

Aplikasi Sistem Pencarian Lagu menggunakan Algoritma Levenshtein Distance

Yulia Darnita¹, Muntahanah²

^{1,2}Program Studi Informatika, Fakultas Teknik, Universitas Muhammadiyah Bengkulu

Jl. Bali, Bngkulu 38119

yuliadarnita@umb.ac.id¹

muntahanah@umb.ac.id²

Abstract— Nowadays, in everyday life, humans cannot be separated from the sound of music both inside the house and outside the home. The music in human life is very useful because music is a relaxation to release stress, or to change the atmosphere. But many music lovers sometimes forget the songs they often hear and sing, making them wonder what the song title is from the lyrics they sing, the method is less efficient because maybe the place we ask also doesn't know the song title we mean. ASR technology often does not meet the accuracy or accuracy of mobile devices, especially due to noisy environmental factors. Therefore, this application will implement the Levenshtein Distance algorithm. This algorithm will correct sound input errors, both due to noise and pronunciation errors. Levenshtein distance is a string matrix that is used to measure the difference between two strings, the value of the distance between two strings determined by the minimum number of change operations needed to make a transformation from a string to another string, these operations are insertion, deletion, or substitution. Based on the results of testing carried out it can be concluded: This system can help users in searching song titles based on some lyrics with optimal search results. Application of the Levenshtein Distance Algorithm can help overcome the problems with keyword spelling by adding, inserting and deleting characters.

Keywords: Music, Songs, Lyrics, algoritma Levenshtein Distance

Abstrak— Saat ini dalam kehidupan sehari-hari manusia tidak terlepas dari suara musik baik itu di dalam rumah maupun di luar rumah. Adanya musik di dalam kehidupan manusia sangat bermanfaat karena musik sebagai relaksasi untuk melepas stress, atau untuk mengubah suasana. Akan tetapi banyak penikmat musik terkadang lupa terhadap lagu yang sering mereka dengar dan nyanyikan, sehingga membuat mereka menjadi bertanya-tanya apa judul lagu dari lirik yang ia nyanyikan, cara tersebut kurang efisien sebab mungkin tempat kita bertanya juga tidak mengetahui judul lagu yang kita maksudkan. teknologi ASR ini sering tidak menemui akurasi atau ketepatan pada perangkat *mobile*, khususnya karena faktor lingkungan yang berisik. Oleh karena itu pada aplikasi ini akan diimplementasikan algoritma Levenshtein Distance Algoritma ini akan memperbaiki kesalahan *input* suara, baik karena *noise* maupun kesalahan pengucapan. 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*). Berdasarkan hasil pengujian yang dilakukan dapat diambil kesimpulan : Sistem ini dapat membantu penggunaannya dalam melakukan pencarian judul lagu berdasarkan sebagian lirik dengan hasil pencarian yang optimal, Penerapan Algoritma Levenshtein Distance dapat membantu mengatasi permasalahan pada kesalahan ejaan kata kunci dengan mekanisme penambahan, penyisipan dan penghapusan karakter.

Kata Kunci: Musik, Lagu, Lirik, algoritma Levenshtein Distance

I. PENDAHULUAN

Saat ini dalam kehidupan sehari-hari manusia tidak terlepas dari suara musik baik itu di dalam rumah maupun di luar rumah. Adanya musik di dalam kehidupan manusia sangat bermanfaat karena musik sebagai relaksasi untuk melepas stress, atau untuk mengubah suasana. Akan tetapi banyak penikmat musik terkadang lupa terhadap lagu yang sering mereka dengar dan nyanyikan, sehingga membuat mereka menjadi bertanya-tanya apa judul lagu dari lirik yang ia nyanyikan pada kerabat-kerabatnya, cara tersebut kurang efisien sebab mungkin tempat kita bertanya juga tidak mengetahui judul lagu yang kita maksudkan. Hal tersebut mendorong diciptakan kemudahan-

kemudahan yang mampu melayani manusia berkomunikasi dengan komputer layaknya manusia dengan manusia. Salah satu kemudahan yang diusahakan dalam penelitian ini adalah suatu aplikasi pencarian judul lagu yang digunakan.

Dengan demikian diharapkan dapat membantu para pengguna dalam mencari lagu ketika pengguna tidak mengetahui atau lupa dengan judul lagu yang sering didengar dan dinyanyikan.

Teknologi Automatic Speech Recognition (ASR) telah menjadi cara yang populer untuk komunikasi dengan *smartphone*. Teknologi ASR menyediakan komunikasi verbal antara user dan perangkat *mobile*, serta menyediakan pengalaman

interaksi yang paling natural terhadap user. Bagaimanapun, performa recognition pada teknologi ASR ini sering tidak menemui akurasi atau ketepatan pada perangkat mobile, khususnya karena faktor lingkungan yang berisik. Oleh karena itu, pada aplikasi ini akan diimplementasikan algoritma Levenshtein Distance untuk memperbaiki tingkat akurasi dari input suara user. Algoritma ini akan memperbaiki kesalahan input suara, baik karena noise maupun kesalahan pengucapan.

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 satu string menjadi string lainnya. Operasi-operasi tersebut adalah penyisipan (insertion), penghapusan (deletion), atau penukaran (substitution) [1].

II. LANDASAN TEORI

A. Android

Android adalah sistem operasi yang berbasis Linux untuk telepon seluler seperti telepon pintar dan komputer tablet. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia [2].

B. Lirik Lagu

Lirik lagu merupakan ekspresi seseorang tentang suatu hal yang sudah dilihat, didengar maupun dialaminya. Dalam mengekspresikan pengalamannya, penyair atau pencipta lagu melakukan permainan kata-kata dan bahasa untuk menciptakan daya tarik dan kekhasan terhadap lirik atau syairnya. Permainan bahasa ini dapat berupa permainan vokal, gaya bahasa maupun penyimpangan makna kata dan diperkuat dengan penggunaan melodi dan notasi musik yang disesuaikan dengan lirik lagunya sehingga pendengar semakin terbawa dengan apa yang dipikirkan pengarangnya. Lirik atau syair lagu dapat dianggap sebagai puisi begitu pula sebaliknya. Musik dan lagu sebagai sebuah pesan komunikasi dapat menyampaikan pesan motivasi dalam konteks kehidupan untuk mendorong dan

menyemangati individu untuk melakukan sesuatu demi tercapainya [3].

C. Pengenalan Suara ((Speech recognition)

Pengenalan ucapan atau suara (speech recognition) adalah suatu teknik yang memungkinkan sistem komputer untuk menerima input berupa kata yang diucapkan tanpa memperdulikan siapa pembicaranya. Kata-kata tersebut diubah bentuknya menjadi sinyal digital dengan cara mengubah gelombang suara menjadi sekumpulan angka lalu disesuaikan dengan kode-kode tertentu dan dicocokkan dengan suatu pola yang tersimpan dalam suatu perangkat. Hasil dari identifikasi kata yang diucapkan dapat ditampilkan dalam bentuk tulisan sehingga dapat dibaca menggunakan perangkat teknologi. [4].

D. Algoritma

Dalam matematika dan komputasi algoritma merupakan kumpulan perintah yang saling berkaitan untuk menyelesaikan suatu masalah. Perintah-perintah ini dapat diterjemahkan secara bertahap dari awal hingga akhir. Dalam penyusunannya diperlukan urutan serta logika agar algoritma yang dihasilkan sesuai dengan yang diharapkan. Algoritma merupakan bagian terpenting yang tidak dapat dipisahkan dari pemrograman. Meskipun sintaksis dan semantik yang dibuat benar adanya, dengan algoritma yang keliru, permasalahan yang ingin dipecahkan dengan teknik pemrograman tidak akan berhasil. Oleh karena itu, sebelum membuat program aplikasi, hal pertama yang harus kita pahami algoritma atau prosedur pemecahannya. Hal ini bertujuan agar program yang telah dibuat dapat sesuai dengan yang diharapkan [5].

E. Algoritma Levenshtein Distance

Levenshtein Distance dibuat oleh Vladimir Levenshtein pada tahun 1965. Perhitungan edit distance didapatkan dari matriks yang digunakan untuk menghitung jumlah perbedaan string antara dua string. Perhitungan jarak antara dua string ini ditentukan dari jumlah minimum operasi perubahan untuk membuat string A menjadi string B. Ada 3 macam operasi utama yang dapat dilakukan oleh algoritma ini : 1. Pengubahan karakter 2. Penambahan karakter 3. Penghapusan karakter Algoritma ini berjalan mulai dari pojok kiri atas sebuah array dua dimensi yang telah diisi sejumlah karakter string awal dan string target dan diberikan nilai cost. Nilai cost pada ujung kanan bawah menjadi nilai edit distance yang menggambarkan jumlah perbedaan dua string.

Langkah-langkah dari algoritma levenshtein distance (Benisius : 2010) yaitu :

- 1) Misalkan kata pertama adalah string asal (s) dan kata kedua adalah string target (t)
- 2) Tentukan m sebagai panjang dari s
- 3) Tentukan n sebagai panjang dari t 4. $d(0,0) = 0$
- 4) Jika $m = 0$, return n dan stop
- 5) Jika $n = 0$, return m dan stop
- 6) Buat matriks yang berisi 0 sampai m baris dan 0 sampai n kolom
- 7) Periksa setiap karakter dari s (i dari 1 sampai m) dan dari t (j dari 1 sampai n)
- 8) Jika $s[i]$ sama dengan $t[j]$, maka $cost = 0$, jika berbeda maka $cost = 1$
- 9) Tentukan cell $d[i, j]$ dari matriks yang sama dengan angka terendah,

$$d[i, j] = \begin{cases} d[i-1, j] - 1 & (1) \\ d[i, j-1] + 1 & (2) \\ \min(d[i-1, j-1] + cost, \dots) & (3) \end{cases}$$

Keterangan rumus :

- d= Cell-cell yang terdapat dalam matrik
- i = Karakter yang terdapat pada string asal
- j = Karakter yang terdapat pada string target
- Min= Nilai minimum
- Cost= Nilai 0 dan 1

10) Setelah selesai, maka jarak akan didapat dari cell $d(m, n)$

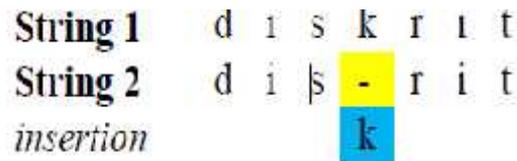
Berikut ini merupakan pseudocode dari algoritma Levenshtein Distance :

```
n = length(s)
m = length(t)
if n = 0 then return m
else if m = 0 then return n
else
  for i = 0 to n do
    cost[0][i] = i
  for i = 0 to n do
    cost[0][i] = i
```

Pada algoritma Levenshtein distance, terdapat tiga macam operasi yang dapat dilakukan yaitu [6] :

1. Operasi Penyisipan Karakter (Insertion)

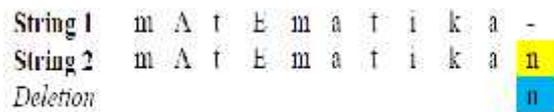
Operasi penyisipan karakter berarti menyisipkan karakter ke dalam suatu string. Contohnya string 'disrit' menjadi string 'diskrit', dilakukan penyisipan karakter 'k' diakhirstring. Penyisipan karakter tidak hanya dilakukan di tengah string, namun bisa disisipkan diawal maupun disisipkan diakhirstring. Ilustrasi:



Gbr 1 Insertion

2. Operasi Penghapusan Karakter (Deletion)

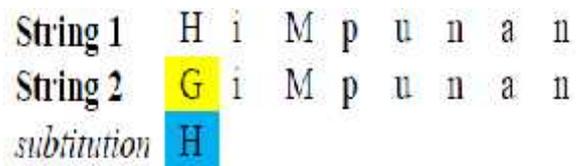
Operasi penghapusan karakter dilakukan untuk menghilangkan karakter dari suatu string. Contohnya string 'matematika' karakter terakhir dihilangkan sehingga menjadi string 'matematika'. Pada operasi ini dilakukan penghapusan karakter 'n'. Ilustrasi:



Gbr 2. Deletion

3. Operasi Penukaran Karakter (Substitution)

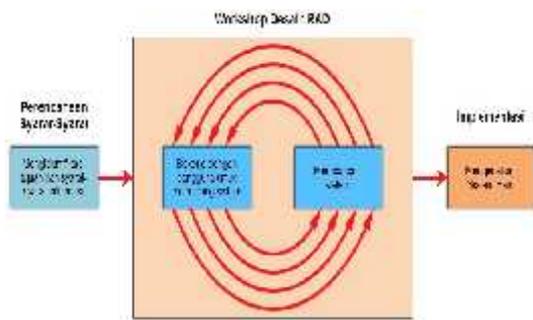
Operasi penukaran karakter merupakan operasi menukar sebuah karakter dengan karakter lain. Contohnya penulis menuliskan string 'gimpunan' menjadi 'himpunan'. Dalam kasus ini karakter 'g' yang terdapat pada awal string, diganti dengan huruf 'h'. Ilustrasi:



Gbr 3. Substitution

III. METODE PENELITIAN

Rapid Application Development (RAD) atau rapid prototyping adalah model proses pembangunan perangkat lunak yang tergolong dalam teknik Incremental (bertingkat). RAD menekankan pada siklus pembangunan pendek, singkat, dan cepat. Waktu yang singkat adalah batasan yang penting untuk model ini, RAD adalah suatu pendekatan berorientasi objek terhadap sistem yang mencakup suatu metode pengembangan serta perangkat-perangkat lunak. RAD bertujuan mempersingkat waktu yang biasanya diperlukan dalam siklus hidup pengembangan sistem tradisional antara perancangan dan penerapan suatu sistem informasi. Pada akhirnya, RAD sama-sama berusaha memenuhi syarat-syarat bisnis yang berubah secara cepat [7].



Gbr 4. RapidApplicationDevelopment (RAD)

A. Analisa Data

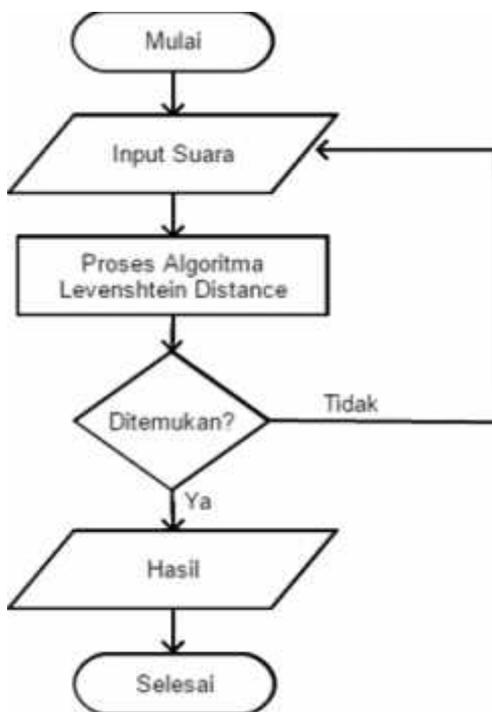
Pada penelitian penulis menggunakan teknik pengumpulan data dengan wawancara, observasi dan studi pustaka untuk mendapatkan data-data dalam penelitian

B. Desain Sistem

Tugas-tugas yang dibutuhkan untuk membangun satu atau lebih representasi dari Sistem

Pencari Lagu Berdasarkan Lirik Dengan Perintah Suara.

a. Flowchart Algoritma Levenshtein Distance



Gbr 5. Flowchart Aplikasi

b. Perhitungan Algoritma Levenshtein Distance

a) Operasi Penyisipan Karakter (Insertion)

Insertion adalah menyisipkan karakter ke dalam suatu string. Penyisipan karakter tidak hanya dilakukan di tengah string, namun bisa disisipkan diawal maupun disisipkan diakhir string.

b) Operasi Penghapusan Karakter (Deletion)

Deletion digunakan untuk menghilangkan karakter dari suatu string.

String 1	s	e	k	r	i	P	s	I
String 2	s	-	k	r	i	P	s	i
Deletion		e						

c. Operasi Penukaran Karakter (Substitution)

Substitution merupakan operasi menukar sebuah karakter dengan karakter lain.

String 1	p	r	o	p	o	C	a	I
String 2	p	r	o	p	o	S	a	L
Substitution						C		

Langkah Penyelesaian Algoritma Levenshtein Distance:

Misalkan S = String Awal, dan T = String Target.

Langkah 1: Inisialisasi

- 1) Hitung panjang S dan T, misalkan m dan n
- 2) Buat matriks berukuran 0...m baris dan 0...n kolom
- 3) Inisialisasi baris pertama dengan 0...n
- 4) Inisialisasi kolom pertama dengan 0...m

Langkah 2: Proses

- 1) Periksa S[i] untuk 1 < i < n
- 2) Periksa T[j] untuk 1 < j < m
- 3) Jika S[i] = T[j], maka entrinya adalah nilai yang terletak pada tepat didiagonal atas sebelah kiri, yaitu d[i,j] = d[i-1,j-1]
- 4) Jika S[i] ≠ T[j], maka entrinya adalah d[i,j] minimum dari:
 - Nilai yang terletak tepat diatasnya, ditambah satu, yaitu d[i,j-1]+1
 - Nilai yang terletak tepat dikirinya, ditambah satu, yaitu d[i-1,j]+1
 - terletak pada tepat didiagonal atas sebelah kirinya, ditambah satu, yaitu d[i-1,j-1]+1

Langkah 3: Hasil entri matriks pada baris ke-i dan kolom ke j, yaitu d[i,j]

Langkah 2 diulang hingga entri d[m,n] ditemukan.

Aturan penyelesaian Levenshtein Distance :

Rule 1 :

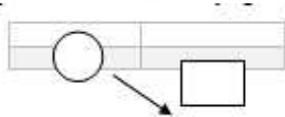
Jika karakter 1 tidak sama dengan karakter 2 maka Nilai terkecil antara Replace, Remove, Insert +1.

Replace	Remove
Insert	Min +1

Replace Remove
Insert Min +1

Rule 2 :

Jika karakter 1 sama dengan karakter 2 maka Nilai yang ada pada atas sudut kiri di salin.



Berikut Contoh dari pengisian matriks Langkah 3: Hasil entri matriks pada baris ke-i dan kolom ke j, yaitu d[i, j].

Levenshtein distance misalkan kita ingin membandingkan kata "Jurnal" dengan "junal"

		J	U	R	N	A	L
J	0	1	2	3	4	5	6
U	1	0	1	2	3	4	5
N	2	1	0	1	2	3	4
A	3	2	1	1	1	2	3
L	4	3	2	2	2	1	2
	5	4	3	3	3	2	1

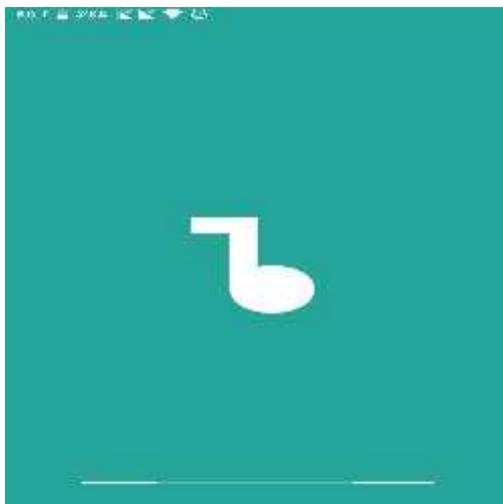
Setelah seluruh matriks terisi nilai, maka perbandingan string "Jurnal" dan "Junal" terletak pada baris ke 5 dan kolom ke 6, yaitu 1.

IV. Hasil dan Pembahasan

A. Hasil

1. SplashScreen

Splash Screen merupakan tampilan pertama kali Aplikasi dijalankan, tampilan splashscreen berdurasi sekitar 2 detik sebelum penggunaanya masuk ke dalam menu utama.



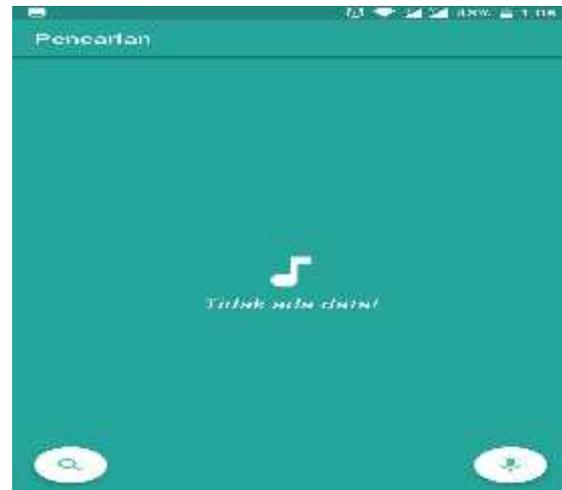
Gbr 6. SplashScreen

Pada gambar 6 di atas merupakan tampilan dari menu utama dari aplikasi yang penulis buat, pada menu utama ini terdapat tombol pencarian untuk mencari lagu berdasarkan lirik baik itu dengan perintah suara atau dengan cara pengetikan di kolom pencarian, tombol data Music untuk menambah dan menghapus lirik lagu sesuai dengan keinginan pengguna, selanjutnya terdapat juga tombol tentang yang di dalamnya akan terdapat informasi seputar tentang pembuat.



Gbr 7. Menu Utama

2. Menu Pencarian



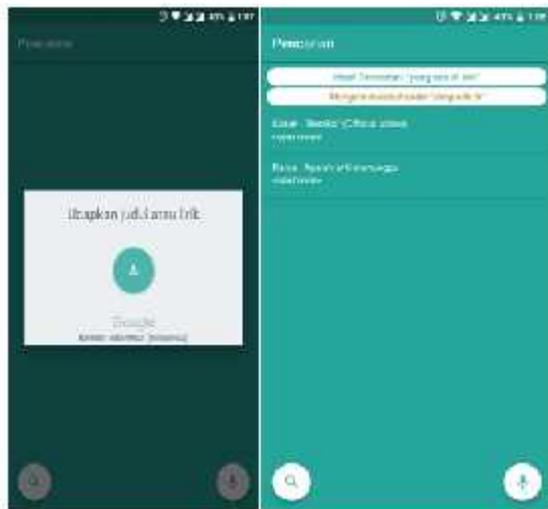
Gbr. 8 Menu Pencarian

Gambar 8 di atas merupakan tampilan dari menu pencarian ketika pengguna mengetuk tombol pencarian pada menu utama, di dalam menu ini terdapat pencarian lagu berdasarkan lirik dengan perintah suara, untuk dapat melakukan pencarian dengan perintah suara pengguna dapat mengetuk iconmic disudut kanan bawah. Untuk pencarian tanpa suara dengan cara mengetik sendiri bagian

lirik lagu, pengguna dapat mengetuk icon cari disudut kiri bawah.

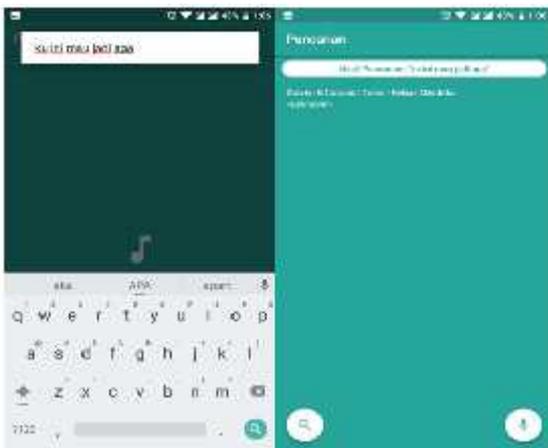
3. Pencarian dan Hasil Dengan Perintah Suara

Pada tampilan 9 di atas merupakan tampilan dari pencarian lagu berdasarkan lirik dengan perintah suara dan hasil pencariannya. Seperti yang kita lihat pada tampilan gambar di atas saat pengucapan penulis mengucapkan kata “yang ada di sini” selanjutnya sistem akan mencari lagu yang mempunyai lirik tersebut dan ketika ditemukan sistem akan menampilkan judul dari lagu yang dimaksud.



Gbr 9. Pencarian dan Hasil Dengan Perintah Suara

4. Pencarian dan Hasil Dengan Pengetikan Lirik



Gbr 10. Pencarian Lagu

Pada tampilan 10 di atas merupakan tampilan dari pencarian lagu berdasarkan lirik tanpa suara dengan cara pengetikan lirik di kolom pencarian dan terdapat juga tampilan hasil dari pencariannya. Seperti yang kita lihat pada tampilan gambar di atas penulis mengetik kata “ku ini mau jadi apa” selanjutnya sistem akan mencari lagu

yang mempunyai lirik tersebut dan ketika ditemukan sistem akan menampilkan judul dari lagu yang dimaksud.



Gbr 11. Music Player

Pada gambar 11 di atas merupakan tampilan dari music Player ketika pengguna mengetuk judul lagu yang ditampilkan oleh sistem setelah sebelumnya pengguna melakukan pencarian lagu berdasarkan lirik. Di dalam menu ini lagu akan dimainkan dan akan ditampilkan juga lirik lagunya agar pengguna dapat memastikan bahwa lagu tersebut adalah lagu yang dicari.

5. Menu Data Music

Menu data music ini berfungsi untuk mengelola data music pada sistem yang dibuat. Untuk dapat menambah playlist atau lagu pada sistem ini pengguna menambahkan terlebih dahulu lagu yang diinginkan ke dalam folder LaraMusic pada memori eksternal smartphone pengguna, lagu yang telah ditambahkan otomatis akan muncul pada sistem. Selanjutnya untuk memasukkan lirik lagu tersebut pengguna cukup mengetuk lagu yang baru ditambahkan sebelumnya maka akan muncul dua pilihan yaitu play dan lirik lagu untuk memasukkan lirik lagu pengguna dapat mengetuk tombol lirik lagu.



Gambar 12. Menu Data Music

	dicari		Ditolak
ketik lirik pada kolom Pencarian	Dapat menampilkan judul lagu yang dicari	Berhasil ditampilkan	() Diterima [] Ditolak
Klik Tombol Kembali	Dapat Kembali ke Menu Utama	Berhasil Kembali	() Diterima [] Ditolak

IV. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil dari perancangan dan pengujian yang telah dilakukan pada Sistem Pencari Lagu Berdasarkan Lirik pada tabel 1 di bawah ini :

1. Sistem ini dapat membantu penggunaanya dalam melakukan pencarian judul lagu berdasarkan sebagian lirik dengan hasil pencarian yang optimal.
2. Penerapan Algoritma Levenshtein Distance dapat membantu mengatasi permasalahan pada kesalahan ejaan kata kunci dengan mekanisme penambahan, penyisipan dan penghapusan karakter.

B. Saran

Sistem pencarian lagu ini masih bersifat off line sehingga diharapkan dikemudian hari dapat dikembangkan secara online.

Referensi

[1] Pratama dan pemungkas. 2016. Analisis Kinerja Algoritma Levenshtein Distance Dalam Mendeteksi Kemiripan Dokumen Teks. *Jurnal Logika*, Jilid 6, No.

[2] Gunawan, G., & Cahyani, M. I. (2018). Penerapan Algoritma Kruskal Dalam Mencari Lokasi Anjungan Tunai Mandiri Bank Rakyat Indonesia Cabang Bengkulu Berbasis Android. *Journal of Technopreneurship and Information System (JTIS)*, 1(2), 44-49..

[3] Hidayat, Rahmat. 2014. Analisis Semiotika Makna Motivasi Pada Lirik Lagu “Laskar Pelangi” Karya Nidji. *eJournal Ilmu Komunikasi*, Volume 2, Nomor

[4] Suryadharma, K. Budiman, G. dan Irawan, B. 2014. Perancangan Aplikasi Speech To Text Bahasa Inggris Ke Bahasa Bali Menggunakan Pocketsphinx Berbasis Android. *e-Proceeding of Engineering*, Vol.1, No.1.

[5] Ramadhani, Cipta. 2015. Dasar algoritma dan Struktur Data dengan Bahasa JAWA. Andi Offset, Yogyakarta.

[6] Junedy, Richard. 2014. Perancangan Aplikasi Deteksi Kemiripan Isi Dokumen Teks dengan Menggunakan Metode Leveshtein Distance. *Jurnal Pelita Informatika Budi Darma Vol. VII No.2*, Jurusan Teknik Informatika, STMIK Budi Darma, Medan.

[7] Kenneth E. Kendall, Julie E. Kendall, 2010, Analisis dan Perancangan Sistem, Jakarta , PT Indeks

B. Pembahasan

Berikut adalah pengujian dari Menu Utama Sistem Pencari Lagu Berdasarkan Lirik pada tabel 1 di bawah ini :

Tabel 1 Pengujian Menu Utama

Kasus Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Buka Aplikasi	Dapat menampilkan Menu Utama	Menu berhasil tampil	() Diterima [] Ditolak
Klik Tombol Keluar	Dapat Keluar dari aplikasi	Berhasil Keluar dari Aplikasi	() Diterima [] Ditolak

Berikut ini adalah hasil pengujian menu pencarian lagu berdasarkan lirik lagu tampak pada table 2 :

Tabel 2 Pengujian menu music player

Kasus Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Klik Tombol Pencarian	Dapat menampilkan Menu Pencarian	Menu berhasil tampil	() Diterima [] Ditolak
Cari dengan suara	Dapat menampilkan judul lagu yang	Berhasil ditampilkan	() Diterima []