p-ISSN 2407-4489
e-ISSN 2528-7303

DETEKSI KEMIRIPAN JUDUL SKRIPSI MENGGUNAKAN ALGORITMA *LEVENSHTEIN DISTANCE* PADA KAMPUS STMIK MIC CIKARANG

Bei Harira Irawan[1], Manase Sahat H Simarangkir[2], Erlinna[3]

[1,3] STMIK MIC Cikarang
[2] Politeknik META Industri
Cikarang Bekasi, Indonesia
beiharira@gmail.com - manasemalo@politeknikmeta.ac.id - erlinnanurul286@gmail.com

Abstrak

Dengan bertambahnya skripsi dari tahun ke tahun, meningkat pula judul skripsi dan penelitian. Bila tidak dideteksi, menyebabkan penelitian yang sebenarnya sudah ada akhirnya diteliti kembali tanpa mengembangkan penelitian sebelumnya. Algoritma *Levenshtein Distance* digunakan untuk mendeteksi kesamaan pada judul skripsi yang ada. Penelitian ini digunakan untuk pencarian judul skripsi secara *multi target* dengan 6 kategori data target dalam database dan penentuan *threshold* >= 4 serta penentuan bobot *similarity* >25% dinyatakan mirip untuk ditampilkan. Data yang ditampilkan adalah dari kategori judul perancangan sistem dengan *distance* 4 dan kemiripan 42,85% dan dari kategori aplikasi berbasis web dengan *distance* 5 dan kemiripan 28,5%. Sedangkan 4 kategori lain memiliki *distance* lebih dari 5 dan persen bobot *similarity* <25% sehingga tidak dianggap mirip dan tidak ditampilkan.

Kata Kunci: multi target, algoritma levenshtein distance, threshold, bobot similarity

Abstract

With the increase in thesis from year to year, the title of thesis and research also increases. If it is not detected, it will cause existing research to be finally re-examined without developing previous research. The Levenshtein Distance algorithm is used to detect similarities in existing thesis titles. This research is used to search the thesis title in a multi-target manner with 6 categories of target data in the database and to determine the threshold >= 4 and to determine the similarity weight >25% which is stated to be similar to display. The data displayed is from the category of system design title with a distance of 4 and a similarity of 42.85% and from the category of web-based applications with a distance of 5 and a similarity of 28.5%. Meanwhile, 4 other categories have a distance of more than 5 and the percent weight of similarity is <25% so they are not considered similar and are not displayed.

Keywords: multi-target, levenshtein distance algorithm, threshold, weight of similarity.

p-ISSN 2407-4489
e-ISSN 2528-7303

PENDAHULUAN

Tugas akhir atau skripsi bagi mahasiswa untuk mendapatkan gelar sarjana merupakan salah satu kewajiban yang harus dipenuhi. Dengan bertambahnya mahasiswa dari tahun ke tahun, meningkat pula judul skripsi dan penelitian. Mahasiswa sering kesulitan mendeteksi apakah judul yang diajukan sudah ada atau belum. Bagi dosen pembimbing pun seringkali sulit untuk menentukan apakah judul yang diajukan layak diterima atau tidak karena kekurangan informasi akan data judul-judul skripsi di kampus. Hal ini menyebabkan masih adanya penelitian yang sebenarnya sudah ada namun akhirnya di teliti kembali tanpa mengembangkan penelitian sebelumnya serta dikhawatirkan ada beberapa skripsi sama yang ternyata sudah dibuat sebelumnya sehingga terjadi kesamaan judul dan bisa dianggap sebagai tindak plagiat.

Penelitian ini menggunakan algoritma *Levenshtein Distance* yang digunakan untuk suatu pendekatan yang dapat memberikan alternatif kesamaan judul skripsi dari sebuah input data sumber. *Levenshtein Distance* merupakan pengukuran dalam bentuk *matrix* dan menghitung jumlah perbedaan dua *string* dengan melalui operasi penambahan (*insert*), penghapusan (*delete*), atau penggantian karakter (*substitute*) pada suatu karakter. *Levenshtein Distance* merupakan salah satu algoritma yang dapat digunakan dalam mendeteksi kemiripan antara dua string yang berpotensi melakukan tindak plagiarisme (Zhan dkk, 2008).

Algoritma *Levenshtein Distance* digunakan untuk memberikan saran atau sugesti judul skripsi yang mendekati dengan kata yang dimasukkan dalam pencarian. Penelitian ini menggunakan antarmuka aplikasi Visual Basic .NET 2012 yang digunakan sebagai aplikasi penguji dari penelitian ini. Pemilihan Visual Basic. NET ini memiliki pertimbangan bahwa aplikasi hanya akan digunakan di PC admin perpustakaan saja dan diharapkan mahasiswa dapat lebih banyak mempelajari khasanah ragam judul skripsi yang sudah pernah dibuat oleh lulusan sebelumnya. Selain itu diharapkan ada interaksi lebih dalam mengenai judul-judul skripsi yang sudah ada dengan petugas perpustakaan.

Penelitian yang dilakukan oleh Widiatry (Widiatry dkk, 2019) dalam membangun, merancang dan menerapkan algoritma *Levenshtein Distance* dalam pencarian judul buku menggunakan Sistem Informasi Perpustakaan pada Fakultas Kedokteran Universitas Palangka Raya, didapat kesimpulan dengan pencarian judul buku *single target* memiliki keakuratan sebesar 75%, untuk pencarian judul buku *multi target* dengan 2 kata memiliki keakuratan sebesar 64,29%, untuk pencarian judul buku *multi target* dengan 3 kata memiliki keakuratan sebesar 66,75%, dan untuk pencarian judul buku *multi target* dengan 4 kata memiliki keakuratan sebesar 70,83%.

Penelitian lain pada analisa kinerja algoritma *Levenshtein Distance* oleh B.P Pratama dkk (Pratama dan Pamungkas, 2016) dalam mendeteksi kemiripan dokumen teks, menambahkan proses *case folding*, *tokenizing*, *stopword removal*, *stemming*, dan *sorting*. Penelitian dilakukan terhadap dua data set dan satu data real. Hasil simulasi didapat bahwa dilakukan proses *sorting* sangat berpengaruh bagi algoritma *Levenshtein Distance*. Hasil

p-ISSN 2407-4489
e-ISSN 2528-7303

terbaik ditunjukan pada proses yang menggunakan *stopword removal, stemming*, dan *sorting* sekaligus yaitu pada data set 1. Hasil lainnya didapat pada proses yang menggunakan penggabungan *stopword* dan *stemming* dengan *sorting* yaitu terbaik pada data set 2. Hasil terbaik pada data *real* didapatkan pada proses yang menggunakan *stemming-sorting*.

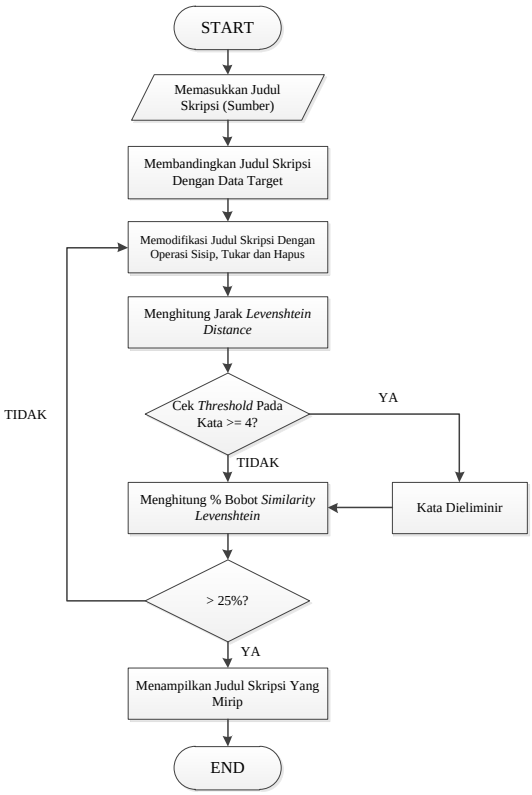
Pada penelitian Ketut Arnawa (Arnawa, 2017) mengenai sistem pencarian judul tugas akhir menggunakan *Levenshtein*. Penelitian tersebut menyimpulkan bahwa algoritma *Levenshtein* dapat membantu mengatasi permasalahan pada kesalahan ejaan kata kunci dengan mekanisme penambahan, penyisipan dan penghapusan karakter.

Penelitian lain berkaitan dengan kecepatan waktu perbandingan dokumen sumber dengan pembanding. Penelitian Febiawan dkk (2019) menunjukkan bahwa proses parsing dokumen berbentuk file pdf ke dalam bentuk string menggunakan *Levenshtein Distance* mempengaruhi kecepatan waktu perhitungan cek *similarity*.

Algoritma *Levenshtein* dapat digunakan untuk menghitung jarak keterbedaan antara dua string (Andhika, 2010) yang merupakan metode pendekatan pencarian string. *Levenshtein Distance* juga dapat berfungsi memberikan rekomendasi kata terdekat ketika user menginputkan kata yang salah atau tidak ada dalam *database* (Sumiari dan Iis, 2019).

METODE PENELITIAN

Penelitian ini akan menggunakan perhitungan *multi target* dengan penentuan nilai *threshold* sebesar 4 kata untuk melakukan pencarian data judul skripsi (pencarian lebih dari satu kata), dimana akan dilakukan perhitungan nilai *distance* per-kata terlebih dahulu, kemudian akan dilanjutkan dengan menghitung bobot *similarity* (%) dari *Levenshtein*. *Threshold* digunakan untuk menentukan kategori ambang batas kemiripan (*similarity*) yang terbagi dua yaitu '*similar*' atau '*not similar'*. Apabila bobot *similarity* melebihi nilai *threshold*, maka dianggap kemiripannya adalah tidak mirip (*not similar*) sehingga kata akan dieliminir. Apabila bobot *similarity* >25% maka data target akan ditampilkan karena dianggap mirip (*similar*). Gambar 1 menunjukkan *Flowcart* perhitungan pada penelitian ini.



Gambar 1. *Flowchart* penelitian

Data target yang digunakan pada penelitian ini menggunakan 96 sampel judul skripsi dengan 6 kategori seperti ditunjukkan pada Tabel 1.

p-ISSN 2407-4489
e-ISSN 2528-7303

Tabel 1. Data target skripsi

No	Kategori	Σ Judul
1	Perancangan Sistem	18
2	Hardware & Mikrokontroller	23
3	Sistem Pakar	12
4	Aplikasi Berbasis Web	18
5	Aplikasi Berbasis Desktop	18
6	Data Mining	7

Hasil akhir data judul skripsi yang ditampilkan adalah judul-judul yang memiliki tingkat persentase *Levenshtein* >25% karena dianggap mirip, seperti yang ditunjukkan pada Tabel 2 berikut ini:

Tabel 2. Persentase kemiripan

No	Range	Keputusan
1	0% - 25%	Tidak Mirip
2	26% - 50%	Mirip
3	51% - 100%	Sangat Mirip

HASIL PENELITIAN DAN PEMBAHASAN

Pada penelitian ini diambil contoh untuk perhitungan *multi target* judul skripsi kategori Perancangan Sistem Informasi yang merupakan data sumber (data yang diinput) dibandingkan dengan data target (data dalam *database*) dari kategori lain sebagai berikut:

Judul skripsi sumber 'Sistem Informasi Penjualan Produk Elektronik Berbasis Web'.

1. Data target dengan kategori Perancangan Sistem Informasi

Sumber	Sistem	Informasi	Penjualan	Produk	Elektronik	Berbasis	Web
Target	Sistem	Informasi	Penjualan	Finish	Good		
Distance	0	0	0	1	1	1	1

$$D(\mathrm{s}+\mathrm{t}) = \sum_{i=0}^{n} d(s_i, t_i)$$

= d(sumber$_1$, target$_1$) + d(sumber$_2$, target$_2$) + d(sumber$_3$, target$_3$) + d(sumber$_4$, target$_4$) + d(sumber$_5$, target$_5$) + d(sumber$_6$, target$_6$) + d(sumber$_7$, target$_7$)

= 0 + 0 + 0 + 1 + 1 + 1 + 1

= 4

Tabel 3. Matriks *Distance* (1)

SUMBER	TARGET						
			sistem	informasi	penjualan	finish	good
		0	1	2	3	4	5
sistem	1	0	1	2	3	4	
informasi	2	1	0	1	2	3	
penjualan	3	2	1	0	1	2	
produk	4	3	2	1	1	2	
elektronik	5	4	3	2	2	2	
berbasis	6	5	4	3	S	3	
web	7	6	5	4	4	4	

Distance bernilai 4 yang ditunjukkan pada Matriks *Distance* (1) di Tabel 3 berada di *range threshold*, maka kata tersebut dieliminasi karena dianggap sebagai kata yang berbeda.

$$B = \left(1 - \frac{d[m,n]}{\max(S,T)}\right) * 100\%$$

$$B = \left(1 - \frac{4}{7}\right) * 100\%$$

$$B = \left(\frac{7}{7} - \frac{4}{7}\right) * 100\%$$

$$B = \left(\frac{3}{7}\right) * 100\% \ = 42{,}85\ \%$$

2. Data target dengan kategori *Hardware* dan *Mikrokontroller*

Sumber	Sistem	Informasi	Penjualan	Produk	Elektronik	Berbasis	Web
Target	Pengontrol	Daya	Lampu	Berbasis	Mikrokontroller	AT89S52	
Distance	1	1	1	1	1	1	1

$$D(\mathrm{s}+\mathrm{t}) = \sum_{i=0}^{n} d(s_i, t_i)$$

= d(sumber$_1$, target$_1$) + d(sumber$_2$, target$_2$) + d(sumber$_3$, target$_3$) + d(sumber$_4$, target$_4$) + d(sumber$_5$, target$_5$) + d(sumber$_6$, target$_6$) + d(sumber$_7$, target$_7$)

= 1 + 1 + 1 + 1 + 1 + 1 + 1

= 7

Tabel 4 menunjukkan Matriks *distance* (2) dimana *Distance* bernilai 7 dan lebih besar dari *range threshold*, maka kata tersebut dieliminasi karena dianggap sebagai kata yang berbeda.

$$B = \left(\frac{0}{7}\right) * 100\% \ = 0\ \%$$

p-ISSN 2407-4489
e-ISSN 2528-7303

Tabel 4. Matriks *Distance* (2)

SUMBER	TARGET						
		pengontrol	daya	lampu	berbasis	mikrokontroller	AT89S52
	0	1	2	3	4	5	6
sistem	1	1	2	3	4	5	6
informasi	2	2	2	3	4	5	6
penjualan	3	3	3	3	4	5	6
produk	4	4	4	4	4	5	6
elektronik	5	5	5	5	5	5	6
berbasis	6	6	6	6	6	6	6
web	7	7	7	7	7	7	7

3. Data target dengan kategori Sistem Pakar

Sumber	Sistem	Informasi	Penjualan	Produk	Elektronik	Berbasis	Web	
Target	Sistem	Pakar	Tes	Kepribadian	Menggunakan	Metode	Forward	Chaining
Distance	0	1	1	1	1	1	1	1

$$D(\mathrm{s}+\mathrm{t}) = \sum_{i=0}^{n} d(s_i, t_i)$$

= d(sumber$_1$, target$_1$) + d(sumber$_2$, target$_2$) + d(sumber$_3$, target$_3$) + d(sumber$_4$, target$_4$) + d(sumber$_5$, target$_5$) + d(sumber$_6$, target$_6$) + d(sumber$_7$, target$_7$) + d(sumber$_8$, target$_8$)
= 0 + 1 + 1 + 1 + 1 + 1 + 1 + 1
= 7

Tabel 5. Matriks *Distance* (3)

SUMBER	TARGET								
		sistem	pakar	tes	kepribadian	menggunakan	metode	forward	chaining
	0	1	2	3	4	5	6	7	8
sistem	1	0	1	2	3	4	5	6	7
informasi	2	1	1	2	3	4	5	6	7
penjualan	3	2	2	2	3	4	5	6	7
produk	4	3	3	3	3	4	5	6	7
elektronik	5	4	4	4	4	4	5	6	7
bebasis	6	5	5	5	5	5	5	6	7
web	7	6	6	6	6	6	6	6	7

Tabel 5 menunjukkan Matriks *distance* (3) dimana *Distance* bernilai 7 dan lebih besar dari *range threshold*, maka kata tersebut dieliminasi karena dianggap sebagai kata yang berbeda.

$B = \left(\frac{1}{8}\right) * 100\% = 12{,}5\ \%$

4. Data target dengan kategori Aplikasi Berbasis Web

Sumber	Sistem	Informasi	Penjualan	Produk	Elektronik	Berbasis	Web
Target	Sistem	Informasi	Ujian	Online	Berbasis	Web	
Distance	0	0	1	1	1	1	1

$$D(\mathrm{s}+\mathrm{t}) = \sum_{i=0}^{n} d(s_i, t_i)$$

= d(sumber$_1$, target$_1$) + d(sumber$_2$, target$_2$) + d(sumber$_3$, target$_3$) + d(sumber$_4$, target$_4$) + d(sumber$_5$, target$_5$) + d(sumber$_6$, target$_6$) + d(sumber$_7$, target$_7$)
= 0 + 0 + 1 + 1 + 1 + 1 + 1
= 5

Tabel 6. Matriks *Distance* (4)

SUMBER	TARGET						
		sistem	informasi	ujian	online	berbasis	web
	0	1	2	3	4	5	6
sistem	1	0	1	2	3	4	5
informasi	2	1	0	1	2	3	4
penjualan	3	2	1	1	2	3	4
produk	4	3	2	2	2	3	4
elektronik	5	4	3	3	3	3	4
bebasis	6	5	4	4	4	4	4
web	7	6	5	5	5	5	5

Tabel 6 menunjukkan Matriks *distance* (4) dimana *Distance* bernilai 5 dan lebih besar dari *range threshold*, maka kata tersebut dieliminasi karena dianggap sebagai kata yang berbeda.

$B = \left(\frac{2}{7}\right) * 100\% = 28{,}5\ \%$

5. Data target dengan kategori Aplikasi Berbasis Desktop

Sumber	Sistem	Informasi	Penjualan	Produk	Elektronik	Berbasis	Web
Target	Perancangan	Sistem	Informasi	Service	Kendaraan	Berbasis	VB .NET
Distance	1	1	1	1	1	0	1

$$D(\mathrm{s}+\mathrm{t}) = \sum_{i=0}^{n} d(s_i, t_i)$$

= d(sumber$_1$, target$_1$) + d(sumber$_2$, target$_2$) + d(sumber$_3$, target$_3$) + d(sumber$_4$, target$_4$) + d(sumber$_5$, target$_5$) + d(sumber$_6$, target$_6$) + d(sumber$_7$, target$_7$)
= 1 + 1 + 1 + 1 + 1 + 0 + 1
= 6

Tabel 7 menunjukkan Matriks *distance* (5) dimana *Distance* bernilai 6 dan lebih besar dari *range threshold*, maka kata tersebut dieliminasi karena dianggap sebagai kata yang berbeda.

$B = \left(\frac{1}{7}\right) * 100\% = 14{,}28\ \%$

p-ISSN 2407-4489
e-ISSN 2528-7303

Tabel 7. Matriks *Distance* (5)

SUMBER	TARGET							
		perancangan	sistem	informasi	service	kendaraan	berbasis	VB .NET
	0	1	2	3	4	5	6	7
sistem	1	1	2	3	4	5	6	7
informasi	2	2	2	3	4	5	6	7
penjualan	3	3	3	3	4	5	6	7
produk	4	4	4	4	4	5	6	7
elektronik	5	5	5	5	5	5	6	7
berbasis	6	6	6	6	6	6	5	6
web	7	7	7	7	7	7	6	6

6. Data target dengan kategori Data Mining

Sumber	Sistem	Informasi	Penjualan	Produk	Elektronik	Berbasis	Web		
Target	Rancang	Data	Mining	Spare	Part	Pada	PT	Asmo	Indonesia
Distance	1	1	1	1	1	1	1	1	1

$$D(\mathrm{s}+\mathrm{t}) = \sum_{i=0}^{n} d(s_i, t_i)$$

= d(sumber$_1$, target$_1$) + d(sumber$_2$, target$_2$) + d(sumber$_3$, target$_3$) + d(sumber$_4$, target$_4$) + d(sumber$_5$, target$_5$) + d(sumber$_6$, target$_6$) + d(sumber$_7$, target$_7$) + d(sumber$_8$, target$_8$)

= 1 + 1 + 1 + 1 + 1 + 1 + 1 +1 + 1

= 9

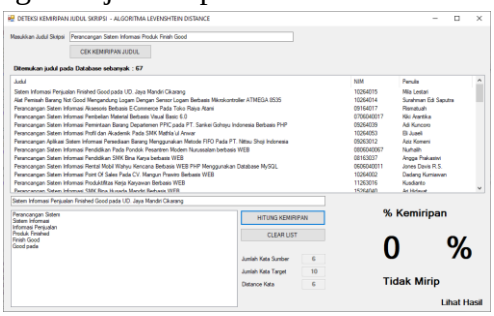
Tabel 8. Matriks *Distance* (6)

SUMBER	TARGET									
		rancang	data	mining	spare	part	pada	pt	asmo	indonesia
	0	1	2	3	4	5	6	7	8	9
sistem	1	1	2	3	4	5	6	7	8	9
informasi	2	2	2	3	4	5	6	7	8	9
penjualan	3	3	3	3	4	5	6	7	8	9
produk	4	4	4	4	4	5	6	7	8	9
elektronik	5	5	5	5	5	5	6	7	8	9
berbasis	6	6	6	6	6	6	6	7	8	9
web	7	7	7	7	7	7	7	7	8	9

Tabel 8 menunjukkan Matriks *distance* (6) dimana *Distance* bernilai 9 dan lebih besar dari *range threshold*, maka kata tersebut dieliminasi karena dianggap sebagai kata yang berbeda.

$$B = \left(\frac{0}{9}\right) * 100\% = 0\ \%$$

Pengujian algoritma ini menggunakan aplikasi Visual Basic .NET 2012 seperti yang ditunjukkan pada Gambar 2.



Gambar 2. Pengujian VB .NET

KESIMPULAN DAN SARAN

Dari hasil sampel pengajuan sebuah judul skripsi untuk perhitungan *multi target* dengan pengujian antara data sumber dengan 6 kategori data target dalam *database* dan penentuan *threshold* >= 4 serta penentuan bobot *similarity* >25% dinyatakan mirip untuk ditampilkan, maka disimpulkan data yang ditampilkan adalah dari kategori judul perancangan sistem dengan *distance* 4 dan kemiripan 42,85% dan dari kategori aplikasi berbasis web dengan *distance* 5 dan kemiripan 28,5%. Sedangkan 4 kategori lain memiliki *distance* lebih dari 5 dan persen bobot *similarity* kurang dari 25% sehingga tidak dianggap mirip dan tidak ditampilkan.

Untuk mendapatkan hasil lebih akurat, disarankan ada penelitian terkait untuk melanjutkan penelitian ini dengan pemberian nilai *threshold* lebih rendah dan penentuan bobot *similarity* lebih besar. Dapat juga dikembangkan dengan penambahan proses *stemming* untuk membuang kata-kata yang tidak diperlukan.

DAFTAR PUSTAKA

Andhika, Fatardhi Rizky. (2010). *Penerapan String Suggestion dengan Algoritma Levenshtein Distance dan Alternatif Algoritma Lain dalam Aplikasi*. Bandung: Institut Teknologi Bandung.

Arnawa, I.B.K.S. (2017). *Implementasi Algoritma Levenshtein Pada Sistem Pencarian Judul Skripsi/Tugas Akhir*. Jurnal Sistem Dan Informatika, STIKOM Bali.

Febiawan, M.H., Setiawan, Agus., dan Primadewi, Ardhin. (2019). *Sistem Pendeteksi Dini Plagiarisme*

p-ISSN 2407-4489
e-ISSN 2528-7303

Menggunakan Algoritma Levenshtein Distance. Jurnal Komtika (Komputasi dan Informatika), Vol. 3 No. 1 | Mei 2019.

Pratama, B.P dan Pamungkas, S.A. (2016). *Analisis Kinerja Algoritma Levenshtein Distance Dalam Mendeteksi Kemiripan Dokumen Teks*. Jurnal LOGIK@, Jilid 6, No. 2, ISSN 1978 – 8568, Hal. 131 – 143.

Sumiari, Ni Kadek., dan Iis, Ni Ketut Dewi Ari Jayanti. (2019). *Optimasi Dashboard Information System STIKOM Bali dengan Algoritma Levenshtein Distance*. Universitas AMIKOM Yogyakarta, ISSN 2354-5771.

Widiatry, W., Sari, N., Pranatawijaya, V., dan Adidyana Anugrah Putra, P. (2019). *Penerapan Algoritma Levenshtein Distance Untuk Pencarian Pada Sistem Informasi Perpustakaan Fakultas Kedokteran Universitas Palangka Raya*. Jurnal SAINTEKOM, 9(1), 66-82. doi:10.33020/saintekom.v9i1.75.

Zhan, Su, Byung-Ryul Ahn, Ki-Yol Eom, Min-Koo Kang, Jin-Pyung Kim, dan Moon-Kyun Kim. (2008). *Plagiarism Detection Using the Levenshtein Distance and Smith-Waterman Algorithm*. The 3rd International Conference on Innovative Computing, Information and Control (ICICIC), DOI: 10.1109/ICICIC.2008.4