IMPLEMENTASI ALGORITMA LEVENSHTEIN DISTANCE DAN ALGORITMA BOYER-MOORE DALAM PENGELOMPOKAN WORK ORDER BERDASARKAN PRODUK DI PT. XYZ

TUGAS AKHIR

Cakra Wijaya Saputra
41518110052

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MERCU BUANA
JAKARTA
2022

IMPLEMENTASI ALGORITMA LEVENSHTEIN DISTANCE DAN ALGORITMA BOYER-MOORE DALAM PENGELOMPOKAN WORK ORDER BERDASARKAN PRODUK (STUDI KASUS: PT. XYZ)

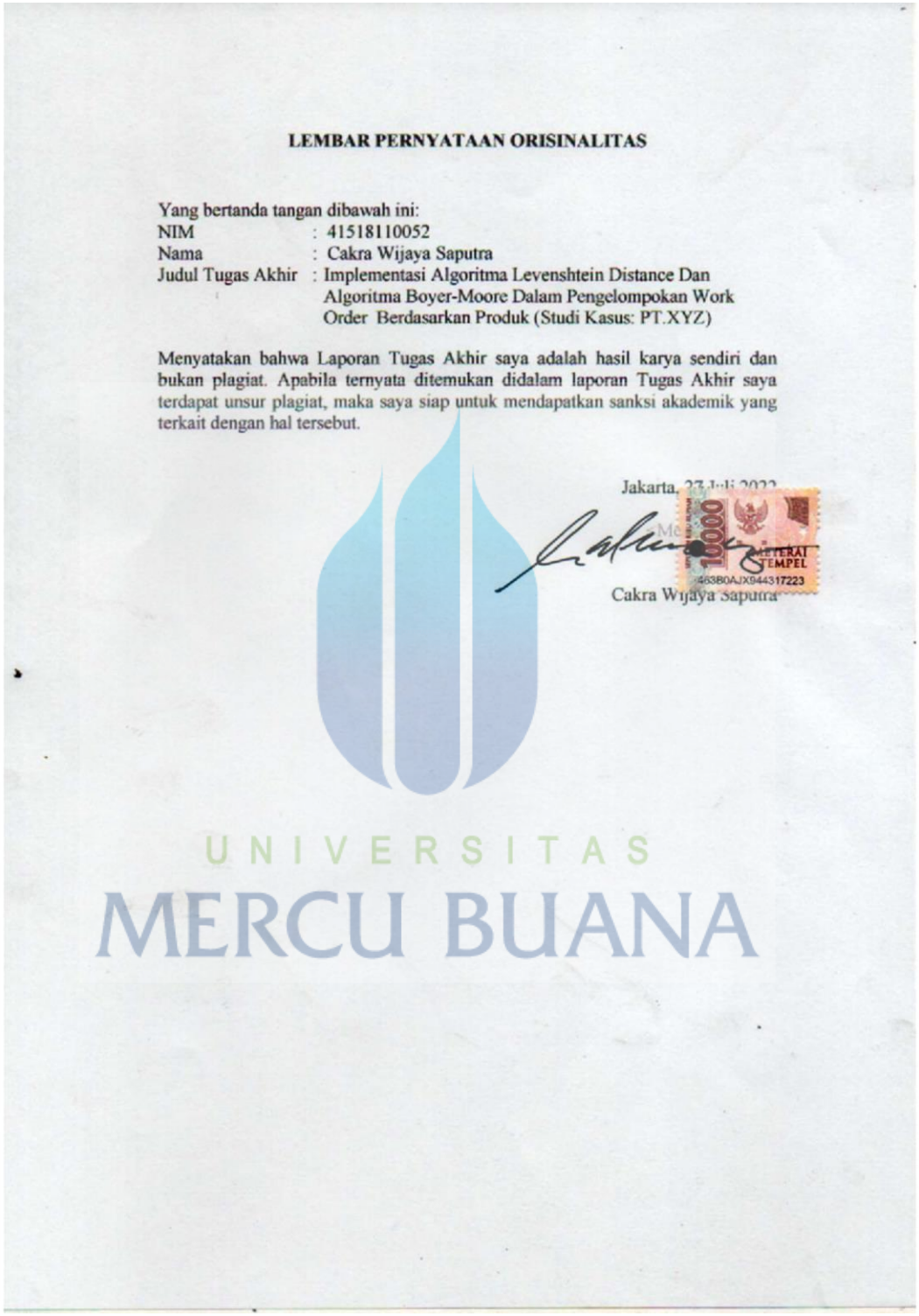
Tugas Akhir

Diajukan Untuk Melengkapi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer

Oleh:
Cakra Wijaya Saputra
41518110052

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MERCU BUANA
JAKARTA
2022

LEMBAR PERNYATAAN ORISINALITAS



LEMBAR PERNYATAAN ORISINALITAS

Yang bertanda tangan dibawah ini:
NIM : 41518110052
Nama : Cakra Wijaya Saputra
Judul Tugas Akhir : Implementasi Algoritma Levenshtein Distance Dan Algoritma Boyer-Moore Dalam Pengelompokan Work Order Berdasarkan Produk (Studi Kasus: PT.XYZ)

Menyatakan bahwa Laporan Tugas Akhir saya adalah hasil karya sendiri dan bukan plagiat. Apabila ternyata ditemukan didalam laporan Tugas Akhir saya terdapat unsur plagiat, maka saya siap untuk mendapatkan sanksi akademik yang terkait dengan hal tersebut.

Jakarta, 27 Juli 2022

Cakra Wijaya Saputra

SURAT PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR

SURAT PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR

Sebagai mahasiswa Universitas Mercu Buana, saya yang bertanda tangan di bawah ini :

Nama Mahasiswa	:	Cakra Wijaya Saputra
NIM	:	41518110052
Judul Tugas Akhir	:	Implementasi Algoritma Levenshtein Distance Dan Algoritma Boyer-Moore Dalam Pengelompokan Work Order Berdasarkan Produk (Studi Kasus: PT.XYZ)

Dengan ini memberikan izin dan menyetujui untuk memberikan kepada Universitas Mercu Buana **Hak Bebas Royalti Noneksklusif** (*None-exclusive Royalty Free Right*) atas karya ilmiah saya yang berjudul diatas beserta perangkat yang ada (jika diperlukan).

Dengan Hak Bebas Royalti/Noneksklusif ini Universitas Mercu Buana berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat dan mempublikasikan tugas akhir saya.

Selain itu, demi pengembangan ilmu pengetahuan di lingkungan Universitas Mercu Buana, saya memberikan izin kepada Peneliti di Lab Riset Fakultas Ilmu Komputer, Universitas Mercu Buana untuk menggunakan dan mengembangkan hasil riset yang ada dalam tugas akhir untuk kepentingan riset dan publikasi selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Jakarta, 27 Juli 2022

Cakra Wijaya Saputra

SURAT PERNYATAAN LUARAN TUGAS AKHIR

SURAT PERNYATAAN LUARAN TUGAS AKHIR

Sebagai mahasiswa Universitas Mercu Buana, saya yang bertanda tangan di bawah ini :

Nama Mahasiswa : Cakra Wijaya Saputra
NIM : 41518110052
Judul Tugas Akhir : Implementasi Algoritma Levenshtein Distance Dan Algoritma Boyer-Moore Dalam Pengelompokan Work Order Berdasarkan Produk Di PT. XYZ

Menyatakan bahwa :

1. Luaran Tugas Akhir saya adalah sebagai berikut :

No	Luaran	Jenis		Status	
1	Publikasi Ilmiah	Jurnal Nasional Tidak Terakreditasi		Diajukan	x
		Jurnal Nasional Terakreditasi	x		
		Jurnal International Tidak Bereputasi		Diterima	
		Jurnal International Bereputasi			
	Disubmit/dipublikasikan di :	Nama Jurnal	: Jurnal Nasional Informatika dan Teknologi Jaringan		
		ISSN	: ISSN 2540-7600 (Online) | ISSN 2540-7597 (Cetak)		
		Link Jurnal	: https://jurnal.uisu.ac.id/index.php/infotekjar		
		Link File Jurnal Jika Sudah di Publish	: https://jurnal.uisu.ac.id/index.php/infotekjar/author/submission/5582		

2. Bersedia untuk menyelesaikan seluruh proses publikasi artikel mulai dari submit, revisi artikel sampai dengan dinyatakan dapat diterbitkan pada jurnal yang dituju.
3. Diminta untuk melampirkan scan KTP dan Surat Pernyataan (Lihat Lampiran Dokumen HKI), untuk kepentingan pendaftaran HKI apabila diperlukan

Demikian pernyataan ini saya buat dengan sebenarnya.

Mengetahui
Dosen Pembimbing TA

Jakarta, 27 Juli 2021

METERAI TEMPEL

(Achmad Kodar, Drs. MT)

Cakra Wijaya Saputra

LEMBAR PERSETUJUAN PENGUJI

LEMBAR PERSETUJUAN PENGUJI

NIM	:	41518110052
Nama	:	Cakra Wijaya Saputra
Judul Tugas Akhir	:	Implementasi Algoritma Levenshtein Distance Dan Algoritma Boyer-Moore Dalam Pengelompokan Work Order Berdasarkan Produk (Studi Kasus: PT.XYZ)

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 27 Juli 2022

Ir. Emil R. Kaburuan, Ph.D. IPM

LEMBAR PERSETUJUAN PENGUJI

LEMBAR PERSETUJUAN PENGUJI

NIM	:	41518110052
Nama	:	Cakra Wijaya Saputra
Judul Tugas Akhir	:	Implementasi Algoritma Levenshtein Distance Dan Algoritma Boyer-Moore Dalam Pengelompokan Work Order Berdasarkan Produk (Studi Kasus: PT.XYZ)

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 27 Juli 2022

Rahmat Rian Hidayat, ST, MMSI

LEMBAR PERSETUJUAN PENGUJI

LEMBAR PERSETUJUAN PENGUJI

NIM	:	41518110052
Nama	:	Cakra Wijaya Saputra
Judul Tugas Akhir	:	Implementasi Algoritma Levenshtein Distance Dan Algoritma Boyer-Moore Dalam Pengelompokan Work Order Berdasarkan Produk (Studi Kasus: PT.XYZ)

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 27 Juli 2022

Dhanny Permatasari Putri, S.Kom, MT

LEMBAR PENGESAHAN

LEMBAR PENGESAHAN

NIM	:	41518110052
Nama	:	Cakra Wijaya Saputra
Judul Tugas Akhir	:	Implementasi Algoritma Levenshtein Distance Dan Algoritma Boyer-Moore Dalam Pengelompokan Work Order Berdasarkan Produk (Studi Kasus: PT.XYZ)

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 27 Juli 2022

Menyetujui,

(Achmad Kodar, Drs. MT)
Dosen Pembimbing

Mengetahui,

(Wawan Gunawan, S.Kom, MT)	(Ir. Emil R. Kaburuan, Ph.D., IPM.)
Koord. Tugas Akhir Teknik Informatika	Ka. Prodi Teknik Informatika

viii

KATA PENGANTAR

Puji syukur kita panjatkan kehadirat Allah SWT atas rahmat , hidayah dan karunia-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul "Implementasi Algoritma Levenshtein Distance Dan Algoritma Boyer-Moore Dalam Pengelompokan Work Order Berdasarkan Produk (Studi Kasus: PT.XYZ)". Tugas Akhir ini dilaksanakan sebagai persyaratan untuk menyelesaikan jenjang Strata 1 (S1) Teknik Informatika Universitas Mercubuana.

Pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih kepada semua pihak yang telah membantu, khususnya kepada:

1. Bapak Achmad Kodar, Drs. MT yang telah membimbing, mendukung dan memberikan saran yang bermanfaat dalam menyelesaikan Tugas Akhir ini.
2. Bapak Ir. Emil R. Kaburuan, Ph.D., IPM. selaku ketua program studi Teknik Informatika Universitas Mercubuana
3. Bapak Wawan Gunawan, S.Kom, MT selaku koordinator Tugas akhir program studi Teknik Informatika Universitas Mercubuana
4. Ibu Saruni Dwiasnati, ST, MM, M.Kom selaku dosen pembimbing akademik
5. Keluarga besar PT XYZ atas segala bantuan dan kerjasamanya
6. Keluarga dan orang tua yang selalu memberikan dukungan, kasih sayang dan doa yang senantiasa mengiringi perjalanan penulis
7. Semua rekan – rekan Teknik Informatika khususnya Reguler 2 angkatan tahun 2018 atas motivasi dan kerjasamanya selama ini

Akhir kata, penulis mengucapkan terima kasih dan berharap Allah SWT memberikan balasan yang berlipat ganda kepada semuanya. Semoga penulisan skripsi ini bermanfaat.

Jakarta, 06 Juli 2022

Penulis

DAFTAR ISI

HALAMAN SAMPUL ..... **i**

HALAMAN JUDUL ..... **i**

LEMBAR PERNYATAAN ORISINALITAS ..... **ii**

SURAT PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR ... **iii**

SURAT PERNYATAAN LUARAN TUGAS AKHIR ..... **iv**

LEMBAR PERSETUJUAN PENGUJI ..... **v**

LEMBAR PENGESAHAN ..... **viii**

ABSTRAK ..... **ix**

ABSTRACT ..... **x**

KATA PENGANTAR ..... **xi**

DAFTAR ISI ..... **xii**

NASKAH JURNAL **1**

KERTAS KERJA **9**

BAB 1. LITERATUR REVIEW **11**

BAB 2. ANALISIS DAN PERANCANGAN **39**

BAB 3. SOURCE CODE **50**

BAB 4. DATASET **56**

BAB 5. TAHAPAN EKSPERIMEN **58**

BAB 6. HASIL SEMUA EKSPERIMEN **61**

DAFTAR PUSTAKA **66**

LAMPIRAN DOKUMEN HAKI **68**

LAMPIRAN KORESPONDENSI **70**

NASKAH JURNAL

Levenshtein Distance And Boyer-Moore For Document Grouping By Product

Saputra, Cakra Wijaya
Mercu Buana University, South Meruya, Jakarta, 11650, Indonesia

ARTICLE INFORMATION

Received: February 00, 00
Revised: March 00, 00
Available online: April 00, 00

KEYWORDS

Algoritma, Levenshtein, Boyer, String Matching

CORRESPONDENCE

Phone: +62 878 7158 3515
E-mail: cakrasaputrawijaya@gmail.com

A B S T R A C T

PT. XYZ currently uses the manual method to perform the archiving and reporting process for Work Orders by writing a manual list with an excel file. In this study, researchers had the opportunity to develop a archiving application. Web-based archiving application at PT. XYZ is an application designed to simplify the process of storing Work Order(.pdf Document) data files at PT. XYZ. This application is designed to provide solutions to problems that arise in the archiving process (process of saving the Work Order file). Currently the data is not centralized in 1 (one) data warehouse and there are no Work Order data processing features available, which makes it difficult for companies to check and analyze the required data. . Answering these problems, this research aims to create an automatic filing system for work orders at PT. XYZ. This application was developed using the prototype method using the Python and Django framework and MySQL database. This application is equipped with a string matching system with the Levenshtein Distance Algorithm and Boyer Moore Algorithm. Some of the Work Order data taken with regex in this application include data related to sales names, product names, principal names, job details, work order dates.

https://doi.org/10.30743/xxxxx

INTRODUCTION

In this era, information and communication technology is a very important role, where the position of information technology seems to have been embedded in people's lives. With the rapid advancement of information and communication technology from year to year, it provides a large space for the community to carry out every activity carried out. Not only in people's lives, business processes have turned into more complex and detailed processes.

The pattern of the business processes of each company changes rapidly and flexibly, the processing of each field is required to be more effective and efficient. Therefore, every Work Order process must be carried out in more detail and detail in order to get maximum data results. PT. XYZ currently does not have a Work Order classification system based on the appropriate product and principal.

PT. XYZ is currently archiving Work Order documents based on products and job task lists in a manual way, therefore it takes time from the Engineer's work so that time becomes ineffective and inefficient. In addition to consuming the core work of the Engineer, this also slows down the reporting process to the management side.

With these problems, it is necessary to have a system that can help overcome delayed business processes in terms of the process to get fast and accurate data. To help answer these problems, this research aims to create a filing system for the work order and list the core tasks of the engineer automatically, as well as grouping the work orders based on the product of each principal at PT. XYZ.

Implementing a data Archiving System will be designed using the Boyer Moore algorithm and Levenshtein Distance. Boyer Moore's algorithm is one of the most efficient string matching algorithms available as it actually does find matches in a sub-linear search time [1]and Levenshtein distance algorithm is one of Approximate string matching algorithm used in search string based on the estimation approach[2]

By Implementing Archiving Appat PT. XYZ hopes can to speed up business flow so that from the management side they can quickly find out which products are often requested and from the engineer side to get quick decisions when there is an imbalance between request orders and engineers available at PT. XYZ.

METHOD

Boyer Moore

The Boyer-Moore algorithm includes the most efficient string matching algorithm compared to other string matching algorithms due to an efficient nature of the algorithm, many string matching algorithms developed based on the concept of Boyer-Moore algorithm[3]The Boyer-Moore Algorithm is an algorithm that can be used in this study, because in this study it will perform string matching with a single pattern. Perform string matching from 1 word to many words in 1 sentence. Boyer Moore's algorithm has been proven as one of the most efficient algorithms in string search applications using natural language (not binary language)[4]. Boyer Moore's algorithm compares pattern characters to text characters from right to left with using two heuristics known as bad character shift and good suffix shift[5]

Levenshtein

The Metric method introduced by Levenshtein [Levenshtein (1966)] measures the similarity between two words by calculating an edit distance. The edit distance is defined as the minimum number of basic editing operations needed to transform a wrong word to a dictionary word. Thus, to correct a wrong word, one retains a set of solutions requiring fewer possible editing operations [6]. The Levenshtein Distance Algorithm is an algorithm used to compare the similarities between two strings, which are usually called the source string and the destination string, the Levenshtein Algorithm is a String Matching Algorithm that has high performance and fast time in making comparisons between two strings.

RESULTS AND DISCUSSION

Based on the results of research conducted at PT. XYZ related to the application of the Levenshtein Distance (LD) Algorithm and Boyer Moore (BD) Algorithm as a help system to perform the data recap process with a web-based application, the authors can find out about the process of comparing the similarity between two strings from the Levenshtein Distance Algorithm and Boyer-Moore Algorithm

Table 1. Sample Data set

No	Sample Data Set
1	Renewal Layanan Microsoft 365 Business Standard Qty 1 license
2	Layanan OSS Alibaba
3	Renewal - Layanan Carbonite Qty 3

Table 2. Sample Data train and Product

No	Sample Data Train	Group by Product
1	Microsoft 365	Microsoft Office 365
2	Alibaba	Alibaba Cloud
3	Carbonite	Carbonite

The following is the process of comparing strings from the Levenshtein Distance Algorithm. The 2-dimensional matrix used in performing calculations or string matching distance values in the Levenshtein Distance Algorithm is a number of functions contained in the

Levenshtein Distance Algorithm in the form of deletion, insertion, exchange, and no change of words.

$D(s,t) = \min D(s-1,t) + 1$ (deletion)
$D(s,t) = min\ D(s,t-1) + 1$ (insertion)
$D(s,t) = min\ D(s-1,t-1) + 1, sj \neq ti$ (exchange)
$D(s,t) = \min D(s-1,t-1)\ , sj = ti$(no changes)

Table 3. Levenshtein Distance – Compare two string

		A	L	I	B	A	B	A
	0	1	2	3	4	5	6	7
A	1	0	1	2	3	4	5	6
L	2	2	0	1	2	3	4	5
I	3	3	4	0	1	2	3	4

In Table 3 it is known by using the Levenshtein Distance Algorithm that the difference between the words ALIBABA and ALI has 4 different letters.

Table 4. Levenshtein Distance – Manual Compare

A	L	I	B	A	B	A
A	L	I	1	2	3	4

And the following is the process of comparing two strings with the Boyer-Moore algorithm

Table 5. Boyer-Moore - Index String

Latter	A	Z	U	R	E
Index	0	1	2	3	4

Table 6. Boyer-Moore - String Value

Latter	A	Z	U	R	E	*
Value	4	3	2	1	5	5

To find out the value of the letter is calculated by
Value = length – index – 1
Letter A = 5 – 0 – 1 = 4
Letter Z = 6 – 1 – 1 = 3
Letter U = 6 – 2 – 1 = 2
Letter R = 6 – 3 – 1 = 1
Letter E = Last Word Length Index 5 (if the next box does not have the same value)
Letter * = is a symbol that is not in the data letter

Table 7. Boyer-Moore - Compare two string

Data 1	A	L	I	B	A	B	A	A	A	Z	U	R	E
Data 2	A	Z	U	R	E								
						A	Z	U	R	E			
									A	Z	U	R	E

It is known from Table 7 is a word search process with the Boyer Moore Algorithm, it is known that

Process 1: Found letters other than the letters in the Data2, therefore must shift as much as 5 boxes according to the values in Table 6
Process 2: After shifting 6 boxes, found the letter E in Data2 and the letter Z in Data1, if it meets the letter Z, then a shift of 3 boxes is carried out
Process 3: After shifting 4 squares, the final letter E, the last letter E meets the letter E and equations are carried out in each of the back boxes

Based on the experimental analysis of string matching with the Levenshtein Distance Algorithm, Boyer-Moore Algorithm and the Merger of Levenshtein Distance & Boyer-Moore Algorithm at PT XYZ using

994 data sets in .pdf format and 200 data trains.

From 994 data set using the Levenshtein Distance Algorithm using 200 data trains, it can be seen that to run the regex process for cleansing the data set, string matching using the Levenshtein Distance method and ingesting to the mysql database takes 65.6 seconds from 994 raw data, string matching results using the Levenshtein Distance Algorithm with a score greater than 50/100 totaling 478 data with an accuracy value of 48%

With the Boyer Moore algorithm with 200 data train data, 775 of the 994 datasets were detected. The Boyer-Moore algorithm has an accuracy of 77.9% and takes 84.1 seconds

Merging the Boyer-Moore Algorithm and Levenshtein Distance Algorithm from 994 data, the process is from the Levenshtein Distance Algorithm will take the 10 highest scores from the data train, after that is done is string matching using the Boyer-Moore Algorithm from a string with 10 train data that has been taken from Levenshtein Distance Algorithm. The time required is 69.22 seconds and the detected data is 696 data, with a combination of the Levenshtein distance algorithm and the Boyer Moore algorithm has an accuracy of 70%

No	Method	Data Set	Data train	Time	Result
1	Levenshtein Distance	994	200	65.6 seconds	994 data, score >=50 478 data. Accuracy 48%
2	Boyer-Moore	994	200	84.1 seconds	775 data. Accuracy 77.9 %
3	Levenshtein Distance + Boyer Moore	994	200	69.22 seconds	696 data. Accuracy 70 %

CONCLUSIONS

The conclusion from using the Levenshtein Distance Algorithm and Boyer-Moore Algorithm is that the Levenshtein Distance Algorithm matches the comparison of words with almost the same number and fast processing time, while the Boyer-Moore Algorithm has the advantage of being able to find strings of words in the pattern, has processing time. which is slower than the Levenshtein Distance but has a high degree of accuracy.

The Levenshtein Distance Algorithm and the Boyer-Moore Algorithm can be the answers regarding the accuracy of the data and the processing time required for the PT XYZ archive application. However, there are some things that need to be improved in the

next research, namely to make it more detailed and specific regarding the train data that will be used.

REFERENCES

[1] A. Ojugo and A. O. Eboka, "Signature-Based Malware Detection Using Approximate Boyer Moore String Matching Algorithm," *International Journal of Mathematical Sciences and Computing*, vol. 5, no. 3, pp. 49–62, 2019, doi: 10.5815/ijmsc.2019.03.05.

[2] M. M. Yulianto, R. Arifudin, and A. Alamsyah, "Autocomplete and Spell Checking Levenshtein Distance Algorithm To Getting Text Suggest Error Data Searching In Library," *Scientific Journal of Informatics*, vol. 5, no. 1, p. 75, 2018, doi: 10.15294/sji.v5i1.14148.

[3] "Searching Process Using Bayer Moore Algorithm in Medicinal Information Media III. RESULT AND DISCUSSION", doi: 10.35940/ijrte.

[4] F. Tinus Waruwu, "The IJICS | Fince Tinus Waruwu | http://ejurnal.stmik-budidarma.ac.id/index.php/ijics Application Of Boyer Moore Algorithm for Text Searching," *International Journal of Informatics and Computer Science (The IJICS*, vol. 1, no. 1, pp. 18–22, 2017, [Online]. Available: http://ejurnal.stmik-budidarma.ac.id/index.php/ijics

[5] A. Z. M. Saleh, N. A. Rozali, A. G. Buja, K. A. Jalil, F. H. M. Ali, and T. F. A. Rahman, "A Method for Web Application Vulnerabilities Detection by Using Boyer-Moore String Matching Algorithm," *Procedia Computer Science*, vol. 72, pp. 112–121, 2015, doi: 10.1016/j.procs.2015.12.111.

[6] A. S. Lhoussain, G. Hicham, and Y. Abdellah, "Adaptating the levenshtein distance to contextual spelling correction," *International Journal of Computer Science and Applications*, vol. 12, no. 1, pp. 127–133, 2015.

[7] H. T. Saidah, M. S. N. Ishlah, and N. N. Rokhmah, "Autocorrect pada Modul Pencarian Drugs e-Dictionary Menggunakan Levenshtein Distance," *Rekayasa Sistem dan Teknologi Informasi*, vol. 4, no. 1, pp. 64–69, 2020, [Online]. Available: https://doi.org/10.29207/resti.v4i1.1401

[8] I. Ahmad, R. I. Borman, G. G. Caksana, and J. Fakhrurozi, "Implementasi String Matching Dengan Algoritma Boyer-Moore Untuk Menentukan Tingkat Kemiripan Pada Pengajuan Judul Skripsi/Ta Mahasiswa (Studi Kasus: Universitas Xyz)," *SINTECH (Science and Information Technology) Journal*, vol. 4, no. 1, pp. 53–58, 2021, doi: 10.31598/sintechjournal.v4i1.699.

[9] S. B. Kishor, "INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH EMERGE TOWARDS LEVENSHTEIN

' S DISTANCE ALGORITHM FOR ESTIMATION Neelesh Channawar," no. 2, pp. 16–17, 2019.

[10] P. Peggy and S. Hansun, "Optimasi Pencarian Kata pada Aplikasi Penerjemah Bahasa Mandarin – Indonesia Berbasis Android dengan Algoritma Levenshtein Distance," *Jurnal ULTIMA Computing*, vol. 7, no. 1, pp. 19–23, 2016, doi: 10.31937/sk.v7i1.449.

[11] M. Mosbach, I. Stenger, T. Avgustinova, and D. Klakow, "Incom.py - A toolbox for calculating linguistic distances and asymmetries between related languages," *International Conference Recent Advances in Natural Language Processing, RANLP*, vol. 2019-Septe, pp. 810–818, 2019, doi: 10.26615/978-954-452-056-4_094.

[12] S. Sarkar, P. Pakray, D. Das, and A. Gelbukh, "JUNITMZ at SemEval-2016 Task 1: Identifying semantic similarity using levenshtein ratio," *SemEval 2016 - 10th International Workshop on Semantic Evaluation, Proceedings*, pp. 702–705, 2016, doi: 10.18653/v1/s16-1108.

[13] B. H. Irawan, M. S. H. Simarangkir, and E. Erlinna, "Deteksi Kemiripan Judul Skripsi Menggunakan Algoritma Levenshtein Distance Pada Kampus Stmik Mic Cikarang," *Edutic - Scientific Journal of Informatics Education*, vol. 7, no. 2, pp. 143–149, 2021, doi: 10.21107/edutic.v7i2.10051.

[14] A. Info, L. Distance, and S. Function, "Implementation of Levenshtein Distance Algorithm in The Digital Biology Dictionary Search Function," vol. 4, no. 2, pp. 89–96, 2021.

[15] M. H. Febiawan, A. Setiawan, and A. Primadewi, "Sistem Pendeteksi Dini Plagiarisme Menggunakan Algoritma Levenshtein Distance," *Jurnal Komtika (Komputasi dan Informatika)*, vol. 3, no. 1, pp. 18–27, 2020, doi: 10.31603/komtika.v3i1.3464.

[16] "2-9 National Booyer - Barcode(done)".

[17] V. F. Utamandani, D. Ariyadi, I. P. Astuti, A. F. Cobantoro, and K. N. Fitri, "Penerapan Algoritma Boyer Moore Pada Pengarsipan Dokumen Di Kua Punung Kab. Pacitan Jawa Timur," *Network Engineering Research Operation*, vol. 6, no. 2, p. 113, 2021, doi: 10.21107/nero.v6i2.226.

[18] F. G. Weddiningrum, A. Kunaefi, and N. Wahyudi, "Detection of Hoax Content on Social Media in Indonesia using a Levenshtein Distance Method," pp. 255–261, 2020, doi: 10.5220/0008905602550261.

[19] R. Rahim, A. S. Ahmar, A. P. Ardyanti, and D. Nofriansyah, "Visual Approach of Searching Process using Boyer-Moore Algorithm," *Journal of Physics: Conference Series*, vol. 930, no. 1, 2017, doi: 10.1088/1742-6596/930/1/012001.

AUTHOR(S) BIOGRAPHY

Saputra, Cakra Wijaya
Cakra Wijaya is a student and a data developer, was born in Tangerang on April 11th 2000. Have an enthusiastic about Cloud Platform and Data Engineering also Python Scripting. For now, He is a Data Developer from one of the biggest Cloud provider in Jakarta, Indonesia

KERTAS KERJA

Ringkasan

Kertas kerja ini merupakan material kelengkapan artikel jurnal dengan judul di atas. Kertas kerja berisi semua material hasil penelitian Tugas Akhir yang tidak dimuat/atau disertakan di artikel jurnal. Di dalam kertas kerja ini disajikan:

1. Literature review

   Merupakan tinjauan pustaka yang digunakan untuk memahami suatu topik yang akan dikerjakan dapat berasal dari jurnal, buku, internet maupun sumber resmi lainnya. Di dalam literatur review dapat berupa teori maupun gagasan dari penelitian lain yang mereka peroleh dengan melakukan penelitian.

2. Analisa dan Perancangan

   Melakukan analisa serta perancangan sistem yang dibutuhkan mulai dari software, hardware dll. Digunakan untuk menunjang keberhasilan dalam penelitian.

3. Dataset

   Pada dataset berupa hasil pengujian terhadap suatu sistem sebelum dilakukan implementasi sistem yang akan dilakukan. Dataset dapat digunakan sebagai dasar perbandingan apabila telah melakukan penelitian.

4. Source code

   Dijelaskan mengenai lingkungan sistem yang digunakan, perintah eksekusi untuk menjalankan program, source code suatu sistem. Menjelaskan berbagai konfigurasi sebelum memulai penelitian.

5. Tahapan eksperimen

   Merupakan tahapan eksperimen yang dilakukan pada penelitian mulai dari analisa sistem, perancangan sistem, konfigurasi sistem, hingga dapat dilakukan implementasi. Terdapat diagram alur, flow chart dll.

6. Hasil eksperimen

Pada bagian ini memuat semua hasil eksperimen yang dikerjakan. Dapat ditulis dalam berupa tabel maupun gambar pengujian. Terdapat kesimpulan hasil penelitian dan saran dari peneliti.