah jumlah term yang terdapat pada dokumen X
|Y| adalah jumlah term yang terdapat pada dokumen Y

Beberapa tahapan dalam algoritma Cosine Similarity adalah sebagai berikut:

1. Melakukan preprocessing terhadap semua dokumen pembanding.

Langkah 1a: melakukan tokenisasi, stopwords removal dan stemming.
Langkah 1b. menghitung bobot untuk setiap term pada dokumen.

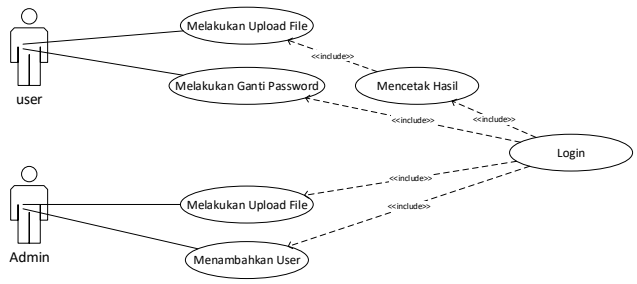
2. Menghitung kemiripan vektor [dokumen] query dengan setiap dokumen yang ada. Penghitungan kemiripan ini menggunakan algoritma Cosine Similarity.

Langkah 2a: menghitung hasil perkalian skalar antara vektor query dengan dokumen lain. Hasil dari perkalian setiap dokumen dengan vektor query nanti akan dijumlahkan.
Langkah 2b: menghitung panjang setiap dokumen, termasuk vektor query dengan cara kuadratkan bobot setiap term untuk setiap dokumen, kemudian lakukan penjumlahan nilai kuadratnya dan langkah terakhir adalah menarik akar dari hasil tersebut.
Langkah 2c: mengimplementasikan perhitungan rumus Cosine Similarity yaitu menghitung kemiripan vektor query dengan D1, D2 dan seterusnya sampai dengan D6.

3. Langkah 3: Mengurutkan hasil perhitungan kemiripan dari yang terbesar.

D. *Use Case Diagram*

Perancangan Use Case Diagram yang menggambarkan keseluruhan aktivitas dari pengguna dengan sistem dijelaskan pada Gambar 3.



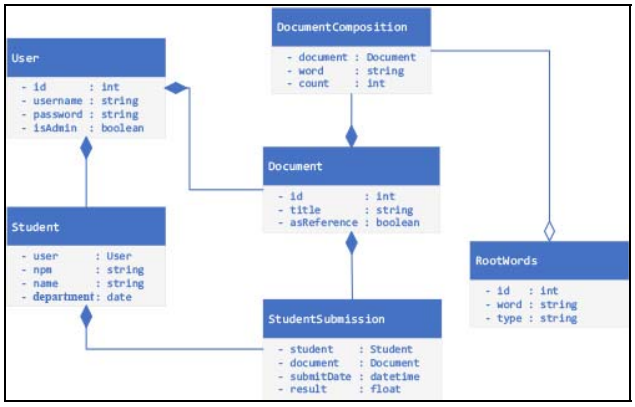
Gbr. 3 Use Case Diagram

Pada Gambar 3 terlihat dua pengguna pada sistem yaitu user dalam hal ini adalah mahasiswa, dan admin atau pengelola aplikasi dalam hal ini adalah staff. User terlibat dalam tiga proses pada aplikasi yaitu user melakukan upload file dokumen, kemudian user bisa melakukan ganti password (dimana sebelumnya data user atau mahasiswa ini sudah terinput di database oleh admin), dan yang ketiga adalah user mencetak hasil yaitu file dokumen pernyataan presentase hasil kemiripan dokumen.

Untuk admin terlibat dua proses pada aplikasi ini yaitu melakukan upload file dokumen untuk dilakukan preprocessing ke dalam database, dan menambahkan user atau data mahasiswa ke dalam database. Kedua pengguna baik user atau admin melakukan proses login terlebih dahulu sebelum menggunakan aplikasi ini.

E. *Basis Data*

Perancangan Class Diagram di sini menggambarkan hubungan antara tabel-tabel untuk penyimpanan data di database.



Gbr. 4 Use Case Diagram

Ada enam tabel penyimpanan di database yang masing-masing digunakan sesuai dengan fungsinya. Tabel yang berkaitan dengan pengguna ada dua yaitu Tabel Student untuk menyimpan data identitas pengguna dalam hal ini adalah mahasiswa, dan Tabel User untuk mengatur hak akses apakah pengguna itu admin atau bukan. Sedangkan tabel yang berhubungan dengan penyimpanan dokumen ada tiga yaitu Tabel Document, Tabel DocumentComposition, dan Tabel RootWords. Tabel Document digunakan untuk menyimpan judul dokumen dan link lokasi penyimpanan dokumen, sedangkan Tabel DocumentComposition digunakan untuk menyimpan semua bagian komponen atau isi dari setiap dokumen yang terunggah. Hasil preprocessing dari setiap dokumen disimpan dalam Tabel RootWords. Informasi atau data yang tersimpan di tabel inilah yang nantinya akan diproses oleh algoritma deteksi kemiripan yaitu Levenshtein Distaince dan Cosine Similarity.

Tabel yang terakhir yaitu Tabel StudentSubmission. Tabel ini digunakan untuk menyimpan hasil deteksi kemiripan dan menyimpan informasi hubungan antara dokumen yang diunggah dengan identitas pemilik dokumen. Informasi "dokumen siapa dengan judul apa" dan "prosentase kemiripan dokumen" ini nantinya yang akan digunakan untuk mengisi informasi yang harus ada halaman form dokumen hasil deteksi kemiripan.

III. PEMBAHASAN

A. *Implementasi Database*

Sebagaimana telah dijelaskan sebelumnya pada Bab 3 bahwa terdapat 6 tabel yang digunakan yaitu tabel user, student, document_compositions, document, student_submissions, dan rootwords.

Tabel user digunakan untuk menyimpan data login pengguna aplikasi, terdiri dari atribut "id" sebagai primary key login pengguna, "name" untuk username login pengguna, "password" untuk password login pengguna, dan "is_admin" untuk keterangan jika "1" maka sebagai admin, jika "0" maka bukan admin.

Tabel student digunakan untuk menyimpan data mahasiswa pengguna aplikasi, terdiri dari atribut "user" sebagai primary key ID mahasiswa yang didapatkan dari ID login pengguna, "npm" untuk NPM mahasiswa, "name" untuk nama mahasiswa, dan "major" untuk jurusan mahasiswa.

Tabel document digunakan untuk menyimpan data identitas dokumen yang diunggah, terdiri dari atribut "id" sebagai primary key ID dokumen, "title" untuk judul dokumen, dan "as_reference" untuk keterangan bahwa dokumen tersebut sebagai pembanding (1) atau bukan (0).

Tabel document_compositions digunakan untuk menyimpan kata-kata penting yang terkandung di setiap dokumen, terdiri dari atribut "document" sebagai primary key ID dokumen yang didapatkan dari tabel "document", "word" untuk kata-kata penting yang diekstrak dari dokumen yang diunggah, dan "count" untuk jumlah dari setiap kata di kolom "word" yang ada di dokumen yang diunggah.

Tabel "student_submissions" digunakan untuk menyimpan data hasil kemiripan dokumen yang diunggah mahasiswa di aplikasi, terdiri dari atribut "document" sebagai primary key ID dokumen yang didapatkan dari tabel "document", "student" sebagai primary key ID mahasiswa yang didapatkan dari tabel "student", "submit_date" untuk menyimpan tanggal dokumen diunggah, dan "result" untuk menyimpan hasil prosentase kemiripan dokumen yang diunggah.

Dan terakhir adalah Tabel "root_words" digunakan untuk menyimpan perpustakaan kosakata berupa kata dasar yang nanti digunakan untuk acuan deteksi kata dalam suatu dokumen, terdiri dari atribut "id" sebagai primary key ID kata, "word" untuk menyimpan kosakata acuan, dan "type" untuk jenis tipe dari kosakata apakah kata benda (nomina), kata kerja (verba), dan lainnya.

B. *Implementasi Sistem*

Desain sistem baik antarmuka maupun proses bisnis diimplementasikan ke dalam aplikasi web dengan bahasa pemrograman PHP untuk antarmuka dan Python untuk algoritmanya. Hasil implementasi sistem dilakukan secara terstruktur dalam tampilan-tampilan layout aplikasi websitenya sesuai pada cara kerja dan fungsionalitas pada setiap layoutnya.

1. Halaman Login Admin dan Mahasiswa

Halaman login ini merupakan halaman awal dari aplikasi. Halaman ini digunakan untuk login pengguna sebelum masuk ke aplikasi. selain itu Halaman ini digunakan sebagai pembeda antara hak akses Mahasiswa dan Admin.

2. Halaman website untuk hak akses Admin

Ketika login pengguna dikenali sebagai Admin, maka akan tampil beberapa halaman website berikut ini:

a. Halaman Upload Referensi : halaman ini digunakan oleh Admin untuk mengunggah dokumen-dokumen jurnal pembanding ke dalam database sistem.
b. Halaman Manajemen Data Mahasiswa : halaman ini digunakan oleh Admin untuk mengelola Data Mahasiswa mulai dari input data baru, edit data, hapus data, dan tampil data mahasiswa secara keseluruhan.
c. Halaman Manajemen Data User : halaman ini digunakan oleh Admin untuk mengelola data user mulai dari input data baru, edit data, hapus data, dan tampil data user secara keseluruhan.

3. Halaman website untuk hak akses Mahasiswa

Ketika login pengguna dikenali sebagai Mahasiswa, maka akan tampil beberapa halaman website berikut ini:

a. Halaman Beranda : halaman ini untuk menampilkan ucapan selamat datang telah mengunjungi aplikasi dan informasi singkat mengenai aplikasi.
b. Halaman Cek Plagiasi Dokumen : halaman ini digunakan Mahasiswa untuk mengupload dokumen jurnal yang akan dibandingkan dan dihitung nilai kemiripannya.

Pada halaman ini isian kolom teks "NPM" dan "Nama Lengkap" otomatis sudah terisi dari data mahasiswa yang bersesuaian dengan username login pengguna. Selanjutnya mahasiswa tinggal mengisi kolom teks "Judul Skripsi" dan mengklik tombol "Browse" untuk mencari keberadaan file dokumen yang akan diunggah. Setelah judul dan file dokumen siap seperti Gambar 5, mahasiswa kemudian mengklik tombol "Proses dan Cetak" untuk memproses atau menghitung kemiripan dokumen yang diunggah dengan dokumen referensi yang ada di database.

Gbr. 5 Tampilan Halaman Isian Deteksi Dokumen

Setelah proses perhitungan kemiripan selesai dijalankan, maka aplikasi akan mencetak file dokumen anti plagiasi. Dokumen ini berisi identitas nama mahasiswa, NPM mahasiswa, judul jurnal publikasi, serta persentase hasil kemiripan jurnal publikasi yang diunggah.

Selain itu dokumen ini juga berisi kalimat pernyataan anti plagiasi oleh mahasiswa yang menjamin bahwa hasil karyanya terbukti tidak plagiat dengan karya mahasiswa alumni yang diunggah sebelumnya. Dokumen ini ditunjukkan pada Gambar 6 yang mana nantinya dokumen ini dapat digunakan untuk salah satu persyaratan yudisium kelulusan mahasiswa.

Algoritma Levenshtein Distance dan Cosine Similarity pada aplikasi diimplementasikan dengan menggunakan bahasa pemrograman Python yang dijadikan Application Programming Interface (API) pada aplikasi web sebagaimana dituliskan pada kode program berikut ini.

Gbr. 6 Tampilan Dokumen Hasil Deteksi Kemiripan

API ini dijalankan pada saat tombol "Proses dan Cetak" diklik di halaman Deteksi Dokumen untuk hak akses Mahasiswa. Sedangkan sebelum menjalankan algoritma ini, dokumen harus melalui tahapan preprocessing yaitu casefolding, tokenizing, filtering, stemming, dan sorting.

Uji coba pada sistem ini dilakukan dengan dua skenario. Skenario yang pertama adalah dengan membandingkan persentasi hasil kemiripan suatu dokumen yang isinya sama dengan ada di sistem dan dengan perhitungan teknis. Uji coba ini digunakan untuk menguji performansi algoritma apakah hasilnya sudah berjalan sesuai alur tahapan algoritma atau belum. Dari hasil uji coba, sistem menuliskan persentase 100% untuk dokumen yang sama dan kurang dari 100% untuk dokumen yang memiliki kemiripan dengan tingkatan tertentu.

Skenario yang kedua adalah menjalankan proses bisnis pada perancangan untuk menguji apakah aplikasi sudah berjalan dengan baik sesuai dengan alur prosedur aktifitas pengguna dan fungsi-fungsinya. Dan dari hasil uji coba pada skenario ini, program mampu bekerja dengan setiap fungsionalitas yang baik.

IV. KESIMPULAN

Dari rangkaian implementasi sistem dan hasil uji coba, maka kesimpulan dari hasil penelitian yang telah dilakukan adalah sebagai berikut:

1. Sistem aplikasi dapat digunakan oleh mahasiswa untuk membantu mengecek jurnal publikasi skripsinya apakah ada indikasi kemiripan dengan dokumen jurnal mahasiswa lain dengan prosentase kemiripan tertentu.
2. Dari hasil uji coba proses bisnis aplikasi dinyatakan bahwa setiap fungsionalitas dari setiap bagian aplikasi dapat bekerja dengan baik.
3. Algoritma Levenshtein Distance dan Cosine Similarity mampu mendeteksi dan mengukur tingkat kemiripan dokumen jurnal publikasi skripsi mahasiswa yang diunggah pada aplikasi.

Pada penelitian selanjutnya, aplikasi ini bisa dikembangkan dengan meletakkan pada sistem yang lebih luas lagi yaitu sistem publikasi dan plagiasi karya ilmiah dosen dan mahasiswa. Dimana pada sistem ini nantinya, selain sebagai tempat publikasi karya ilmiah, dokumen yang akan dicek plagiasinya, dalam hal ini adalah dokumen yang dibandingkan kemiripannya, akan memiliki ukuran baik jumlah halaman maupun file yang lebih besar daripada dokumen jurnal publikasi.

REFERENSI

[1] Sora, 2014, Mengetahui Pengertian Dokumen dan Dokumentasi, http://www.pengertianku.net /2014/09/mengetahui-pengertian-dokumen-dan-dokumentasi.html, Diakses pada 2 Juni 2015.

[2] NN, 2019, Dokumen Elektronik, http:// http://kamusbisnis.com/arti/dokumen-elektronik/. Diakses pada 18 Februari 2019.

[3] Sastroasmoro, S., 2007, Beberapa Catatan Tentang Plagiarisme, Majalah Kedokteran Indonesia, Volume : 57, Nomor : 8, Agustus 2007.

[4] Harlian, M., 2015. Text Mining, http://iwanarif.lecturer.pens.ac.id/kuliah /dm/6Text%20 Mining.pdf, Diakses pada 4 Juni 2015.

[5] Ijalandhika, 2015. https://ijalandhika.wordpress.com/2015/03/15/algoritma-levenshtein/. Diakses pada 18 Februari 2019.

[6] Andriyani, N.M., 2010 , Implementasi Algoritma Levenshtein Distance dan Metode Empiris untuk menampilkan saran perbaikan kesalahan pengetikan dokumen berbahasa Indonesia, Skripsi, Teknik Informatika, Universitas Udayana, Bali.