

Implementasi *Levenshtein Distance* Pada Aplikasi Pencarian Barang Di Berbagai *E-Marketplace* Menggunakan Teknik *Web Scraping*

Ahmad Khairun Arsyad^{*1}, Bambang Pramono², Isnawaty³, Muhammad Yamin⁴, Ihsan Sarita⁵

^{1,2,3,4,5}Program Studi Teknik Informatika, Fakultas Teknik dan Universitas Halu Oleo
e-mail: ^{*1}irunwazed@gmail.com, ²bambangpramono09@gmail.com, ³isna.1711@gmail.com, ⁴putra0683@gmail.com, ⁵ihsansarita@yahoo.co.id

Abstrak

Pola pembelian barang masyarakat dicendrung oleh harga yang murah dan kualitas yang baik. Dengan banyaknya berbagai macam e-marketplace saat ini akan mengakibatkan tumpang tindih harga terhadap barang promosi yang sama pada e-marketplace yang berbeda. Hal ini akan membuat konsumen tidak efisien dalam membandingkan barang yang akan dibeli mengingat barang tersebut juga terdapat pada e-marketplace lainnya yang juga mungkin memiliki harga lebih rendah. Berdasarkan hal ini maka dibuatlah aplikasi search engine yang dapat melakukan pencarian terhadap barang yang sama di beberapa e-marketplace.

Untuk membuat aplikasi ini maka diimplementasikan sebuah metode pencarian string yang memadukan data dari beberapa e-marketplace yang akan dipasang pada search engine. Salah satu metode yang tepat adalah Levenshtein Distance dengan teknik web scraping untuk pengumpulan data. Hasil dari penelitian ini menunjukkan bahwa kesesuaian kata pada search engine dan output hasil pencarian memiliki kesesuaian yang baik bila nilai kesesuaian berada diatas 0.67 dalam jarak 0.0-1.0.

Kata kunci— E-marketplace, Search Engine, Levenshtein Distance.

Abstract

The pattern of purchasing public goods is tended by low prices and good quality. With so many different types of emarketplaces today it will cause overlapping prices of the same promotional items in different e-marketplaces. This will make consumers inefficient in comparing the items to be purchased considering that the goods are also found in other e-marketplaces which may also have lower prices. Based on this, a search engine application is made that can search for the same item in several e-marketplaces.

To make this application a string search method is implemented that combines data from several e-marketplaces that will be installed on search engines. One appropriate method is Levenshtein Distance with web scraping techniques for data collection. The results of this study indicate that the suitability of words on search engines and the output of search results have a good suitability if the suitability value is above 0.67 in the range 0.0-1.0.

Keywords— E-marketplace, Search Engine, Levenshtein Distance.

1. PENDAHULUAN

Perkembangan teknologi informasi yang sangat meningkat saat ini, mengakibatkan perkembangan pengguna internet di Indonesia tumbuh semakin meningkat. Salah satu perubahan yang besar yaitu pada aspek ekonomi. aspek ekonomi merupakan aspek yang sangat dipengaruhi oleh peningkatan penggunaan internet. Dengan banyaknya penduduk Indonesia yang menggunakan internet, membuat model pertumbuhan ekonomi yang semula masih konvensional kini semakin mengarah kepada perekonomian yang berbasis atau menggunakan fasilitas internet. Salah satu bentuk pemanfaatan internet terhadap aktifitas perekonomian adalah toko *online* (*online shop*). *Online shop* atau yang biasa disebut *olshop* hadir ditengah masyarakat didasarkan dari kecendrungan masyarakat dalam pemanfaatan internet pada aspek kehidupan. Bahkan saat ini hampir semua bentuk kegiatan jual beli dapat dilakukan hanya dengan menggunakan perangkat elektronik dan internet. Sehingga saat ini transaksi jual beli konvensional mulai tergeser dengan transaksi jual beli modern, maka muncullah istilah perdagangan elektronik yang biasa disebut *electronic commerce* (*e-commerce*).

E-commerce sangat menguntungkan bagi produsen dan juga sangat memudahkan bagi konsumen, karena sistem pelayanan melalui internet sangat efisien, praktis dan dapat menghemat waktu untuk memasarkan produk atau jasa. Kemudahan ini menyebabkan hadirnya berbagai macam toko yang tidak memiliki bentuk fisik yaitu toko-toko maya di dunia internet atau yang dewasa ini disebut *electronic marketplace* (*e-marketplace*).

E-marketplace yang kian merebak di dunia maya membuat persaingan untuk mendapatkan pengunjung menjadi meningkat. Beberapa *e-marketplace* mengandalkan potongan harga atau menawarkan barang dagangan dengan harga yang murah untuk menarik minat pengunjung. Namun dengan kian banyaknya *e-marketplace* yang ada saat ini membuat pengunjung kesulitan dalam menentukan atau membandingkan suatu barang yang sama pada *e-marketplace* yang berbeda.

Agar memudahkan pencarian barang pada sebuah *e-marketplace*, biasanya terdapat sistem pencarian atau mesin pencarian pada laman *website e-marketplace* yang dapat memfilter data sesuai dengan kata kunci yang dimasukkan oleh pengunjung. Penggunaan mesin pencarian dianggap lebih efisien dan cepat dibandingkan dengan memberikan menu *list* kepada pengunjung untuk menentukan barang yang dituju. Namun untuk memadukan data dari berbagai *website e-marketplace* yang ada, dapat dilakukan dengan menggunakan teknik *Web Scraping*. *Web Scraping* sendiri adalah pengambilan sebuah dokumen semi terstruktur dari berbagai sumber, yang umumnya berupa halaman-halaman web dalam bahasa markup, yang kemudian dianalisis untuk mengambil data tertentu. Dengan adanya sistem pencarian menggunakan teknik *web scraping*, *search engine* dapat melakukan pencarian *string input*-an terhadap beberapa *website* sekaligus.

Dalam menyempurnakan sebuah sistem pencari yang dapat memuat data dari berbagai sumber dibutuhkan sebuah algoritma yang dapat menganalisis *string*. *Levenshtein Distance* merupakan sebuah algoritma yang dapat mengukur kesamaan antara *string*. Penggabungan antara teknik *Scraping* dan algoritma *Levenshtein Distance* ini yang nantinya akan membantu pengunjung dalam menemukan dan membandingkan barang yang diinginkan.

Ariyani, Sutardi dan Ramadhan (2016) menyimpulkan perhitungan *Levenshtein Distance* menggunakan data real yaitu data dokumen berplagiat yang diambil dari artikel atau berita menghasilkan nilai similarity yang tinggi yaitu diatas 85 % sampai 100 % untuk dokumen yang tingkat kemiripannya tinggi. Sedangkan untuk dokumen dengan tingkat kemiripan yang rendah atau tidak berplagiat maka menghasilkan nilai similarity dibawah 40% [1].

Pratama dan Pamungkas (2016) menyimpulkan bahwa algoritma *Levenshtein Distance* dapat menghitung jumlah perbedaan antara dua kata. Untuk menghitung antar dokumen teks dengan cara pembentukan suatu matriks kata untuk mendapatkan nilai jaraknya, kemudian melakukan perhitungan bobot kemiripan antar dokumen teks berdasarkan nilai jarak tersebut [2].

Panjaitan (2018) menyimpulkan bahwa dengan melakukan implementasi *Web Scraping* dapat mengumpulkan data melalui *website* dengan cara ekstraksi elemen tag website menjadi data yang dapat disimpan ke dalam database, sehingga dapat membentuk sebuah model pengumpulan data cuaca yang termonitoring dengan baik dan efisien. Identifikasi keadaan cuaca menjadi lebih akurat dengan memanfaatkan data yang besar dari hasil scraping dengan dikombinasikan dengan *K-Nearest Neighbor*. Sehingga menghasilkan akurasi identifikasi sebesar 98% [3].

Syaifudin, Syafiandini dan Prasadana (2018) menyimpulkan bahwa web scraping dapat digunakan sebagai teknik pengambilan data harga, data rating, nama dan gambar barang. Data-data tersebut kemudian dapat dimasukkan kedalam website agar diolah berdasarkan kategori yang sesuai [4].

Melihat permasalahan diatas, maka peneliti mengembangkan sebuah sistem yang mampu melakukan pencarian barang terhadap beberapa *e-marketplace* yang ada, agar dapat melakukan perbandingan atas barang yang sama terhadap *e-marketplace* yang berbeda. Atas dasar inilah peneliti mengangkat judul “Implementasi *Levenshtein Distance* pada Aplikasi Pencarian Barang di Berbagai *E-Marketplace* Menggunakan Teknik *Web Scraping*”.

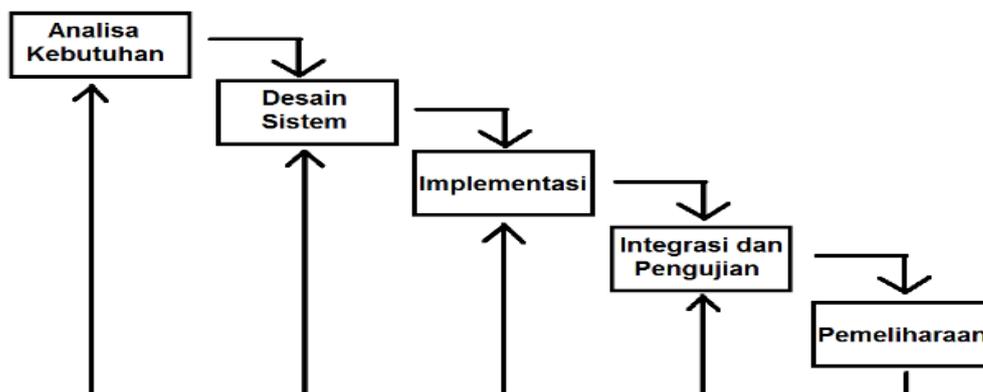
2. METODE PENELITIAN

2.1 Metode Pengumpulan Data

Tahap ini digunakan pada perancangan aplikasi menggunakan studi literatur dan pengambilan data barang dari *e-marketplace* Tokopedia, Bukalapak dan FJB Kaskus menggunakan teknik *web scraping*. *Web scrapping* berhubungan dengan pengindeksan *web* yang merupakan suatu teknik umum yang dipakai hampir semua *search engine*. Perbedaannya *web scraping* lebih berfokus pada transformasi dari suatu web yang tidak terstruktur, umumnya dalam format HTML menjadi suatu format data terstruktur yang dapat disimpan dan dianalisa pada database atau lembar kerja [5]. Metode studi literatur ini dilaksanakan dengan melakukan studi kepustakaan yang relevan. Metode ini dilakukan untuk mencari penelitian yang sejenis dengan penelitian ini.

2.2 Metode Pengembangan Sistem

Tahap ini menggunakan metode *waterfall*. Metode *waterfall* melakukan pendekatan alur hidup perangkat lunak secara berurutan mulai dari analisis kebutuhan mengenai mencari permasalahan yang terjadi yaitu meningkatnya pengguna internet di Indonesia hingga mencapai 8% [6], desain sistem meliputi tentang skema pembuatan sistem, pengkodean menggunakan bahasa pemrograman python dikarenakan dalam *permormance load browser* dan *web server* lebih cepat dalam merespon permintaan *client* dibandingkan dengan bahasa pemrograman php [7], pengujian sistem, dan tahapan pendukung. Berikut adalah model gambar metode *waterfall*.

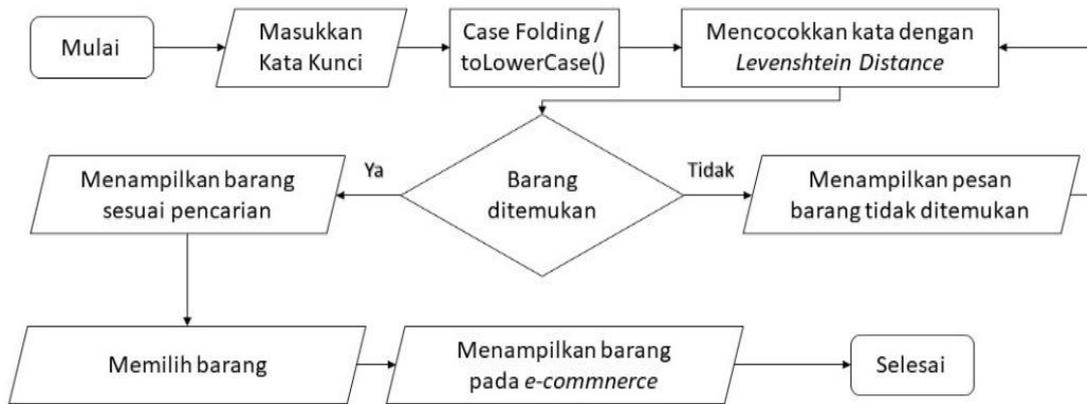


Gambar 1 Tahapan Metode Pengembangan Sistem

2.3 Flowchart Sistem

Adapun alur kerja dari flowchart diagram sistem pencarian adalah sebagai berikut:

- User memasukkan kata yang ingin dicari.
- Kata yang dimasukkan akan melalui proses perubahan kata menjadi huruf kecil.
- Kata yang dicari akan dicocokkan menggunakan algoritma *Levenshtein Distance*.
- Menampilkan daftar produk yang berhubungan dengan kata kunci.
- Jika produk tidak ditemukan, maka menampilkan pesan “Produk tidak ditemukan” dan user dapat melakukan pencarian ulang. Tapi jika barang ditemukan, maka akan menampilkan produk.
- User memilih produk yang diinginkan.
- Menampilkan halaman penjualan produk yang telah dipilih.



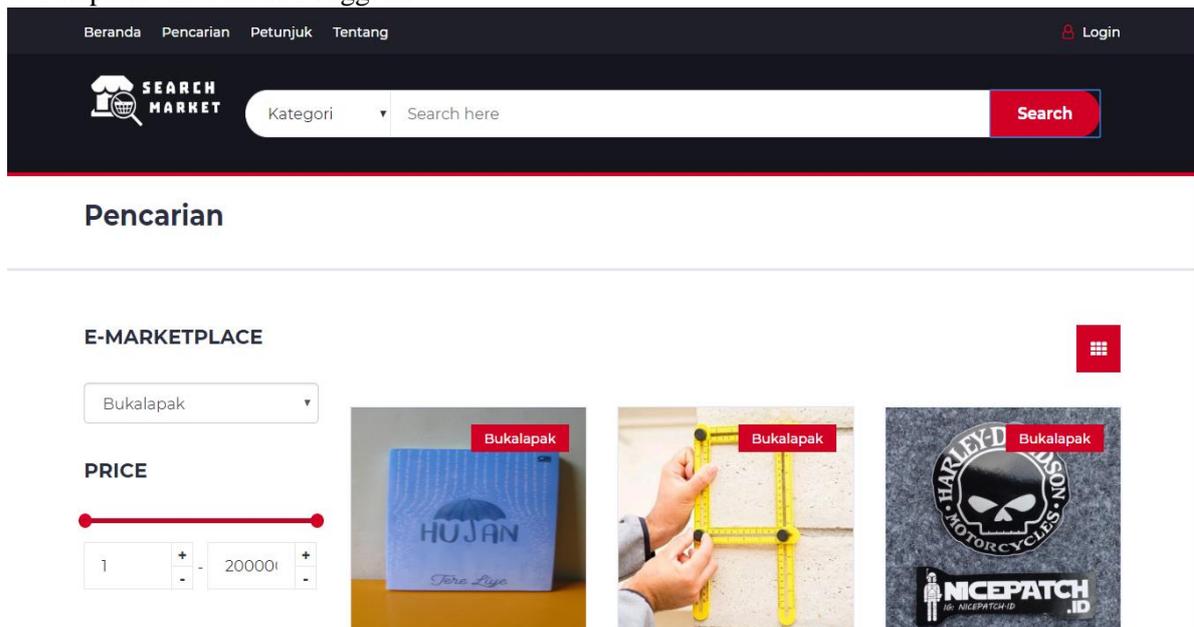
Gambar 2 Flowchart Sistem Pencarian Barang

3. HASIL DAN PEMBAHASAN

3.1 Implementasi Tampilan Antarmuka Sistem

Aplikasi pencarian barang diimplementasikan menjadi aplikasi berbasis *web*.

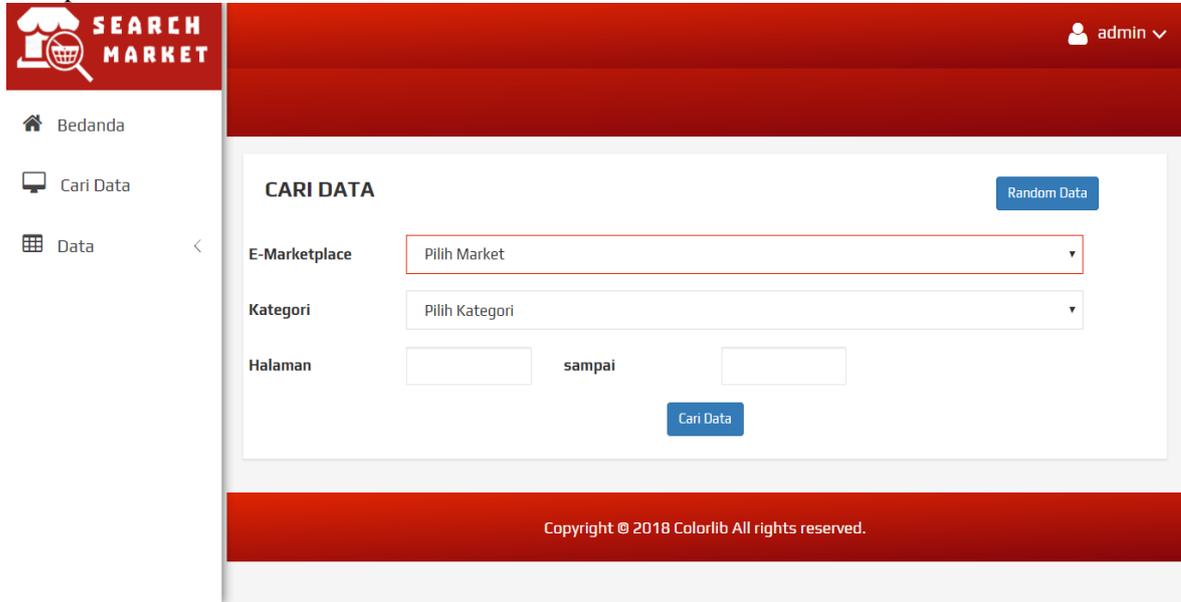
a. Tampilan Antarmuka Pengguna



Gambar 3 Tampilan Antarmuka Pengguna

Pada gambar 3 terdapat tampilan saat pengguna menggunakan sistem untuk melakukan pencarian barang yang diinginkan dengan memasukkan kata kunci beserta filter yang ada. Filter yang dapat dilakukan yaitu filter harga, karegori dan sumber *e-marketplace*.

b. Tampilan Antarmuka Admin



Gambar 4 Tampilan Antarmuka Admin

Pada gambar 4 terdapat tampilan menu cari data berfungsi untuk mencari barang menggunakan teknik *web scraping* dari *e-marketplace* yang dipilih untuk menambahkannya ke dalam basis data *system*.

3.2 Pengujian Sistem

3.2.1 Pengujian *black box*

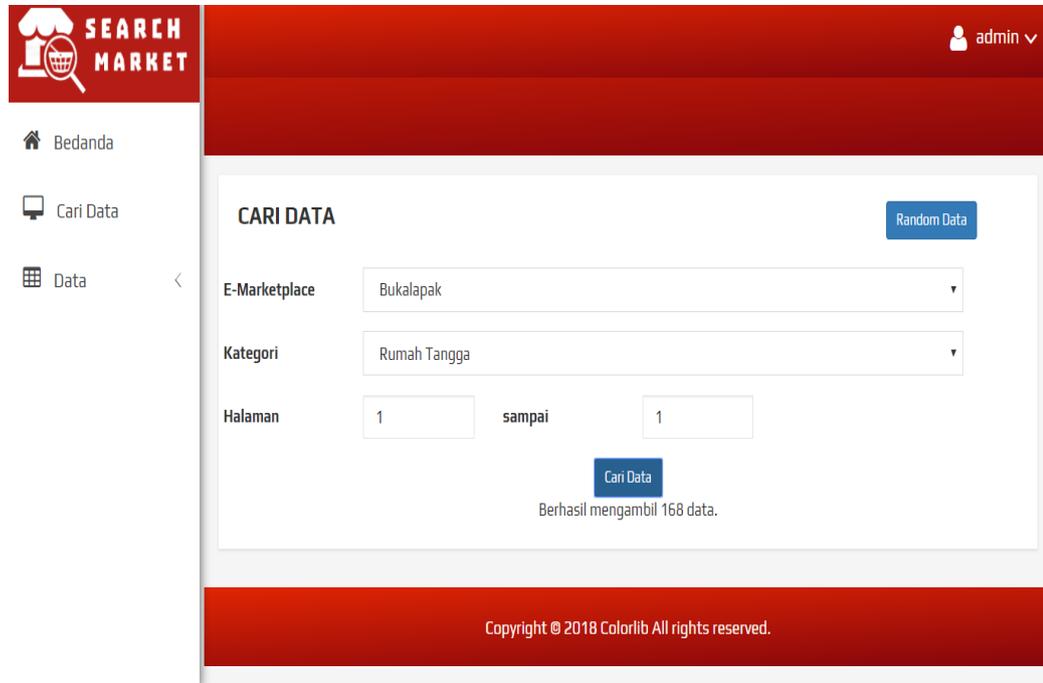
Pengujian *black box* dilakukan dengan menguji perangkat lunak dari segi fungsionalitas perangkat lunak. Fungsionalitas perangkat lunak yang diuji sesuai dengan use case pada tahap desain. Peneliti membagi pengujian menjadi lima bagian. Setiap bagian diuji sesuai dengan skenario use case pada tahap desain.

Tabel 1 Pengujian *Black Box*

<i>Input / Event</i>	<i>Proses</i>	<i>Output</i>	Hasil Uji
Membuka Website	Proses masuk pada aplikasi pencarian barang	Menampilkan Menu Beranda	Sesuai
Memilih Menu Pencarian	Menampilkan data barang	Data Barang ditampilkan	Sesuai
Memilih Menu Petunjuk	Menampilkan petunjuk penggunaan aplikasi	Petunjuk penggunaan aplikasi ditampilkan	Sesuai
Memilih Menu Tentang	Menampilkan Menu Tentang	Menampilkan tentang aplikasi	Sesuai
Pencarian Kata	Memasukkan kata yang ingin dicari dan menekan tombol <i>search</i>	Menampilkan hasil pencarian	Sesuai

3.2.2 Pengujian *web scraping* pada Bukalapak

Pengujian ini dilakukan untuk memeriksa apakah sistem berhasil untuk melakukan pengambilan data pada *website* Bukalapak. Pengujian ini dilakukan pada *website* bukhalapak pada kategori rumah tangga pada halaman satu terdapat 168 data.



Gambar 5 Pengujian *Web Scraping* pada Bukalapak

3.2.3 Pengujian pencarian menggunakan *Levenshtein Distance*

Pengujian ini dilakukan untuk memeriksa apakah sistem berhasil melakukan pencarian menggunakan algoritma *Levenshtein Distance* dengan kata kunci yang dimasukkan oleh user. Pencarian ini dilakukan dengan cara user memasukkan kata kunci dan menekan tombol *search*, setelah itu sistem akan menghitung nilai kedekatan antara kata kunci dan semua kata yang ada pada database. Kata yang memiliki nilai kedekatan lebih besar dari nilai batasan *levenshtein distance* akan ditampilkan sebagai hasil dari pencarian yang dilakukan oleh user.

Tabel 2 Pengujian Pencarian

No	Kata Pencarian	Judul Barang	Nilai
1	xiaomi	Jual REDMI 4X Tempered Glass <u>Xiaomi</u> Redmi 4 X	1.0
3	xiaomi	Jual Sarung Tangan Touring di lapak X M N <u>xiamen</u>	0.5
4	xiaomi	Indomie MI GORENG 1 PCS <u>Indomi</u> Mie Instan	0.5

Berdasarkan tabel diatas, berhasil menampilkan barang yang akan digunakan pada percobaan selanjutnya dengan memilih kata yang mempengaruhi nilai *levenshtein distance* pada judul barang. Kata yang akan digunakan yaitu *xiaomi*, *xiamen* dan *indomi*. Kata-kata ini akan dibandingkan dengan kata *xiaomi* agar mendapatkan nilai rata-rata yang akan dijadikan sebagai nilai batasan *Levenshtein Distance*. *Levenshtein Distance* adalah sebuah matriks *string* yang digunakan untuk mengukur perbedaan atau jarak (*distance*) antara dua *string*. Nilai *distance* antara dua *string* ini ditentukan oleh jumlah minimum dari operasi-operasi perubahan yang diperlukan untuk melakukan transformasi dari suatu *string* menjadi *string* lainnya. Operasi-operasi tersebut adalah penyisipan (*insertion*), penghapusan (*deletion*), atau penukaran

(substitution). *Levenshtein Distance* merupakan salah satu algoritma yang dapat digunakan dalam mendeteksi kemiripan antara dua *string* yang berpotensi melakukan tindak plagiarisme. Berikut rumus dan perhitungan dari Levenshtein Distance:

$$lev_{a,b}(i,j) = \begin{cases} \max(i,j) \\ \min \begin{cases} lev_{a,b}(i-1,j) + 1 \\ lev_{a,b}(i,j-1) + 1 \\ lev_{a,b}(i-1,j-1) + 1_{(ai \neq bj)} \end{cases} \end{cases} \quad (1)$$

Tabel 3 Pengujian Pertama

	-	x	i	a	o	m	i
-	0	1	2	3	4	5	6
x	1	0	1	2	3	4	5
i	2	1	0	1	2	3	4
a	3	2	1	0	1	2	3
o	4	3	2	1	0	1	2
m	5	4	3	2	1	0	1
i	6	5	4	3	2	1	0

Tabel 4 Pengujian Kedua

	-	x	i	a	o	m	i
-	0	1	2	3	4	5	6
x	1	0	1	2	3	4	5
i	2	1	0	1	2	3	4
a	3	2	1	0	1	2	3
m	4	3	2	1	1	1	2
e	5	4	3	2	2	2	2
n	6	5	4	3	3	3	3

Tabel 5 Pengujian Ketiga

	-	x	i	a	o	m	i
-	0	1	2	3	4	5	6
i	1	1	1	2	3	4	5
n	2	2	2	2	3	4	5
d	3	3	3	3	3	4	5
o	4	4	4	4	3	4	5
m	5	5	5	5	4	3	4
i	6	6	5	6	5	4	3

Berdasarkan Tabel 3, 4 dan 5 terdapat hasil nilai *Levenshtein Distance* y dengan nilai 0, 3 dan 3. Kemudian nilai tersebut diubah menjadi nilai *similarity* dengan rumus:

$$Sim = 1 - \frac{\text{nilai LD}}{(\text{kata1,kata2})} \quad (2)$$

Dari rumus itu ditemukan nilai *similarity* dari Tabel 1 adalah 1 dengan jumlah kata terbanyak yaitu 6 (xiaomi). Kemudian nilai *similarity* dari Tabel 2 adalah 0.5 dengan jumlah kate terbanyak adalah 6 (xiaomen). Dan yang terakhir nilai *similarity* Table 3 adalah 0.5 dengan jumlah kata terbanyak adalah 6 (indomi). Rata-rata dari ketiga nilai *similarity* tersebut menjadi nilai batasan atau kesesuaian pencarian *Levenshtein Distance* yang bernilai 0.67.

4. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan mengenai Implementasi *Levenshtein Distance* pada Aplikasi Pencarian Barang di Berbagai *E-Marketplace* Menggunakan Teknik *Web Scraping*, maka diperoleh kesimpulan yaitu:

1. Metode *Web Scraping* yang digunakan untuk mendapatkan data barang pada Bukalapak, Tokopedia dan FJB Kaskus yaitu dengan cara mem-parsing file HTML tersebut. Setelah itu, dilakukan proses ekstraksi data yang diinginkan dengan menggunakan *library* beautifulsoup pada Bahasa pemrograman python.
2. Algoritma *Levenshtein Distance* dapat diterapkan dengan baik dalam proses pencarian barang dengan nilai kesesuaian kata pada *search engine* dan output hasil pencarian memiliki nilai kesesuaian yang baik bila nilai kesesuaian berada diatas 0.67 dalam jarak 0.0-1.0.

5. SARAN

Adapun beberapa saran untuk pengembangan lebih lanjut dari Implementasi *Levenshtein Distance* pada Aplikasi Pencarian Barang di Berbagai *E-Marketplace* Menggunakan Teknik *Web Scraping*, yaitu:

1. Sistem diharapkan dapat dikembangkan lagi dengan menambahkan jumlah *e-marketplace* yang terdaftar serta penambahan fitur seperti filter kategori barang terlaris dan filter kategori barang diskon.
2. Sistem diharapkan mampu menangani *error* yang terjadi dalam proses *web scraping* dengan lebih cerdas terutama penanganan *error* jika terjadi perubahan tampilan dari *website* yang di-*scraping*.

DAFTAR PUSTAKA

- [1] N. H. Ariyani, Sutardi, and R. Ramadhan, "Aplikasi Pendeteksi Kemiripan Isi Teks Dokumen Menggunakan Metode Levenshtein Distance," *semanTIK*, vol. Vol 2, no. 1, pp. 279–286, 2016.
- [2] B. P. Pratama and S. A. Pamungkas, "Analisis Kinerja Algoritma Levenshtein Distance dalam Mendeteksi Kemiripan Dokumen Teks," *J. "LOG!K@,"* vol. 6, no. 2, pp. 131–143, 2016.
- [3] P. M. Panjaitan, "Sistem Monitoring Cuaca dan Identifikasi Keadaan Cuaca Menggunakan Teknik Web Scraping," UNIVERSITAS SUMATERA UTARA, 2018.
- [4] Y. W. Syaifudin, A. Ferti Syafiandini, and H. R. Prasadana, "Aplikasi Pencarian Penjualan Laptop Menggunakan Teknologi Web Scraping," *Jip*, vol. 4, no. 4, pp. 246–251, 2018.
- [5] M. S. Utomo, "Web Scraping pada Situs Wikipedia menggunakan Metode Ekspresi Regular," *J. Teknol. Inf. Din.*, vol. 18, no. 2, pp. 153–160, 2013.
- [6] Tim APJII, "Potret Zaman Now Pengguna dan Perilaku Internet Indonesia," *BULEtin APJII*, pp. 1–7, 2018.
- [7] P. Winardi, M. I. Abdillah, D. Hermanto, and E. P. Widiyanto, "Analisis Perbandingan Performa Web Server Berbasis PHP dan Web Server Berbasis Python Twisted," no. x, 2018.