

Implementasi Algoritma Levenshtein Distance Untuk Misspelled Word Pada Pencarian Lagu Melayu

Marcel Rino Batisya¹, Klaudius Jevanda BS^{2*}

^{1,2} Universitas Katolik Musi Charitas/Informatika
Jl. Bangau No 60 Palembang, Indonesia

e-mail: ¹marcelrinobatisya10@gmail.com, ^{2*}k_jevanda@ukmc.ac.id

Informasi Artikel

Diterima: 02-02-2023

Direvisi: 03-03-2023

Disetujui: 12-03-2023

Abstrak

Saat ini, banyak orang tidak bisa hidup tanpa musik. Musik merupakan bagian yang sangat berguna bagi kehidupan manusia untuk bersantai, menghilangkan stres, ataupun mengubah suasana hati. Musik sangat mudah ditemukan di banyak *website* saat ini. Namun orang-orang yang menyukai musik terkadang lupa atau salah paham dengan judul lagu yang sering didengarkan dan dinyanyikan khususnya lagu melayu di berbagai *website*. Sehingga membuat kesulitan untuk mencari judul lagu melayu yang diinginkan, karena *website* tidak dapat menginformasikan ejaan kata yang dimaksud sesuai dengan arti judul lagu. Maka dari itu, penulis membuat aplikasi pencarian lagu melayu terkhusus karya Arif Putra berbasis *website* dengan menggunakan algoritma *Levenshtein Distance*. Algoritma ini digunakan untuk memperbaiki kesalahan ejaan judul lagu, baik karena kecepatan mengetik ataupun *typo* dari *user*. *Levenshtein Distance* menghitung jumlah minimum dari operasi perubahan paling sedikit yang diperlukan untuk mengubah satu *string* menjadi *string* lainnya. Operasi perubahan tersebut adalah penyisipan, penghapusan, dan pertukaran. Sedangkan metode pengembangan sistem yang digunakan penulis adalah metode *Waterfall*. Hasil penelitian menunjukkan penerapan algoritma *Levenshtein Distance* dapat memperbaiki pencarian judul lagu yang memiliki kesalahan ejaan kata. Sedangkan dari hasil responden berjumlah 85 orang menyatakan 23,53% sangat setuju dan 60% setuju bahwa algoritma *Levenshtein Distance* yang diterapkan dapat meminimalisir kesalahan ejaan pada pencarian lagu melayu karya Arif Putra.

Kata Kunci: Algoritma *Levenshtein Distance*; Lagu Melayu, Kesalahan ejaan kata

Abstract

Today, many people cannot live without music. Music is a very useful part of human life to relax, relieve stress, or change mood. Music is very easy to find on many websites today. However, people who like music sometimes forget or misunderstand the title of the song that is often heard and sung, especially Malay songs, on various websites. So that makes it difficult to find the desired Malay song title, because the website cannot inform the spelling of the word in question according to the meaning of the song title. Therefore, the author created a web-based search application for Malay songs, especially Arif Putra's works, using the *Levenshtein Distance* algorithm. This algorithm is used to correct spelling mistakes in song titles, either due to typing speed or typos from users. *Levenshtein Distance* calculates the minimum number of change operations required to convert one string into another. The change operations are insert, delete, and exchange. While the system development method used by the author is the *Waterfall* method. The results showed that the application of the *Levenshtein Distance* algorithm can improve the search for song titles that have spelling errors. Meanwhile, from the results of the respondents, there were 85 people who stated that 23.53% strongly agreed and 60% agreed that the *Levenshtein Distance* algorithm applied could minimize spelling errors in searching Malay songs by Arif Putra.

Keywords: *Levenshtein Distance Algorithm*; Malay Song; Misspelled word

1. Pendahuluan

Musik merupakan suatu karya seni suara dalam bentuk lagu atau komposisi musik. Musik sendiri mengkomunikasikan pikiran dan

perasaan penciptanya melalui beberapa aspek unsur musik. Unsur musik tersebut meliputi ritme, melodi, harmoni, bentuk, struktur lagu, dan ekspresif secara keseluruhan (Jamalus,



1988:1-2). Menurut Banoe (2003:288) musik merupakan sebuah karya seni yang mengkaji dan mendeskripsikan banyak suara ke dalam pola yang dapat dipahami dan dimengerti oleh manusia. Pada saat ini musik juga merupakan salah satu karya seni yang sangat pesat perkembangannya. Hal ini diakui oleh pengamat musik Wendi Putranto dalam kutipan Pratiwi (2022) bahwa musik yang diminati oleh orang-orang pun menjadi semakin beragam sehingga membuat *Cross Genre* atau musik lintas genre mulai menduduki posisi utama di kancah musik secara umum.

Salah satu *Cross Genre* yang ada di Indonesia yaitu musik ber *genre pop* melayu yang merupakan *genre* terbesar perkembangannya di Indonesia. Menurut Rahayu Kertawigun bahwa *genre* musik Dangdut dalam artian melayu di Indonesia memiliki peluang booming di era digital saat ini. Hal ini disebabkan karena musik ber *genre* Melayu yang masuk dalam kategori musik Melayu standar yang bisa disandingkan dengan musik ber *genre* EDM (*Electronic Dance Music*). Sehingga melahirkan musik melayu modern yang justru terlahir *fresh* dan mudah masuk ke usia remaja sampai dengan lanjut usia (Randy, 2019).

Seperti penyanyi Arif Putra, meskipun ia masih remaja tetapi sudah banyak melantunkan musik Melayu modern. Orang-orang mengenalnya dengan nama Arif. Berdasarkan tanggal lagu dan *Insight* terpopuler di *Youtube*, Arif yang bernama panjang Arif Putra masuk ke dalam peringkat 2 terpopuler selama 34 minggu. Arif Putra ini merupakan penyanyi yang berasal dari Padang, Sumatera Barat. Sejalan dengan tempat asalnya, Padang merupakan salah satu tempat berkembangnya orkes melayu yang disebut sebagai lagu-lagu Melayu Deli (Widiarko, 2023).

Kepopuleran inilah yang membuat penulis mengangkat nama Arif Putra sebagai subjek penelitian. Sehingga *user* baik usia remaja maupun lanjut usia yang menggemari Arif Putra dapat melihat seluruh lagu karya Arif Putra. Akan tetapi *user* terkadang salah menafsirkan judul lagu yang sering didengar dan dinyanyikan di berbagai *website* yang disebabkan karena ada terjadinya *human error* atau *typo*. Tom Stafford, seorang psikolog di *University of Sheffield* di Inggris mengemukakan bahwa *typo* merupakan contoh tipikal dari bentuk kesalahan yang dilakukan oleh manusia. Otak manusia tidak selalu mampu menangkap hal-hal setepat yang seharusnya (Pratama, 2022).

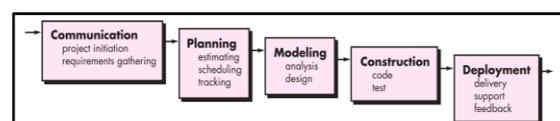
Terjadinya *typo* tersebut membuat *website* lain terkadang tidak dapat mencari judul lagu dengan maksimal, sehingga

membuat hambatan bagi *user* sebagai penikmat musik dalam melakukan pencarian judul lagu. Berdasarkan permasalahan tersebut, penulis melakukan penelitian tentang lagu-lagu ber *genre* Melayu berbasis *website* yang dipadukan dengan sistem pencarian yang dapat memudahkan *user* untuk mencari lagu melayu yang diinginkan dengan menggunakan algoritma yang belum diterapkan pada *website* lain yakni algoritma *levenshtein distance*. Algoritma ini berguna untuk membenarkan penulisan ejaan kata, sehingga memudahkan *user* dalam pencarian lagu Melayu. Adapun algoritma ini dibuat oleh Vladimir Levensthein pada tahun 1965 untuk menghitung jumlah minimum operasi pengubahan, penghapusan, penyisipan karakter yang digunakan untuk menemukan pemisahan antar dua *string* (Darnita & Muntahanah, 2018).

Penelitian-penelitian sebelumnya yang memiliki kesamaan dengan penulis terkait algoritma *levenshtein distance* adalah Kuswanto (2020); Sadiyah *et al* (2020); Larasati and Marisa (2019); Sari *et al* (2019); Oktaviyani *et al* (2019); Santoso *et al* (2019); Darnita and Muntahanah (2018); dan Rosmala and Risyad (2017). Dimana perbedaannya dengan penelitian sebelumnya adalah subjek penelitian untuk pencarian lagu melayu karya Arif Putra yang diimplementasikan dalam sebuah aplikasi berbasis *website*.

2. Metode Penelitian

Metode pengembangan sistem yang digunakan dalam penelitian ini adalah *waterfall*. Dimana *waterfall* adalah teknik yang telah teruji untuk membuat perangkat lunak secara metodelis dan berurutan. Meskipun umumnya dianggap kuno, model generik dalam rekayasa perangkat lunak yang disajikan Winston Royce pada tahun 1970-an sejauh ini merupakan model paling populer di lapangan *Software Engineering* (SE). Di *waterfall* (Gambar 1), setiap tahap berturut-turut harus menunggu tahap sebelumnya selesai sebelum melanjutkan tahap selanjutnya (Pressman, 2015:42).



Gambar 1. Tahapan *Waterfall*

Tahapan *Waterfall* adalah sebagai berikut :

1) *Communication*

Tahapan pertama, penulis menganalisis masalah, pengumpulan data, dan definisi fitur *software* untuk pencarian judul lagu Melayu. Pengumpulan data menggunakan

studi literatur dan observasi yang diperoleh dari artikel jurnal, buku dan *internet*.

2) *Planning*

Tahapan ini, penulis menetapkan rencana pembuatan *software* yang terdiri dari tugas teknis yang akan dilakukan, kemungkinan resiko yang terjadi, sumber daya yang dibutuhkan dalam membuat *software*, hasil yang akan dibuat dan jadwal pengerjaan *software*.

3) *Modeling*

Pada tahapan ini, penulis melakukan tahapan merancang dan memodelkan arsitektur sistem yang akan dibangun menggunakan *Unified Modeling Language* (UML). Setelah itu, penulis mendesain yang sesuai dengan rancangan dan pemodelan arsitektur sistem berupa antarmuka sistem.

4) *Construction*

Tahapan selanjutnya, penulis melakukan proses menerjemahkan hasil desain yang telah dibuat ke dalam *coding* yang dapat dikenali oleh mesin menggunakan pemrograman HTML, CSS, PHP dan MySQL sebagai basis data. Setelah itu, langkah selanjutnya dilakukan *testing* terhadap *software* yang telah dibuat guna menemukan kesalahan atau *error* yang kemudian bisa diperbaiki pada *software* tersebut.

5) *Deployment*

Tahapan ini, penulis melakukan tahapan implementasi *software* kepada *user*. Implementasi ini bertujuan untuk mengetahui perbaikan, kelayakan dan evaluasi dari *software* yang telah dibuat oleh penulis, yang selanjutnya *user* diminta untuk mengisi kuesioner.

Penerapan Algoritma *Levenshtein Distance* dapat diilustrasikan secara manual dengan beberapa tahapan berikut :

1. Inisialisasi

S = *String* Awal, dan T = *String* Target
 Misalkan, S = Remlan Malam dan T = Rembulan Malam

a) Hitung panjang S dan T

S = Remlan Malam = 11 string
 T = Rembulan Malam = 14 string

b) Buat matriks berukuran 0...S baris dan 0...T kolom

Tabel 1. Membuat *Matrix*

	S	
T		0

c) Terapkan baris pertama dengan 0...T dan kolom pertama dengan 0...S

Tabel 2. Inisialisasi Matrix Sesuai Panjang String

	S	R	e	m	l	a	n	M	a	l	m	
T	0	1	2	3	4	5	6	7	8	9	0	1
R	1											
e	2											
m	3											
b	4											
u	5											
l	6											
a	7											
n	8											
	9											
M	10											
a	11											
l	12											
a	13											
m	14											

2. Proses

- a) Memeriksa S[i] untuk 1 < i < S dan T[j] untuk 1 < j < T. Jika S[i] = T[j], maka entrinya adalah nilai yang terletak tepat di sebelah kanan diagonal kiri berwarna merah, yaitu d[1,1] = d[1-1,1-1] = d[0,0] = 0

Tabel 3. Jika S[i] = T[j] maka Nilai = 0

	S	R	e	m	l	a	n	M	a	l	m	
T	0	1	2	3	4	5	6	7	8	9	0	1
R	1	0										
e	2											
m	3											
b	4											
u	5											
l	6											
a	7											
n	8											
	9											
M	10											
a	11											
l	12											
a	13											
m	14											

- b) Jika S[i] ≠ T[j], maka entrinya adalah d[i,j] = d[2,1] minimum dari:

- 1) Nilai yang naik satu tingkat tepat diatasnya pada warna biru, ditambah satu, yaitu :
 $d[i,j-1]+1$
 $d[2,1-1]+1$
 $d[2,0]+1 = 2+1 = 3$
- 2) Nilai yang naik satu tingkat tepat dikirinya pada warna merah , ditambah satu, yaitu
 $d[i-1,j]+1$
 $d[2-1,1]+1$
 $d[1,1]+1 = 0 + 1 = 1$
- 3) Nilai yang naik satu tingkat tepat didiagonal atas sebelah kirinya pada warna abu-abu, ditambah satu, yaitu
 $d[i-1,j-1]+1$
 $d[2-1,1-1]+1$
 $d[1,0]+1 = 1 + 1 = 1$
 Ini berarti nilai minimum yang diperoleh pada matrix $d[2,1]$ memiliki *distance* = 1.

Tabel 4. Jika $S[i] \neq T[j]$

	S	R	e
T	0	1	2
R	1	0	1

- c) Jika $S[i] = T[j]$ dan jika $S[i] \neq T[j]$ sampai $d[2,4]$.

Tabel 5. Penyelesaian Sampai $d[i,j] = d[2,4]$

	S	R	e	m	l	a	n	M	a	l	m	
T	0	1	2	3	4	5	6	7	8	9	0	1
R	1	0	1	2	3	4	5	6	7	8	9	0
e	2	1	0	1	2	3	4	5	6	7	8	9
m	3	2	1	0	1	2	3	4	5	6	7	8
b	4	3	2									

- d) Pada *Matrix* $d[i,j] = d[3,4]$ dimana memiliki $i = 3$ dan i merupakan S = l dan dimana $j = 4$ dan j merupakan T = i. Dimana pada matrix $S[i] \neq T[j]$, maka menggunakan rumus $d[i,j]$ minimum untuk mencari nilai minimum dari *distance* yang ada dan mengubah nilai *distance* yang tadinya 0 menjadi +1 dengan rumus nilai minimum dari penyisipan, penghapusan, dan pergantian karena pada garis potong memiliki *matrix* yang berbeda atau huruf yang berbeda.

Tabel 6. Jika $S[i] \neq T[j]$

	S	R	e	m	l	a	n	M	a	l	m	
T	0	1	2	3	4	5	6	7	8	9	0	1
R	1	0	1	2	3	4	5	6	7	8	9	0

e	2	1	0	1	2	3	4	5	6	7	8	9
m	3	2	1	0	1	2	3	4	5	6	7	8
b	4	3	2	?								
u	5											
l	6											
a	7											
n	8											
	9											
M	0											
a	1											
l	1											
a	2											
a	1											
a	3											
m	1											
	4											

- e) Jika $S[i] \neq T[j]$, maka entriya adalah $d[i,j] = d[3,4]$ minimum dari:

- 1) Nilai yang terletak tepat diatasnya pada warna biru, ditambah satu, yaitu
 $d[i,j-1]+1$
 $d[3,4-1]+1$
 $d[3,3]+1 = 0+1 = 1$
- 2) Nilai yang terletak tepat dikirinya pada warna merah , ditambah satu, yaitu
 $d[i-1,j]+1$
 $d[3-1,4]+1$
 $d[2,4]+1 = 2 + 1 = 3$
- 3) Nilai yang terletak pada tepat didiagonal atas sebelah kirinya pada warna abu-abu, ditambah satu, yaitu
 $d[i-1,j-1]+1$
 $d[3-1,4-1]+1$
 $d[2,3]+1 = 1 + 1 = 2$

Jadi nilai minimum yang diperoleh pada *matrix* $d[3,4]$ memiliki *distance* = 1

Tabel 7. Penyelesaian $d[i,j] = d[3,4]$

	S	R	e	m	l	a	n	M	a	l	m	
T	0	1	2	3	4	5	6	7	8	9	0	1
R	1	0	1	2	3	4	5	6	7	8	9	0
e	2	1	0	1	2	3	4	5	6	7	8	9
m	3	2	1	0	1	2	3	4	5	6	7	8
b	4	3	2	1								

- f) Jika $S[i] = T[j]$ dan jika $S[i] \neq T[j]$ sampai $d[11,14]$

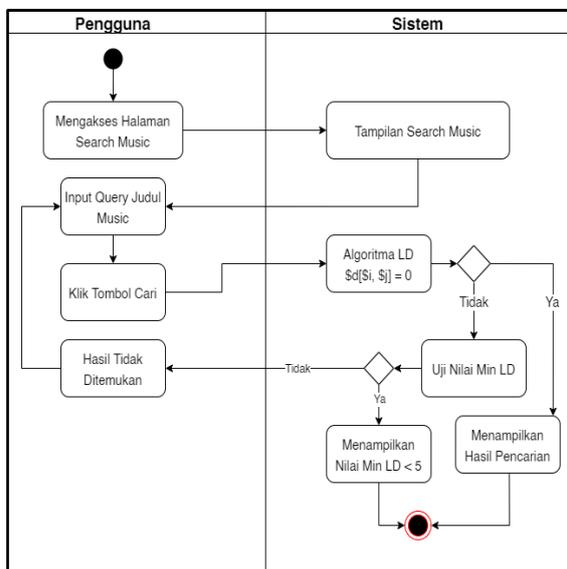
Tabel 8. Penyelesaian $d[i,j] = d[11,14]$

	S	R	e	m	l	a	n	M	a	l	m
T	0	1	2	3	4	5	6	7	8	9	0
R	1	0	1	2	3	4	5	6	7	8	9
e	2	1	0	1	2	3	4	5	6	7	8
m	3	2	1	0	1	2	3	4	5	6	7
b	4	3	2	1	1	2	3	4	5	6	7
u	5	4	3	2	2	2	3	4	5	6	7
l	6	5	4	3	2	3	3	4	5	6	6
a	7	6	5	4	3	2	3	4	5	5	6
n	8	7	6	5	4	3	2	3	4	5	6
1	9	8	7	6	5	4	3	2	3	4	5
M	0	9	8	7	6	5	4	3	2	3	4
a	1	1	1	1	1	1	1	1	1	1	1
l	1	1	1	1	1	1	1	1	1	1	1
a	2	1	0	9	8	7	6	5	4	3	2
1	1	1	1	1	1	1	1	1	1	1	1
a	3	2	1	0	9	8	7	6	5	4	3
m	1	1	1	1	1	1	1	1	1	1	1
m	4	3	2	1	0	9	8	7	6	5	4

3. Hasil

Jadi pada kasus ini jarak atau *distance* dari kata S = Remlan Malm sebagai *String* Awal dan T = Rembulan Malam sebagai *String* Target adalah 3.

Prosedur penelitian dalam bentuk *activity* diagram dari pencarian lagu melayu menggunakan algoritma *levenshtein distance* seperti pada Gambar 2.



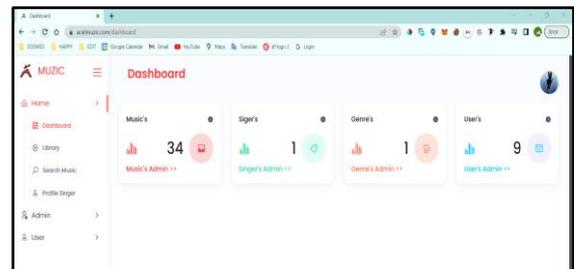
Gambar 2. Activity Diagram Algoritma Levenshtein Distance

Hasil implementasi *software* akan diujikan ke *user* menggunakan analisis statistik *deskriptif* dengan cara menyebarkan kuesioner. Dimana statistika *deskriptif* merupakan metode yang digunakan untuk menafsirkan ataupun meringkas data yang kemudian ditampilkan dalam bentuk yang dapat dipahami semua orang. Bentuk tersebut salah satunya rata-rata data yang dihitung dan ditampilkan dalam bentuk tabel maupun grafik (Wahyudi & Djamaris, 2018).

3. Hasil dan Pembahasan

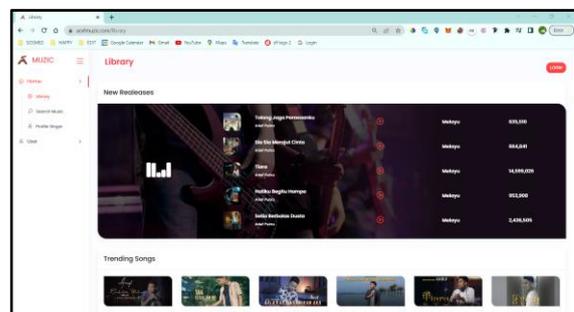
1. Implementasi *Interface Website*

Website dapat diakses oleh *Admin* dan *User*. Beberapa tampilan *interface* dari implementasi *website* yang telah dibuat yakni :



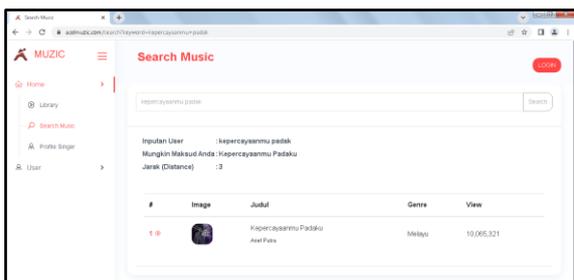
Gambar 3. Interface Dashboard

Pada *interface dashboard* (Gambar 3) hanya dapat di akses oleh *Admin* yang memberikan informasi seputar banyaknya data dalam aplikasi *website* dan memiliki tombol detail yang terletak di bawah informasi, dimana jika di klik *text* tersebut akan langsung masuk ke halaman admin untuk *Music's Admin*, *Singer's Admin*, *Genre's Admin*, dan *User's Admin*.



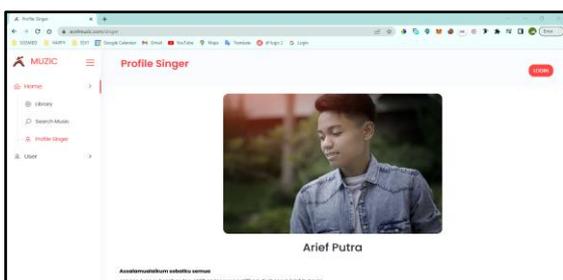
Gambar 4. Interface Library

Pada *interface library* (Gambar 4) memiliki beberapa *cardbox* yang berisi informasi dari *New Songs*, *Top 8 Trending*, dan *Songs*. Informasi tersebut dapat diakses oleh *Admin* maupun *User*. Jika di mengklik salah satu dari *cardbox* tersebut, maka menampilkan informasi detail lagu.



Gambar 5. Interface Search Music

Pada interface search music (Gambar 5), user dapat mencari judul lagu dengan cara memasukkan keyword yang diinginkan dan mengklik tombol Search. Selanjutnya sistem akan melakukan proses pencarian menggunakan algoritma levenshtein distance untuk mengecek kesalahan ejaan judul lagu yang diinginkan oleh user.

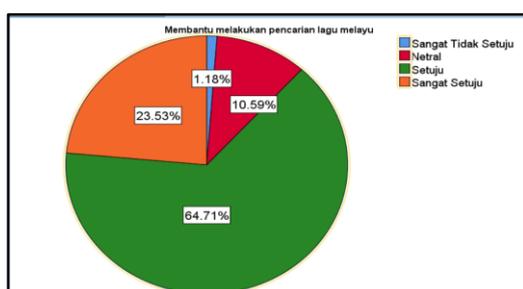


Gambar 6. Interface Profile Singer

Pada Interface Profile Singer (Gambar 6) dapat diakses baik Admin maupun User yang berisi informasi dari penyanyi yakni Arief Putra.

2. Analisis Statistik Deskriptif

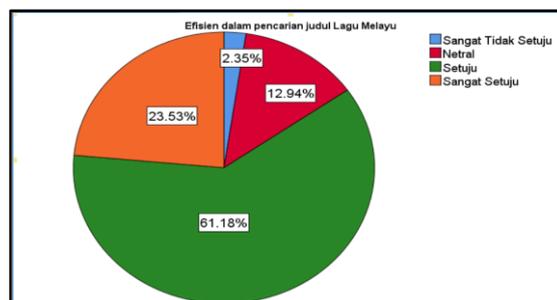
Analisis menggunakan statistik deskriptif. Dimana data yang digunakan dalam pengujian diambil dari kuesioner (<https://forms.gle/jy5k4eo6hHbFHLps9>) yang diisi oleh 85 responden secara random dari usia > 15 tahun sampai > 50 tahun dengan 10 pertanyaan. Berikut ini adalah hasil analisis dari beberapa pertanyaan yang didapat dari data responden :



Gambar 7. Membantu Melakukan Pencarian Judul Lagu Melayu

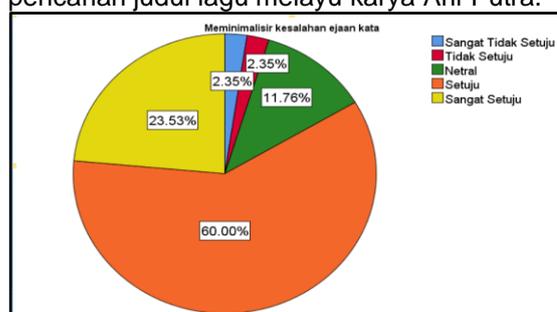
Berdasarkan hasil pengelolaan data kuesioner tentang website membantu melakukan pencarian judul lagu melayu

(Gambar 7) diketahui bahwa responden menjawab setuju dengan presentase 64.71% dan sangat setuju dengan 23.53%. Hal ini dapat dinyatakan bahwa hampir seluruh responden menjawab setuju bahwa website ini sangat membantu melakukan pencarian lagu melayu karya Arif Putra.



Gambar 8. Efisien dalam Pencarian Judul Lagu Melayu

Berdasarkan hasil pengelolaan data kuesioner (Gambar 8) diketahui bahwa 61.18% dan 23.53% responden menjawab setuju ataupun sangat setuju. Hal ini dapat dinyatakan bahwa hampir seluruh responden menjawab setuju bahwa website lebih efisien dalam pencarian judul lagu melayu karya Arif Putra.



Gambar 9. Meminimalisir Kesalahan Ejaan Kata

Berdasarkan hasil pengelolaan data kuesioner (Gambar 9) diketahui bahwa 60% menjawab setuju dan 23,53% menjawab sangat setuju. Hal ini dapat dinyatakan bahwa hampir seluruh responden menjawab setuju bahwa website membantu dalam meminimalisir kesalahan ejaan kata.

4. Kesimpulan

Berdasarkan penelitian yang telah dilakukan, dapat disimpulkan bahwa penerapan algoritma Levenshtein Distance untuk Misspelled Word berhasil diimplementasikan untuk pencarian lagu melayu karya Arif Putra. Berdasarkan pengujian software ke user dari kuesioner dengan jumlah 85 responden menyatakan 23.53% sangat setuju dan 60% setuju bahwa algoritma Levenshtein Distance yang diterapkan dapat meminimalisir kesalahan

ejaan kata pada *website* pencarian lagu melayu karya Arif Putra.

Referensi

- Banoë, Pono. (2003). *Kamus Musik*. Yogyakarta: Kanisius.
- Darnita, Yulia, & Muntahanah, Muntahanah. (2018). *Aplikasi Sistem Pencarian Lagu Menggunakan Algoritma Levenshtein Distance*. *Journal of Technopreneurship and Information System (JTIS)* 1(3):61–67.
- Jamalus. (1988). *Panduan Pengajaran Buku Pengajaran Musik Melalui Pengalaman Musik*. Jakarta: Proyek Pengembangan Lembaga Pendidikan.
- Kuswanto, Welly. (2020). *Implementasi Algoritma Levenshtein Distance Dengan Restful Web Service Pada Kata Bahasa Indonesia Ke Bahasa Jawa Berbasis Web*. *JUSTINDO (Jurnal Sistem Dan Teknologi Informasi Indonesia)* 5(2):78–85. doi: 10.32528/justindo.v5i2.3447.
- Larasati, Isbalaikana, Marisa, Fitri. (2019). *Penerapan Algoritma Levenshtein Distance dalam Aplikasi Pengolahan Surat*. *Jurnal Pengembangan Manajemen Informatika dan Komputer (DINAMIKA DOTCOM)* Vol 10, No 2 tahun 2019
- Oktaviyani, Enny Dwi, Sherly Christina, and Deddy Ronaldo. (2019). *Keywords Search Correction Using Damerau Levenshtein Distance Algorithm*. *Conference Senatik STT Adisutjipto Yogyakarta* Vol 5:167–76. doi:http://dx.doi.org/10.28989/senatik.v5i0.344
- Pratama, Kevin Rizky. (2022). *Kenapa Sering Typo Saat Mengetik?*. <https://tekno.kompas.com/read/2022/02/09/18150017/kenapa-sering-typo-saat-mengetik-?page=all>
- Pratiwi, Ryan Sarah. (2022). *Cross-Genre Bakal Tetap Jadi Tren Musik Di Tahun 2022*. <https://lifestyle.kompas.com/read/2022/07/31/192018420/cross-genre-bakal-tetap-jadi-tren-musik-di-tahun-2022?page=all>
- Pressman. (2015). *Software Engineering : A Practitioner's Approach (Eight Edit)*. McGraw-Hill Education. Vol. 59. 8th ed. New York: Raghu Srinivasan.
- Randy, Erha. (2019). *Peluang Musik Dangdut Di Era Digital, Ini Kata Rahayu Kertawiguna*. <https://nalar.id/peluang-musik-dangdut-di-era-digital-ini-kata-rahayu-kertawiguna/>
- Rosmala, Dewi, and Zulfikar Muhammad Risyad. (2017). *Algoritma Levenshtein Distance Dalam Aplikasi Pencarian Isu Di Kota Bandung Pada Twitter*. *MIND Journal* Vol 2, No 2 : 1–12. doi: 10.26760/mindjournal.v2i2.1-12.
- Sadiah, Halimah Tus, Muhamad Saad Nurul Ishlah, and Nisa Najwa Rokhmah. (2020). *Autocorrect Pada Modul Pencarian Drugs E-Dictionary Menggunakan Levenshtein Distance*. *Rekayasa Sistem Dan Teknologi Informasi* 4(1):64–69.
- Santoso, Puji, Pundhi Yuliawati, Ridwan Shalahuddin, and Aji Prasetya Wibawa. (2019). *Damerau Levenshtein Distance for Indonesian Spelling Correction*. *Jurnal Informatika* Vol 13, No 2 : 11. doi: 10.26555/jifo.v13i2.a15698
- Sari, Yunita Purnama, Gede Aditra Pradnyana, and I. Made Agus Wirawa. (2019). *Pengembangan Aplikasi Kamus Bahasa Bima - Bahasa Indonesia Menggunakan Algoritma Levenshtein Distance Sebagai Spell Checker Berbasis Android*. *KARMAPATI (Kumpulan Artikel Mahasiswa Pendidikan Teknik Informatika)* Vol 8 No 2:86–95. DOI: <https://doi.org/10.23887/karmapati.v8i2.17964>
- Wahyudi, David, & Djamaris, Aurino R. A. (2018). *Metode Statistik Untuk Ilmu Dan Teknologi Pangan*. Penerbit Universitas Bakrie
- Widiarko, Iqbal. (2023). *Sejarah Musik Dangdut Hingga Perkembangannya Di Indonesia*. *Celebrities.Id*. Retrieved (<https://www.celebrities.id/read/sejarah-musik-dangdut-60m1io>)