

# IMPLEMENTASI SPELLING CORRECTION DENGAN D-LD DAN RABIN KARP PLAGIARISM CHEKING PADA APLIKASI PENDAFTARAN SKRIPSI

Viny Christanti M.<sup>1</sup>, Bagus Mulyawan<sup>2</sup>, Desi Arisandi<sup>3</sup>

<sup>1,2,3</sup> Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas  
Tarumanagara, Jln. Letjen S. Parman No. 1, Jakarta, 11440, Indonesia  
E-mail: <sup>1</sup>viny@untar.ac.id, <sup>2</sup>bagus@untar.ac.id, <sup>3</sup>desia@fti.untar.ac.id

## **Abstract**

*This study aims to produce an application that is useful for improving the quality of student theses. The use of the spell-checking application can help reduce spelling errors in abstracts inputted by students. Making this application can help students to produce theses that avoid copying with the previous thesis. The results of this study have resulted in the design and manufacture of a thesis registration system that is equipped with a spelling correction menu and a plagiarism check. The spelling correction menu is created using the D-LD method to examine abstract documents. While the Rabin Karp method is used to check the similarity of documents with other documents. In this study the user can see misspelled words based on the mark that appears on each wrong word. Users can see the percentage of thesis similarity with other theses.*

**Kata kunci**—*Damerau-Levensthein Distance, Rabin Karp, Similarity Document, Spelling Correction*

## **ABSTRAK**

*Penelitian ini bertujuan untuk menghasilkan sebuah aplikasi yang berguna untuk meningkatkan kualitas skripsi mahasiswa. Penggunaan aplikasi pemeriksa ejaan dapat membantu mengatasi kesalahan ejaan pada abstrak yang diinput oleh mahasiswa yang nantinya akan ditampilkan pada abstrak skripsi setiap mahasiswa. Pembuatan aplikasi ini juga akan membantu mahasiswa untuk menghasilkan skripsi yang terhindar dari penjiplakan dengan skripsi sebelumnya. Hasil penelitian ini telah menghasilkan perancangan dan pembuatan sistem pendaftaran skripsi dengan dilengkapi menu spelling correction dan cek plagiat. Menu spelling correction dibuat dengan metode D-LD yang dapat memeriksa dokumen abstrak. Sedangkan menu untuk memeriksa plagiat menggunakan metode Rabin Karp untuk memeriksa kemiripan dokumen dengan dokumen lainnya. Pada penelitian ini telah dilakukan penelitian untuk memperbaiki hasil koreksi ejaan dengan D-LD dan menambahkan menu pemeriksaan plagiat dengan metode Rabin Karp.*

**Kata kunci**—*Damerau-Levensthein Distance, Rabin Karp, Similarity Document, Spelling Correction*

## 1. PENDAHULUAN

### 1.1 Latar Belakang

Pendaftaran skripsi adalah salah satu proses pendataan skripsi yang diajukan pada saat mahasiswa melakukan tugas akhir untuk memperoleh kelulusan. Banyaknya skripsi yang dibuat membuat timbulnya banyak permasalahan pada pengelolaan data skripsi [1]. Beberapa masalah tersebut antara lain kesalahan pengetikan judul atau abstrak skripsi pada saat pendataan dilakukan, terjadinya persamaan judul atau isi skripsi, sulitnya mahasiswa mencari referensi tugas akhir sebelumnya apabila melakukan pencarian secara manual dan lainnya. Masih terdapat masalah-masalah yang timbul dalam mengelola tugas akhir baik dari sisi fakultas maupun dari sisi mahasiswa.

Masalah-masalah pengelolaan skripsi mahasiswa biasanya diselesaikan dengan membangun sebuah sistem online yang dapat digunakan mahasiswa untuk mendaftar skripsi, mencari referensi skripsi sebelumnya, mencari topik skripsi dan lainnya [1]. Sedangkan dari sisi fakultas proses pengelolaan skripsi secara online dapat mengurangi pekerjaan karyawan pada saat pendataan skripsi dimana mahasiswa dapat membantu menginput data skripsi. Dan bukan karyawan sendiri yang mengetikan data-data skripsi mahasiswa. Selain itu memudahkan bagi fakultas dalam proses pengarsipan dan lainnya. Oleh karena itu sistem pengelolaan skripsi perlu dilengkapi berbagai macam fitur tambahan yang dapat berguna bagi seluruh pihak.

Secara umum fasilitas koreksi ejaan dapat merupakan *stand alone application* atau *embedded system*. Koreksi ejaan sebagai *stand alone application* adalah aplikasi terpisah yang disediakan untuk memeriksa ejaan. Sedangkan *embedded application* adalah koreksi ejaan yang menjadi sebuah fitur pada application lainnya seperti pada text editor atau *word processing*. Saat ini *word processing* seperti *Microsoft Office* atau *Open Office*, sudah memiliki *auto spelling correction* yang dapat digunakan untuk membantu memberikan koreksi pada ejaan yang salah yang disediakan untuk Bahasa Inggris. Penggunaan koreksi ejaan pada text editor biasanya pengguna dapat mengetik dan secara otomatis kata akan diperiksa dan dikoreksi apabila terdapat kesalahan.

Pada penelitian sebelumnya, peneliti telah melakukan penelitian untuk membuat koreksi ejaan Bahasa Indonesia sebagai *stand alone application*. Peneliti telah menghasilkan 2 aplikasi yang dapat digunakan sebagai sistem koreksi ejaan yang dapat digunakan secara mandiri untuk Bahasa Indonesia [2][3]. Pada penelitian pertama, diteliti sebuah koreksi ejaan untuk menghasilkan metode yang dapat melakukan pemeriksaan kata dan koreksi kata dengan cepat dan akurat [2]. Sedangkan pada penelitian berikutnya dilakukan penelitian untuk menghasilkan sistem koreksi ejaan yang dapat memeriksa dan mengkoreksi ejaan untuk secara keseluruhan dari sebuah dokumen [3].

Hasil kedua penelitian tersebut akan digunakan sebagai referensi untuk membangun sebuah koreksi ejaan yang dapat berguna bagi pendaftaran skripsi. Banyaknya data skripsi yang saat ini sudah disimpan dalam bentuk digital maka perlu sebuah sistem online berupa web atau android yang dapat digunakan untuk membantu pengguna mengelola data skripsi yang ada dalam sebuah perguruan tinggi [1][4]. Saat ini mahasiswa dapat melihat karya-karya skripsi mahasiswa senior dengan mudah melalui website perguruan tinggi masing-masing. Mahasiswa tidak perlu lagi mencari skripsi senior dengan cara manual.

Dalam pengelolaan skripsi tersebut, data yang ditampilkan haruslah benar terhindar dari berbagai kesalahan. Kesalahan pengejaan masih sering terjadi pada data skripsi yang ditampilkan secara online akibat kurang telitinya proses pengetikan. Agar terhindar dari kesalahan pengejaan maka perlu dibuat sebuah sistem yang memiliki modul koreksi ejaan yang

langsung terhubung pada saat mahasiswa memasukan abstrak skripsi masing-masing ke website pengelolaan skripsi.

Pada penelitian sebelumnya metode *Damerau-Levenshtein Distance* merupakan metode yang memiliki dasar operasi pengubahan, penambahan, dan penghapusan karakter untuk menghitung besarnya biaya transformasi kata seperti yang telah dilakukan. Penggunaan metode ini telah menghasilkan akurasi sebesar 85% [2]. Akurasi tersebut perlu ditingkatkan agar dapat mengkoreksi ejaan lebih tepat. Oleh karena itu dibutuhkan penelitian untuk meningkatkan akurasi koreksi ejaan. Pada penelitian sebelumnya jenis kata yang dapat dikoreksi adalah jenis kesalahan kata yang berupa *non-real word error*. Artinya kata-kata yang dapat diperbaiki hanyalah kata-kata yang tidak memiliki arti atau memiliki kesalahan yang tidak ada didalam kamus seperti “kamu” diketik “kmu” [5].

Permasalahan lainnya adalah sering terjadinya judul dan isi skripsi yang sama dengan skripsi senior. Kesamaan judul atau isi skripsi dapat saja terjadi secara sengaja atau tidak sengaja. Banyaknya topik skripsi tidak menutup kemungkinan secara tidak sengaja mahasiswa memilih topik yang sama. Kesamaan topik dan isi skripsi bisa saja juga terjadi karena mahasiswa secara sengaja menyalin skripsi senior lainnya yang mungkin bias diperoleh dari senior atau menyalin dari buku skripsi. Agar terhindar dari kesamaan judul dan isi skripsi sebaiknya dibuat sebuah sistem yang dapat menyaring setiap judul yang diajukan oleh mahasiswa sebagai topik skripsinya.

Penambahan fungsi pemeriksaan plagiat dilakukan sebagai salah satu bentuk pencegahan terjadinya plagiarisme atau duplikasi baik disengaja maupun tidak. Sehingga diharapkan penggunaan sistem membantu mahasiswa untuk memeriksa terlebih dahulu apakah tulisan yang dibuat sudah ada pada skripsi sebelumnya. Fungsi plagiat dibuat dengan menggunakan konsep menghitung kemiripan antar satu dokumen dengan dokumen lainnya.

Plagiarisme umumnya dibagi menjadi dua jenis di antaranya plagiarisme ide dan kata demi kata yakni pencurian ide dan kata demi kata, yang dapat mencakup kalimat, paragraf, atau keseluruhan tulisan [6]. Dalam karya tulis ilmiah, plagiarisme ide sering dihubungkan dengan laporan hasil penelitian replikatif yaitu penelitian yang mengulang penelitian orang lain [6]. Sedangkan pada plagiarisme kata demi kata, penulis menggunakan kata-kata penulis lain tanpa menyebutkan sumbernya [7]. Seperti diketahui metode Rabin Karp adalah metode pencocokan substring yang cukup cepat namun memiliki beberapa kelemahan. Pada penelitian digunakan metode Rabin Karp untuk melakukan pemeriksaan plagiat yang lebih cepat.

Tujuan dari penelitian ini adalah membuat aplikasi pendaftaran skripsi yang dilengkapi dengan modul spelling correction dan plagiarism checking. Walaupun banyak aplikasi *text editor* yang dapat digunakan sebagai pemeriksa ejaan namun aplikasi tersebut berada terpisah dari sistem pendaftaran skripsi. Pemeriksa ejaan yang dimaksud pada penelitian ini adalah pemeriksa ejaan yang terintegrasi dengan pendaftaran skripsi sehingga dapat membantu mahasiswa dalam mengkoreksi kata-kata yang pada saat diketik di menu pengisian abstrak skripsi pada sistem pendaftaran. Sehingga hasil input mahasiswa terhindar dari kesalahan pengetikan dan dapat ditampilkan pada sistem pencari skripsi mahasiswa dengan baik dan benar.

Saat ini, aplikasi sejenis yang dapat digunakan sebagai alat untuk memeriksa plagiat seperti Turnitin dan lainnya sudah banyak. Namun aplikasi tersebut merupakan aplikasi berbayar yang cukup mahal. Dimana harga dari aplikasi tersebut bisa mencapai 150 juta pertahun. Aplikasi yang sudah ada memiliki fitur yang sangat lengkap dan bukan hanya memeriksa plagiat terhadap dokumen secara lokal, aplikasi tersebut dapat memeriksa dan membandingkan secara online dengan dokumen yang ada di dunia maya. Mahal dan luasnya cakupan aplikasi yang sudah ada membuat kesulitan pada pemakaian pemeriksaan plagiat.

Pada penelitian ini fokus pemeriksa plagiat adalah pemeriksaan terhadap skripsi mahasiswa sebelumnya bukan untuk diperiksa secara luas terhadap dokumen yang ada di dunia maya. Sehingga peruntukan aplikasi ini lebih kepada membantu mahasiswa dalam mencari topik dan menghindari terjadinya pengajuan judul topik yang sama dengan topik mahasiswa sebelumnya. Selain itu plagiat lebih ditujukan pada mencegah terjadinya plagiarism secara sengaja atau tidak dengan skripsi mahasiswa sebelumnya.

## 2. METODE PENELITIAN

### 2.1 Spelling Coreection dengan D-LD

Proses pendeteksian salah eja dilakukan dengan mencari kata yang tidak tercantum dalam suatu kamus ataupun korpus. Makin besar kamus atau korpus yang dimiliki maka semakin bagus untuk mendeteksi kata yang salah eja [8]. Kemudian mencoba membangkitkan calon yang memiliki urutan kata yang mirip dengan kata yang salah. Kemudian calon tersebut akan diurutkan menggunakan suatu perhitungan jarak antara sumber dan *target* kesalahan [8]. Konsep tersebut dipakai pada setiap sistem koreksi ejaan.

*Damerau-Levensthein Distance* atau DLD sendiri merupakan salah satu metode yang memiliki dasar operasi pengubahan, penambahan, dan penghapusan karakter yang dimodifikasi oleh Damerau dengan menambahkan konsep pertukaran dua huruf [9]. DLD hanya merupakan salah satu algoritma perhitungan yang menghitung jarak kedekatan antara 2 kata. Setelah 2 kata dihitung kedekatannya maka perlu dilakukan perhitungan maka akan diperoleh daftar kata yang memiliki jarak terdekat dengan kata yang salah eja. Sederetan kata tersebut perlu dipilih untuk menghasilkan 1 kata yang dianggap paling tepat untuk mengkoreksi kata yang salah eja. Pada penelitian sebelumnya pemilihan kata yang paling tepat dikombinasikan dengan menggunakan N-Gram [2][3]. Namun kekurangan metode ini adalah belum dapat memperbaiki kesalahan dalam bentuk *real word*.

### 2.2 Plagiarisme dengan Rabin Karp

Plagiarisme adalah sebuah bentuk penjiplakan atau menyalin karya lain tanpa mengutip atau tanpa merubah setiap kata yang ada di dalam karya tersebut. Pendeteksi plagiarism sudah banyak dibuat dan diteliti berbagai algoritma dipakai untuk mendeteksi isi kalimat apakah sama atau tidak walaupun kalimat tersebut dirubah bentuk. Algoritma Winnowing merupakan salah satu dari beberapa metode yang berfungsi sebagai pengecekan plagiarisme. Dengan algoritma ini, tulisan target dibandingkan dengan tulisan sumber, lalu hasil persentase kesamaan kata dihitung menggunakan dua metode untuk melihat persentase kesamaan kata yang lebih tinggi.

Algoritma *Rabin Karp* diciptakan oleh Michael O. Rabin dan Richard M. Karp pada tahun 1987 menggunakan hashing untuk menemukan *pattern* di dalam *string* teks. Metode ini merupakan sebuah metode alternatif untuk menyelesaikan masalah *string matching* yang menggunakan fungsi hash [10]. Metode ini melakukan suatu proses untuk mencari kesamaan pada sebuah *pattern* dengan sebuah *text string*.

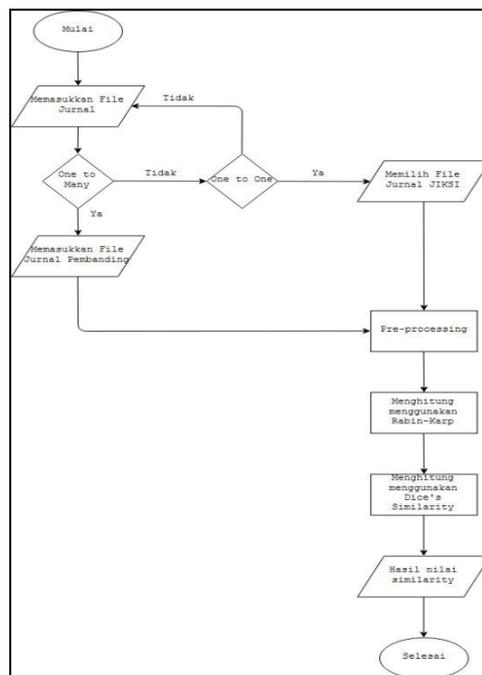
Algoritma *Rabin Karp* adalah algoritma pencocokan string yang menggunakan fungsi *hash* sebagai pembanding antara *string* yang dicari ( $m$ ) dengan *substring* pada teks ( $n$ ) [13]. Apabila hash keduanya sama maka akan dilakukan perbandingan sekali lagi terhadap karakter-karakternya. Apabila hasil keduanya tidak sama, maka *substring* akan bergeser ke kanan. Pergeseran dilakukan sebanyak  $(n-m)$  kali. Perhitungan nilai hash yang efisien pada saat pergeseran akan mempengaruhi performa dari algoritma ini.

Berikut ini merupakan langkah-langkah dalam mencocokkan kata dengan algoritma Rabin-Karp [10]:

1. Lakukan proses *preprocessing* (tokenisasi, *case folding*, dan *filtering*) untuk membuat kalimat menjadi kata, menghilangkan huruf besar, membuang kata-kata tidak penting dan memotong imbuhan.
2. Tentukan panjang *k-gram* yang akan digunakan dan pecah kata yang digunakan sesuai dengan panjang *k-gram* yang ditentukan.
3. Mencari nilai *hash* dari teks sumber dan kata yang ingin dicari menggunakan fungsi  $hash = T[i]B^{m-1} + \dots + T[l+m-1] \text{ mod } q$ .  $T[i]$  merupakan nilai ASCII dari string yang dicari sedangkan nilai  $q$  merupakan angka modulo yang berupa bilangan prima.
4. Lakukan iterasi dari indeks  $i=0$  sampai  $i=n-m+1$  untuk membandingkan nilai *hash* dari kata yang ingin dicari dengan nilai *hash* dari teks sumber pada indeks  $i$  sampai dengan  $i+m-1$ . Jika nilai *hash* sama, maka akan diperiksa lebih lanjut apakah kata yang dicari sama dengan bagian teks dari sumber pada indeks  $i$  sampai dengan  $i+m-1$ . Jika sama, maka telah ditemukan kata yang cocok. Jika tidak maka dilanjutkan dengan membandingkan nilai *hash* dari kata yang dicari dengan nilai *hash* teks sumber pada indeks berikutnya.
5. Lakukan perhitungan  $String\ matching = 2C/A+B * 100\%$  untuk menentukan kemiripan antara 2 string,  $C$  merupakan nilai dari *hash value* yang sama sedangkan  $A$  merupakan jumlah *K-Gram* teks sumber dan  $B$  merupakan jumlah *K-Gram* teks target.

Algoritma Rabin Karp adalah salah satu algoritma yang sudah digunakan sebagai pendeteksi plagiat pada tingkat perguruan tinggi untuk memeriksa apakah ada pencurian karya yang dilakukan antar mahasiswa [11]. Algoritma Rabin Karp merupakan algoritma yang menjadi dasar algoritma winnowing. Pada dasarnya kedua algoritma tersebut memiliki konsep yang sama yaitu merubah karakter secara berurutan dalam bentuk ASCII dan membangun nilai hash. Dimana algoritma Rabin Karp merupakan salah satu algoritma substring yang dapat mendeteksi kesamaan substring dengan cepat.

Menu plagiat terdiri menjadi dua bagian yaitu skema *one to one* dan skema *one to many*. Pada gambar 1 dapat dilihat skema *one to many* yaitu pengecekan satu dokumen dengan banyak dokumen sekaligus.

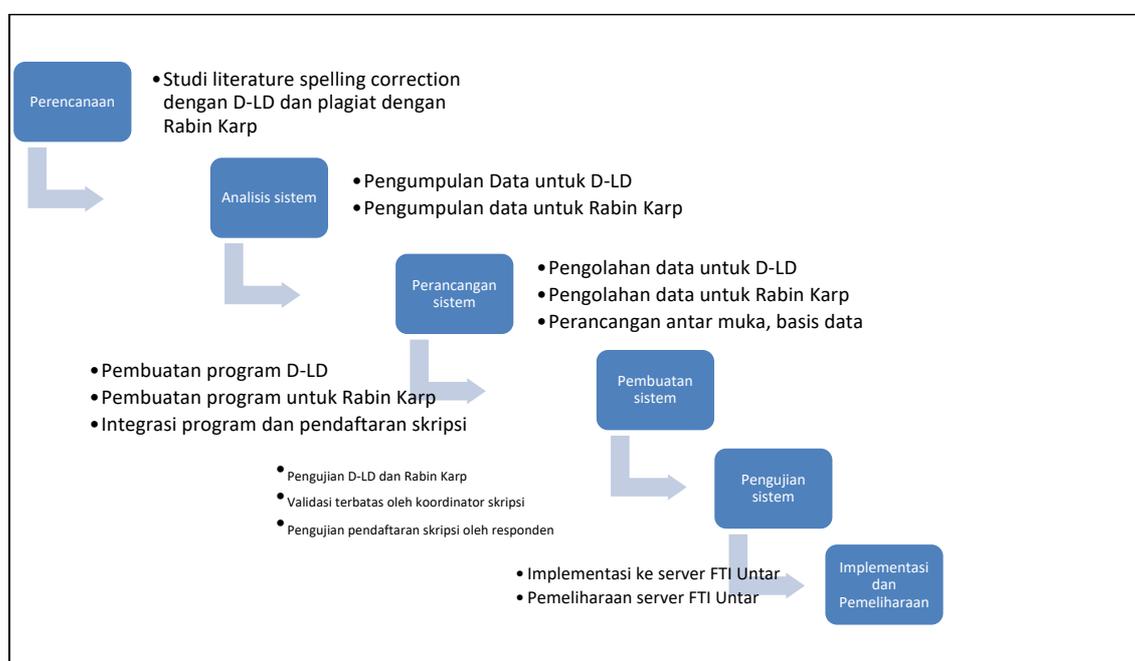


Gambar 1 Skema Pemeriksaan Plagiat

Pada metode *one to many*, hal pertama yang perlu dilakukan adalah menyiapkan dokumen abstrak skripsi yang nantinya dibandingkan dengan abstrak skripsi yang sudah tersimpan dalam sistem. Proses selanjutnya yang dilakukan yaitu pra-pemrosesan berupa *case folding*, tokenisasi dan *stopword removal*. Kemudian dilakukan pengecekan kata pada kamus untuk memeriksa sinonim kata. Lalu, dilakukan *stemming* untuk menghilangkan imbuhan dan menghilangkan spasi sehingga terbentuk abstrak baru.

### 2.3 Pendaftaran Skripsi dengan *Spelling Correction* dan *Plagiarisme*

Penelitian ini terdiri dari 3 tahapan. Tahap pertama adalah pembuatan sistem pengkoreksian kalimat, tahap kedua adalah pembuatan modul untuk pemeriksaan plagiarisme dan yang terakhir adalah penerapan *spelling correction* dan *plagiarism checking* terhadap pendaftaran skripsi. Metode yang digunakan pada penelitian ini adalah waterfall pada metode *System Development Life Cycle* (SDLC) yang merupakan salah satu metode pembuatan program aplikasi. SDLC terdiri dari perencanaan, analisis sistem, perancangan sistem, pembuatan sistem, pengujian sistem oleh peneliti dan pemeliharaan. Tahapan dapat dilihat pada gambar 2.



Gambar 2. Tahapan SDLC pendaftaran skripsi dengan spelling correction dan plagiarisme checking

Penelitian ini dilakukan oleh tim peneliti yang terdiri dari 3 orang dosen, 2 orang mahasiswa dan 1 orang asisten penelitian dengan tugas sebagai berikut:

1. Ketua tim adalah salah satu dosen memiliki kepakaran dalam bidang Information Retrieval dan Natural Language Processing. Ketua tim bertugas merencanakan, menganalisis dan merancang kebutuhan sistem metode D-LD dan Rabin Karp.
2. Anggota tim pertama adalah pakar didalam bidang jaringan dan aplikasi web yang bertugas menyediakan server dan mengatur web
3. Anggota tim kedua adalah pakar di bidang analisis dan perancangan sistem, database yang bertugas merancang antar muka dan database aplikasi.
4. Mahasiswa FTI 2 orang yang membantu dalam menyusun tiap modul sistem dan membuat dokumentasi. Beberapa modul dapat menjadi bagian tugas akhir mahasiswa.
5. Asisten peneliti adalah alumni mahasiswa FTI yang bertugas untuk membantu membuat program, menyiapkan dan mengimplementasikan program ke server.

Penelitian dilakukan dengan menggunakan PC dan laptop sebagai perangkat keras. Sedangkan perangkat lunak yang digunakan adalah office, Java, Perl dan Html. Pada penelitian ini juga perlu menyediakan satu buah unit PC yang akan digunakan sebagai server untuk mengimplementasikan aplikasi sehingga dapat digunakan oleh mahasiswa untuk mencoba secara langsung pendaftaran skripsi.

### 3. HASIL DAN PEMBAHASAN

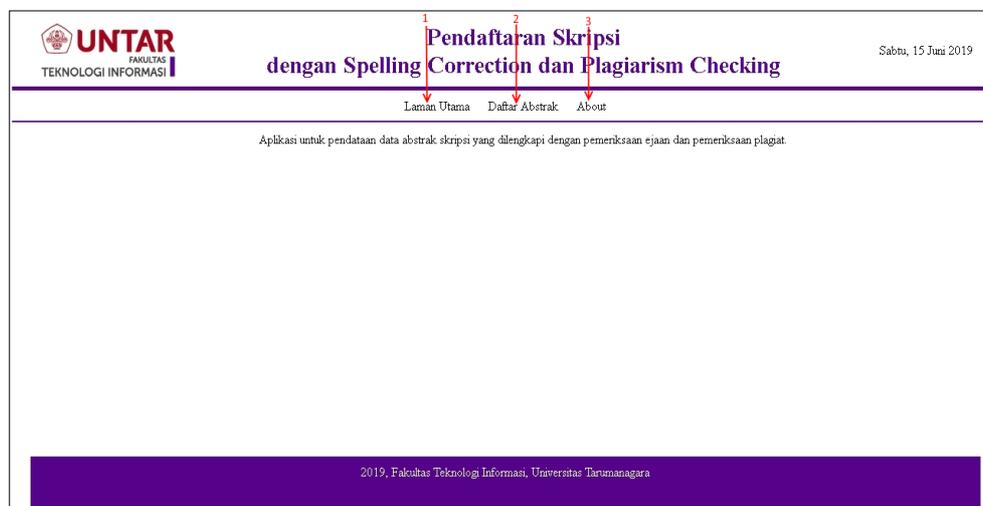
#### 3.1 Hasil Penelitian

Pembuatan “Pendaftaran Skripsi dengan Spelling Correction dan Plagiarism Checking” dilakukan dengan menggabungkan beberapa penelitian sebelumnya. Pada penelitian ini hanya difokuskan pada proses pendataan data abstrak skripsi yang diinput oleh mahasiswa pada saat melengkapi data skripsi yang dibutuhkan untuk memenuhi persyaratan pendaftaran skripsi. Penelitian tahap sebelumnya telah menghasilkan Koreksi Ejaan dan Cek Plagiat untuk data abstrak. Koreksi ejaan dilakukan dengan Damerau-Levensthein Distance. Kemudian Cek Plagiat dibuat dengan metode Rabin Karp. Hasil dari penelitian tersebut dikembangkan untuk menghasilkan hasil yang lebih baik.

Aplikasi “Pendaftaran Skripsi dengan Spelling Correction dan Plagiarism Checking” yang terdiri dari beberapa bagian yaitu:

1. Modul Laman Utama

Pada modul ini ditampilkan menu utama dari pendaftaran skripsi. Aplikasi ini hanya merupakan aplikasi untuk mendata abstrak dari skripsi mahasiswa sehingga belum digabung dengan aplikasi sistem informasi skripsi yang mengelola pendaftaran skripsi secara keseluruhan.



Gambar 3 Modul Laman Utama

2. Modul Isi data skripsi

Pada modul ini pengguna diminta untuk mengisi data abstrak yang berupa judul skripsi, isi abstrak skripsi dan kata kunci yang ada di dalam skripsi.

The screenshot shows a web form titled "Pendaftaran Skripsi dengan Spelling Correction dan Plagiarism Checking". At the top, there are navigation links: "Laman Utama", "Daftar Abstrak", and "About". Below these is a button labeled "Kembali ke daftar abstrak" with a red arrow pointing to it labeled "4". The form itself is titled "Form abstrak" and contains three main input areas: "Judul" (Title), "Abstrak" (Abstract), and "Keyword". Red arrows point to these areas with labels: "1" points to the Title field, "2" points to the Abstract field, and "3" points to the Keyword field. Below the Keyword field, there is a red arrow pointing to the text "Belum input abstrak" labeled "5".

Gambar 4 Modul Isi data Skripsi

### 3. Modul koreksi ejaan

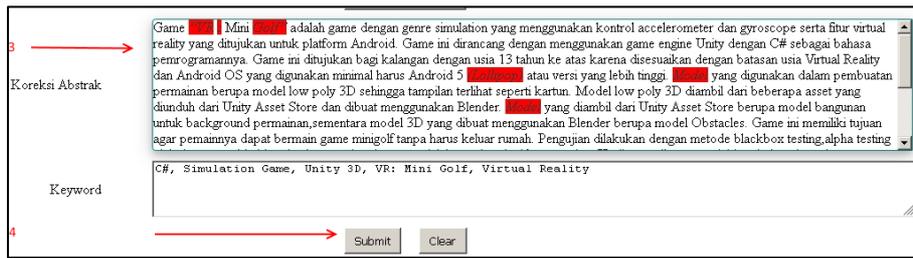
Proses koreksi ejaan terdapat pada menu isi data skripsi. Proses koreksi ejaan akan muncul setelah mahasiswa selesai mengetikkan data skripsi pada menu yang sudah disediakan dan mahasiswa mengklik menu selanjutnya untuk mengisi kata kunci. Proses koreksi ejaan dilakukan dengan metode Damerau-Levensthein Distance dan N-Gram. Setelah koreksi ejaan selesai memeriksa, maka pada modul ini akan muncul data abstrak skripsi yang diberi tanda pada kata-kata yang salah eja. Pada menu ini juga dimunculkan kata-kata yang menjadi saran untuk memperbaiki kata yang dideteksi salah eja.

The screenshot shows the "Form abstrak" after the spelling correction process. The "Judul" field contains "PEMBUATAN GAME SIMULASI 'VR: MINI GOLF' DENGAN VIRTUAL REALITY BERBASIS AND". The "Abstrak" field contains a paragraph of text with several words highlighted in red, indicating spelling errors. A red arrow labeled "1" points to the text "Ada Kesalahan Ejaan, silakan perbaiki teks abstrak di atas". Below the abstract, there is a button labeled "Ulang Periksa Ejaan" with a red arrow labeled "2" pointing to it. The "Koreksi Abstrak" field shows the same text as the abstract, but with the red highlights removed and replaced by suggestions for correction. A red arrow labeled "3" points to this field. The "Keyword" field contains "C#, Simulation Game, Unity 3D, VR: Mini Golf, Virtual Reality". A red arrow labeled "4" points to the "Submit" button at the bottom of the form.

Gambar 5 Modul Koreksi Ejaan

### 4. Modul pemeriksaan plagiat

Modul pemeriksaan plagiat tidak terlihat pada aplikasi ini. Mahasiswa hanya menekan tombol submit dan proses pemeriksaan akan otomatis bekerja dan membandingkan kemiripan data abstrak yang baru diinput dengan data abstrak yang sudah ada dalam basisdata. Proses pemeriksaan dilakukan dengan metode Rabin Karp.



Gambar 6 Modul Pemeriksaan Plagiat

5. Modul hasil persentase plagiat  
 Modul ini menampilkan hasil presentase dari kemiripan abstrak dengan basisdata. Hasil presentase kemiripan ditampilkan dalam bentuk tabel yang berisi abstrak mana saja yang memiliki tingkat kemiripan tertinggi hingga terendah.

4	Judul	5	% Dice	6	% Jaccard	7	% Rabin Karp	8	Kesimpulan
	Perancangan Game Platformer "Monster's Shooter" Berbasis Android		21.710		12.177		24.434		Plagiarisme Ringan
	Perancangan Game "Satria Ladaang" pada Platform Android		16.176		8.8		18.563		Plagiarisme Ringan
	Pembuatan game adventure & puzzle Medieval Age: The Escape Dengan Virtual Reality		13.333		7.1428		15.055		Plagiarisme Ringan
	Perancangan Game Platformer Adventure "Neo The Defender Of Nature Land" Berbasis Android		12.839		6.8601		13.458		Plagiarisme Ringan
	Perancangan Game Platformer Tiga Dimensi "End of Darkness" Menggunakan Unity		10.731		5.6701		10.921		Plagiarisme Ringan
	Pembuatan Game Action Puzzle "Soul Haunter"		10.362		5.4644		10.666		Plagiarisme Ringan
	KLASIFIKASI CITRA BATIK INDONESIA DAN MALAYSIA DENGAN METODE MODIFIED DISCRIMINANT ANALYSIS		4.4585		2.2801		4.5936		Plagiarisme Ringan
	Pengenalan penyakit kulit pada bagian wajah dengan Backpropagation Neural Network (BPNN)		4.2682		2.1806		4.3478		Plagiarisme Ringan
	Perancangan Game Combat-flight Simulator "Nightgale"		2.8985		1.4705		3.8590		Plagiarisme Ringan

Gambar 7 Modul Hasil Presentase

### 3.2 Pembahasan Koreksi Ejaan

Pada aplikasi ini, pengguna dapat melakukan input abstrak pada saat mendaftarkan isi skripsinya. Pada gambar 8 dapat dilihat input data abstrak yang dilakukan pengguna. Mahasiswa mengisi abstrak pada field yang disediakan kemudian secara otomatis sistem akan memeriksa terlebih dahulu ketikan mahasiswa tersebut.

Saat ini perkembangan teknologi informasi semakin meningkat. Kehadiran teknologi semakin memudahkan manusia dalam melakukan pekerjaan. Salah satu implementasi teknologi adalah pengenalan wajah manusia. Namun, pada implementasinya ditemui kesulitan karena, komputer tidak memiliki kecerdasan dan kemampuan seperti manusia untuk dapat membedakan wajah manusia. Setiap manusia memiliki keunikan dan ciri khas masing-masing. Antara satu manusia dengan manusia pasti memiliki perbedaan meskipun tergolong kembar.

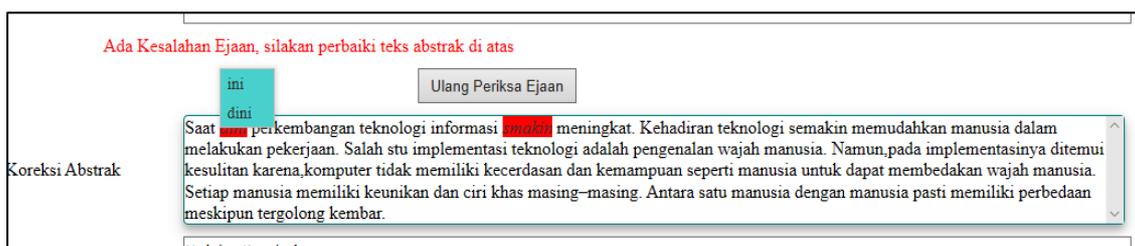
Gambar 8 Isi salah satu abstrak mahasiswa

Setelah menginput abstrak, sistem memeriksa ejaan yang diketik mahasiswa dan ternyata terdapat kesalahan pengetikan. Kesalahan berhasil dideteksi dengan munculnya tanda merah pada hasil pemeriksaan seperti pada gambar 9.

Saat **dini** perkembangan teknologi informasi **smakin** meningkat. Kehadiran teknologi semakin memudahkan manusia dalam melakukan pekerjaan. Salah stu implementasi teknologi adalah pengenalan wajah manusia. Namun,pada implementasinya ditemui kesulitan karena,komputer tidak memiliki kecerdasan dan kemampuan seperti manusia untuk dapat membedakan wajah manusia. Setiap manusia memiliki keunikan dan ciri khas masing-masing. Antara satu manusia dengan manusia pasti memiliki perbedaan meskipun tergolong kembar.

Gambar 9 Kata yang terdeteksi salah ketik

Apabila cursor diarahkan menuju kata yang salah maka akan muncul daftar kata yang menjadi saran kata seperti pada gambar berikut:



Gambar 10 Mengganti kata yang salah dideteksi

Pengguna dapat memilih salah satu kata yang dianggap sebagai kata yang tepat untuk memperbaiki kata yang salah tersebut. Apabila pengguna tidak mau menggantinya maka tidak perlu memilih salah satu kata tersebut.

### 3.3 Pembahasan Plagiat dengan Rabin Karp

Setelah pemeriksaan ejaan selesai dilakukan, tahap selanjutnya adalah melakukan submit abstrak ke dalam sistem. Pada gambar 11 dapat dilihat sebuah dokumen abstrak yang berjudul “CNN CBIR Plagiat”. Dokumen tersebut menyalin sebagian isi dokumen ( $\pm 50\%$ ) abstrak dengan judul “Klasifikasi Dengan Metode Convolutional Neural Networks Untuk Content-Based Image Retrieval”. Isi dokumen abstrak “CNN CBIR Plagiat” yang diuji adalah sebagai berikut:

Pencarian kumpulan citra yang mempunyai kemiripan dengan citra input, tanpa mengetahui nama dari citra tersebut, membuat sistem pencarian yang menerapkan konsep Content-Based Image Retrieval, sangat diperlukan. Pada umumnya, sistem CBIR menggunakan ciri visual seperti warna, tepi gambar, tekstur, dan kesesuaian nama pada citra input dengan citra-citra pada basis data. Sistem CBIR yang dibangun mempunyai 2 tahap. Tahap pertama dilakukan dengan klasifikasi untuk mendapatkan kelasnya. Selanjutnya, dari hasil klasifikasi, akan dicari kumpulan citra yang sesuai dengan citra inputnya. Metode untuk klasifikasi yaitu Convolutional Neural Networks, sedangkan retrieval, digunakan perhitungan kedekatan jarak vektor dengan Cosine Similarity.

Gambar 11 Abstrak yang diinput pada saat mahasiswa mengisi abstrak

Setelah melakukan proses pemeriksaan maka dapat disimpulkan bahwa abstrak tersebut memiliki tingkat kemiripan terbesar dengan dokumen “Klasifikasi Dengan Metode Convolutional Neural Networks Untuk Content-Based Image Retrieval” yaitu sebesar 69.565%. Dimana hal tersebut dapat terjadi karena abstrak yang diinput mengambil hampir separuh dari abstrak tersebut. Sedangkan apabila dibandingkan dengan abstrak lainnya, kemiripannya sangat kecil karena merupakan abstrak dengan topik yang berbeda.

**Tabel 1** Hasil kemiripan abstrak “CNN CBIR Plagiat” dengan abstrak lainnya

Judul	% Rabin Karp
Klasifikasi Dengan Metode Convolutional Neural Networks Untuk Content-Based Image Retrieval	69.565
Aplikasi Pencarian Objek Wisata Bersejarah Di Pulau Jawa Menggunakan Convolutional Neural Network	7.5685
Aplikasi Klasifikasi Gerak Dan Objek Video Dengan Convolutional Neural Network- Long Short Term Memory	6.3524
Klasifikasi Citra Batik Indonesia Dan Malaysia Dengan Metode Modified Discriminant Analysis	5.8495
Perancangan Game Combat-Flight Simulator "Nightingale"	3.5989
Pendeteksian Sel Darah Putih Pada Citra Preparat Tanpa Pewarnaan Dengan Hough Transform	2.3121
Pengenalan Penyakit Kulit Pada Bagian Wajah Dengan Backpropagation Neural Network (BPNN)	1.5345
Sistem Pengoreksian Ejaan Teks Bahasa Indonesia Dengan Damerau Levenshtein Distance Dan Recurrent Neural Network	1.2944
Perancangan Aplikasi Sistem Manajemen Dokumen Dan Pencarian Teks Dengan Menggunakan Metode Optical Character Recognition (OCR)	1
Perancangan Aplikasi Penjadwalan Dan Estimasi Produksi Pada PT. Indometic Dengan Metode Time Series Dan Economic Order Quantity	0.6535
Klasifikasi Citra Batik Indonesia Dan Malaysia Dengan Konsep Machine Learning Menggunakan Metode Two Dimensional PCA	0.4405
Perancangan E-Learning SMA DAMAI Pada Platform Android Dan Website	0.4179

#### 4. KESIMPULAN

Hasil penelitian ini telah menghasilkan sebuah aplikasi yang menggabungkan koreksi ejaan sebagai *embedded system* yang dapat berguna untuk memeriksa ejaan pada saat abstrak skripsi diinput ke dalam sistem. Hasil dari penelitian ini dapat diteruskan untuk disatukan dengan Sistem Informasi Skripsi FTI Untar atau sistem pengelolaan skripsi lainnya. Sehingga hasil penelitian ini dapat diterapkan dimasyarakat dan digunakan untuk meningkatkan kualitas skripsi mahasiswa.

#### UCAPAN TERIMA KASIH

Terima kasih kami ucapkan kepada Direktorat Penelitian dan Pengabdian kepada Masyarakat Untar yang telah mendanai penelitian ini. Terima kasih kepada rekan dosen, mahasiswa dan asisten peneliti yang telah membantu penelitian ini.

#### DAFTAR PUSTAKA

- [1] Yamasari, Y., Asmunin, A., dan Kurniawan, I., F., 2015, Pengembangan Sistem Informasi Manajemen Skripsi Di Jurusan Teknik Elektro, Ft Unesa (The Developing Of Management Information System Final Project At The Electrical Engineering Departement, Ft Unesa)(Bachelor Thesis Monitoring System of Electrical En., *Jurnal Manajemen Informatika*, No. 01, Vol. 4.
- [2] Christanti, V., M., dan Naga, D., S., 2018, Fast and Accurate Spelling Correction Using Trie and Damerau-levenshtein Distance Bigram, *TELKOMNIKA*, No. 2, Vol. 16, hal. 827-833.

- [3] Mawardi, V., C., Susanto, N., dan Naga, D., S., 2018, Spelling Correction for Text Documents in Bahasa Indonesia Using Finite State Automata and Levinshtein Distance Method, *MATEC Web of Conferences*, Vol. 164, hal. 01047.
- [4] Nuryana, Y., and Mulyani, A., 2017, Pengembangan Aplikasi Pengendalian Skripsi Berbasis Android Untuk Mahasiswa Dan Dosen, *Jurnal Algoritma*, No.2, Vol.14, no. 2, hal. 58-64.
- [5] Jurafsky, D., Martin, J., H., 2008, *Speech and Language Processing*, Ed. 2, Prentice Hall, Upper Saddle River.
- [6] Sastroasmoro, S, 2007, Beberapa Catatan Tentang Plagiarisme, *Majalah Kedokteran Indonesia*, No.8, Vol. 57, No. 8, hal 239-244.
- [7] Soelistyo, H., 2011, *Plagiarisme: Pelanggaran Hak Cipta Dan Etika*, Penerbit Kanisius.
- [8] Jurafsky, D., & Martin., J., H., 2014. *Speech And Language Processing*, Prentice Hall, Upper Saddle River.
- [9] Schulz, K. U., dan Mihov, S., 2002, Fast String Correction With Levenshtein-Automata, *International Journal On Document Analysis And Recognition*, No. 1, Vol. 5, hal. 67-85.
- [10] Nugroho, E., 2011, Perancangan Sistem Deteksi Plagiarisme Dokumen Teks Dengan Menggunakan Algoritma Rabin-Karp, *Skripsi/Tesis/Disertasi*, Jurusan Ilmu Komputer, Universitas Muhammadiyah, Malang.
- [11] Shivaji, S., K., dan Prabhudeva, S., 2015, Plagiarism detection by using karp-rabin and string matching algorithm together, *International Journal of Computer Applications*, No. 23, Vol. 115.