

Penerapan Metode *Levenshtein Distance* untuk Mengukur Similaritas pada Pola Suara Burung yang Menggunakan *Discrete Cosine Transform*

Hendro Nugroho¹, Andy Rachman², Isa Albana³

^{1,2}Teknik Informatika, ³Sistem Infromasi

^{1,2,3}Institut Teknologi Adhi Tama Surabaya

Surabaya, Indonesia

e-mail: ¹dosh3ndro@itats.ac.id, ²andy.rach1910@itats.ac.id, ³isaalbanna@itats.ac.id

Diajukan: 9 Januari 2023; Direvisi: 17 Juli 2023; Diterima: 24 Agustus 2023

Abstrak

Pengenalan suara burung untuk mengetahui jenis burung sering dilakukan pada penelitian, karena burung sering terdengar dari pada kelihatan bentuk fisiknya. Berbagai macam ragam suara burung sangatlah menarik untuk dilakukan penelitian. Penelitian kali ini menggunakan metode levenshtein distance untuk mengetahui similaritas suara burung pada data ekstraksi fitur yang menggunakan metode discrete cosine transform (DCT). Data yang di-input adalah suara burung Kenari, Red Lories, Beo Merah Meksiko dengan jenis format fail WAV. Langkah penelitian ini adalah data suara burung WAV diekstraksi fitur diantaranya (1) Hamming Widowing, (2) FFT, (3) Mel Filter Bank, dan (4) DTC. Setelah mendapatkan nilai ekstraksi fitur DCT, maka data dibagi menjadi data target (T) dan data sumber (S) untuk dicari nilai similaris menggunakan metode levenshtein distance. Langkah-langkah untuk mendapatkan similaritas di antaranya (1) input nilai DTC data T dan Input nilai DTC data S, (2) menghitung jarak levenshtein distance, dan (3) menghitung nilai similaritas. Hasil yang didapat similaritas pada jenis suara burung Kenari data T dan suara burung Kenari data S memiliki similaritas 37% dan 32%. Jenis suara burung Red Lories data T dan suara burung Red Lories data S memiliki nilai similaritas 16%, 32% dan 21% dan Suara burung Beo Merah Meksiko tingkat similaritas data T dan data S memiliki nilai 58% dan 16%.

Kata kunci: Similaritas, Levenshtein distance, DCT, Pengenalan pola.

Abstract

Recognition of bird sounds to determine bird species is often carried out in research, because birds are often heard from their physical appearance. Various kinds of bird sounds are very interesting to do research. This research uses the Levenshtein distance method to find out the similarity of bird sounds in feature extraction data using the discrete cosine transform (DCT) method. The data that is input is the sound of Canaries, Red Lories, Mexican Red Parrots with the WAV fail format type. The step of this research is the feature extraction of WAV bird sound data including (1) Hamming Widowing, (2) FFT, (3) Mel Filter Bank, and (4) DTC. After obtaining the DCT feature extraction value, the data is divided into target data (T) and source data (S) to find similar values using the Levenshtein distance method. The steps to obtain similarity include (1) input the DTC data T value and input the DTC data S value, (2) calculate the levenshtein distance, and (3) calculate the similarity value. The results obtained for the similarity of Canary bird sound data T and Canary bird sound data S have a similarity of 37% and 32%. The type of sound of the Red Lories bird data T and the sound of the Red Lories bird data S have a similarity value of 16%, 32% and 21% and the sound of the Mexican Red Parrot the similarity level of data T and data S has a value of 58% and 16%.

Keywords: Similarity, Levenshtein distance, DCT, Pattern recognition.

1. Pendahuluan

Suara hewan dapat memahami perilaku, evolusi dan perbedaan antar jenis hewan[1]. Dengan banyaknya ragam suara hewan inilah sangat menarik untuk dilakukan penelitian. Beragamnya suara

hewan yang sangat menarik untuk dilakukan penelitian adalah suara burung, alasan dalam mengambil penelitian suara burung dikarenakan ada 10.000 spesies burung yang diketahui[1]. Untuk mengetahui jenis burung, dapat dilihat dari bentuk dan suara burung[2]. Untuk mengetahui jenis burung, penggunaan pengenalan pada suaranya sering dilakukan, karena burung sering terdengar daripada terlihat fisiknya[3].

Pada penelitian sebelumnya untuk klasifikasi spesies suara burung dengan menggunakan metode *deep learning* dan *convolutional neural network layers*[3]. Suara burung dilakukan ekstraksi fitur untuk klasifikasi suara burung agar mendapatkan variasi jenis burung[4]. Ekstraksi fitur pada suara burung didapat dari distribusi frekuensi[5]. Hasil frekuensi suara burung diolah menjadi sebuah *frame-frame* yang berkesinambungan sebagai input klasifikasi.

Suara burung dalam bentuk frekuensi dapat membentuk suatu pola untuk mengetahui spesies burung. Penelitian sebelumnya untuk mengenali pola suara burung menggunakan *fourier transform spectrogram* yang dihasilkan dalam bentuk matrik fitur[6]. Pengenalan pola suara burung dari frekuensi suara burung juga dilakukan dengan ekstraksi fitur dengan menggunakan metode *discrete cosine transform (DCT)* menghasilkan pola *frame* nilai *DCT*[2].

Hasil ekstraksi fitur pada suara burung yang dapat membentuk suatu pola yang dapat membuat pengenalan pola (*pattern recognition*) suara. Pada penelitian sebelumnya penggunaan pengenalan pola suara burung mendapatkan nilai *DCT* dalam bentuk *frame* nilai yang berurutan[2]. Urutan data nilai dalam bentuk matrik untuk mencari similaritas pada pola suara menggunakan pendekatan bentuk *string*. Pendekatan data dalam bentuk *string* banyak digunakan dalam segala bidang, contohnya koreksi ejaan dan bioinformatika[7].

Data dalam bentuk *string* mudah digunakan untuk similaritas dengan data lainnya sebagai data sumber dan data target. Dalam menentukan similaritas ditentukan nilai jarak pada antar dua *string*, semakin rendah nilai jarak tersebut, maka semakin tinggi tingkat similaritas antar kedua *string* tersebut[8]. Pengukuran jarak similaritas telah banyak digunakan diantaranya *hamming* dan *levenshtein*. Untuk *hamming* panjang *string* harus memiliki dimensi yang sama, sedangkan *levenshtein* mengukur similaritas dengan panjang dua *string* yang berbeda[9].

Penggunaan metode *levenshtein* dalam perhitungan jarak similaritas dengan menghitung jumlah operasi yang terjadi pada *string* yang dibandingkan seperti penambahan, penghapusan atau pengurangan karakter[8]. Proses operasi yang digunakan metode *levenshtein* tidak mempengaruhi data representasinya untuk tujuan klasifikasi atau similaritas[10]. Penelitian similaritas pola suara burung dari data *DCT* yang dibentuk menjadi *frame string* menggunakan metode *levenshtein distance*. Penggunaan metode *levenshtein distance* untuk mendapatkan nilai similaritas pada dua pola *string DCT* suara burung sehingga mengetahui similaritas jenis burung tersebut, sehingga dapat mengetahui pola *DCT* pada suara burung. Penggunaan metode *levenshtein distance* juga dikarenakan data yang diperoleh memiliki panjang *frame* nilai *DCT* yang berbeda berdasarkan waktu suara burung yang terekam.

2. Metode Penelitian

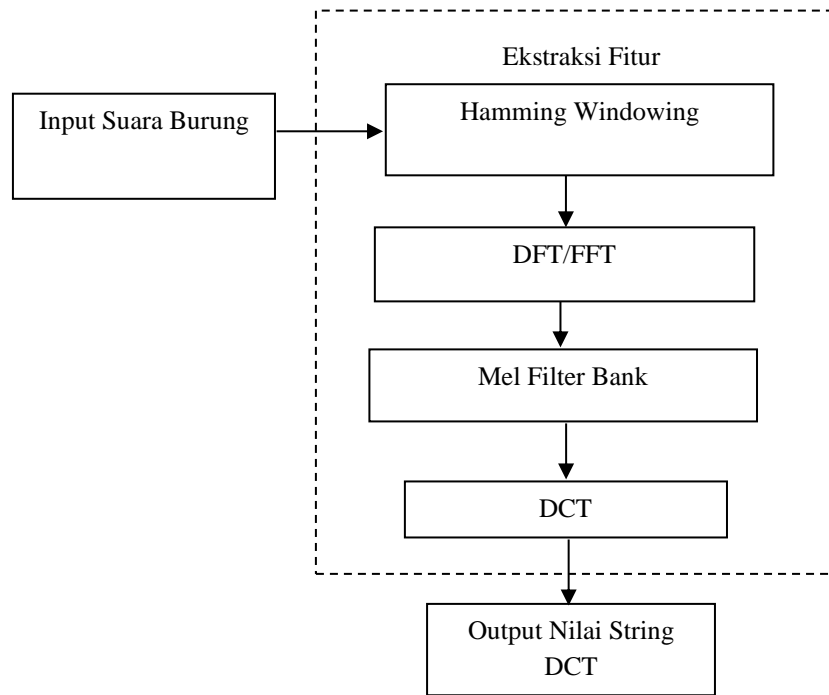
Penelitian similaritas suara burung dilakukan dengan dua cara yaitu metode untuk ekstraksi suara burung dengan menggunakan metode *DCT* dan similaritas nilai *DCT* dengan menggunakan metode *levenshtein distance*. Untuk mengetahui proses ekstraksi fitur pada suara burung dengan menggunakan metode *DCT* digambarkan dalam bentuk blok diagram gambar 1. Proses ekstraksi fitur pada suara burung terdapat beberapa tahapan antara lain input suara burung, *hamming windowing*, *FFT*, *mel filter bank*, dan *DCT*[2].

Data suara burung yang di ekstraksi fitur adalah suara burung kenari, Red Lories, dan Beo Merah Meksiko. File suara burung yang diinputkan dalam bentuk format *WAV (WAVEform)* yang memiliki tiga bagian yaitu *main chunk*, *format chunk* dan *data chunk*. Nilai yang dari file *WAV* dalam bentuk *discrete*, berupa deret bilangan yang merepresentasikan amplitud dalam domain waktu[2]. Contoh format data *input* suara burung dapat dilihat pada tabel 1. Data input suara burung dibagi menjadi data sumber (S) dan data target (T). Pembagian data suara burung digunakan untuk mencari similaritas pada data tersebut[11][9][8].

Dalam proses ekstraksi fitur suara burung dilakukan proses *hamming windowing*, penggunaan *hamming windowing* untuk membuat blok dari line frekuensi yang paling dekat[12]. Untuk mendapatkan nilai *hamming windowing* menggunakan persamaan 1 dan persamaan 2. Di dalam persamaan $W(n)$ sebagai window dan $n = 0, 1, 2, \dots, N-1$. Hasil dari *hamming windowing* ini menghasilkan *sidelobe level* yang tidak terlalu tinggi (kurang dari -43 dB), dan *noise* yang tidak terlalu besar[12][2].

$$Y[n] = X[n] * W[n] \quad (1)$$

$$W(n) = 0.54 - 0.46 \cos \left[\frac{2\pi n}{N-1} \right] \quad (2)$$



Gambar 1. Blok Diagram Ekstraksi Fitur Suara Burung

Tabel 1. Contoh Data Suara Burung Format Fail WAV

Jenis Burung	Chunk Size	Sample Rate	Byte Rate	Bits/Sampew	Panjang Data
Kenari	27796	8000	32000	16	27760
Red Lories	9560	8000	32000	16	76480
Beo Merah	83948	8000	32000	16	83912

Setelah dilakukan proses *hamming windowing* dilakukan proses *discrete fourier transform (DFT)/fast fouriee transform (FFT)*. Fungsi dari *DFT/FFT* digunakan untuk mendapatkan nilai vektor dari frekuensi sinyal diskrit waktu $x(n)$ [13]. Penggunaan *DTF* dapat dilihat pada persamaan 3. Penggunaan *DFT* dalam komputasi sangatlah lama, karen perhitungan yang membutuhkan N^2 perkalian bilangan kompleks[2], maka untuk mengatasi permasalahan tersebut menggunakan metode *FFT*. *FFT* menggunakan kompleksitas sebesar $X(N \log_2 N)$ [14].

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi n \left[\frac{k}{N} \right]}, 0 \leq k \leq N - 1 \quad (3)$$

Setelah mendapatkan nilai *FFT* dilakukan proses *mel filter bank*. Penggunaan metode *mal filter bank* mengubah sinyal suara pada domain frekuensi menjadi frekuensi *mel* dan menunjukkan beberapa besaran energi pada rentang frekuensi yang ada masing-masing *filter mel*[15]. Untuk mendapatkan nilai frekuensi *mel* menggunakan persamaan 4. Setelah mendapatkan nilai frekuensi *mel* dilakukan perhitungan *filter bank*. Penggunaan *filter bank* menggunakan representasi konvolusi yang dapat dilihat pada persamaan 5.

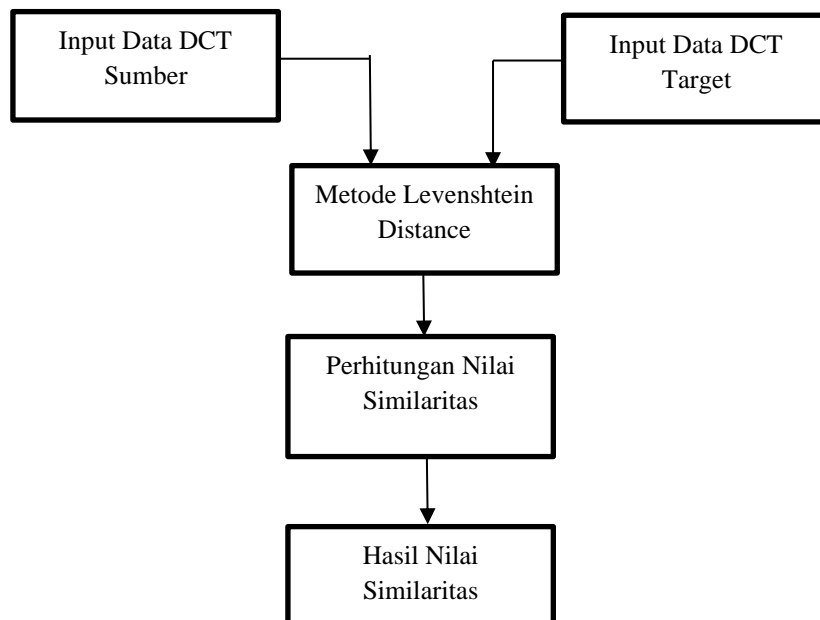
$$F(Mel) = \left[2595 * \ln \left[1 + \frac{f}{700} \right] \right] \quad (4)$$

$$Y[i] = \sum_{j=1}^{N_s} S[j]H_i [j] \quad (5)$$

Setelah mendapatkan nilai *mel filter bank* yang berupa power spectrum, maka proses selanjutnya menggunakan metode *DCT*. Penggunaan metode *DCT* untuk mendapatkan nilai vektor akustik[16]. Hasil *DCT* yang bernilai koefisien nol pada umumnya dihilangkan[17]. Penghilangan nilai nol, dikarenakan tidak riabile terhadap *speack recognition*[18]. Untuk mendapatkan nilai *DCT* dapat dilihat pada persamaan 6.

$$C_n = \sum_{k=1}^K (Log S_k) \cos \left[n \left(k - \frac{1}{2} \right) \frac{\pi}{K} \right]; n = 1, 2, \dots K \tag{6}$$

Data sumber (S) dan target (T) pada suara burung dilakukan proses ekstraksi fitur *DCT*, maka dilakukan proses metode *levenshtein distance*. Metode *levenshtein distance* digunakan untuk mendapatkan nilai similaritas. Pengambilan data dilakukan pada suara burung Kenari, Red lories dan Beo Merah Meksiko. Data sumber pada suara burung tersebut memiliki waktu satu menit dan data target memiliki waktu 2 menit, sehingga panjang *frame* dari data sumber dan target tidak sama. Dengan panjang *frame* nilai *DCT* yang berbeda, maka dilakukan similaritas data pada suara burung kenari pada data S dan suara burung Kenari pada data T, sama halnya dengan suara burung Red lories dan Bei Merah Meksiko. Untuk mengetahui cara kerja metode *levenshtein distance* untuk mendapatkan nilai similaritas dapat dilihat pada gambar 2.



Gambar 2. Blok Diagram Similaritas Suara Burung

Pada diagram similaritas suara burung, terdapat dua nilai input *DCT* yaitu data sumber (S) dan data target (T). Panjang *frame* data sumber lebih kecil dari pada data target (S<T), dikarenakan waktu data suara burung data sumber lebih pendek (1 menit) daripada data target (2 menit).

Panjang data *frame DCT* pada data sumber dan target S<T, maka operasi yang lakukan pada metode *levenshtein distance* menggunakan tiga operasi yaitu (1) tidak ada proses penghapusan, (2) operasi penambahan/disisipkan, dan (3) operasi pemindahan [9][8][11]. Tiga operasi pada metode *levenshtein distance* menghitung jumlah perubahan karekter pada operasi yang dipakai dalam pencarian similaritas *string* yang ada pada *frame*.

Setelah menghitung jumlah operasi yang digunakan untuk mengetahui similaritas data S dan T, maka dilakukan perhitungan nilai similaritas menggunakan persamaan 7.

$$similaritas = \left(1 - \frac{dis(x,y)}{Max(S,T)} \right) X 100 \tag{7}$$

Dengan *dis(x,y)* adalah nilai jumlah jarak atau *distance* yang terletak pada baris ke x dan kolom ke y Dan S adalah panjang *string/frame* sumber untuk T adalah panjang *string/frame* target.

3. Hasil dan Pembahasan

Bagian hasil dan pembahasan ini dibagi menjadi dua subbab untuk menjelaskan hasil dan pembahasan yang diperoleh di dalam penelitian. Subbab pertama menjelaskan tentang hasil yang diperoleh yang menjelaskan tentang output nilai *DCT*, hasil similaritas dan grafik yang menggambarkan bentuk kesamaan pola nilai *DCT* pada jenis suara burung. Subbab untuk pembahasan menjelaskan tentang tingkat keberhasilan di dalam penelitian ini untuk similaritas suara burung berdasarkan jenisnya.

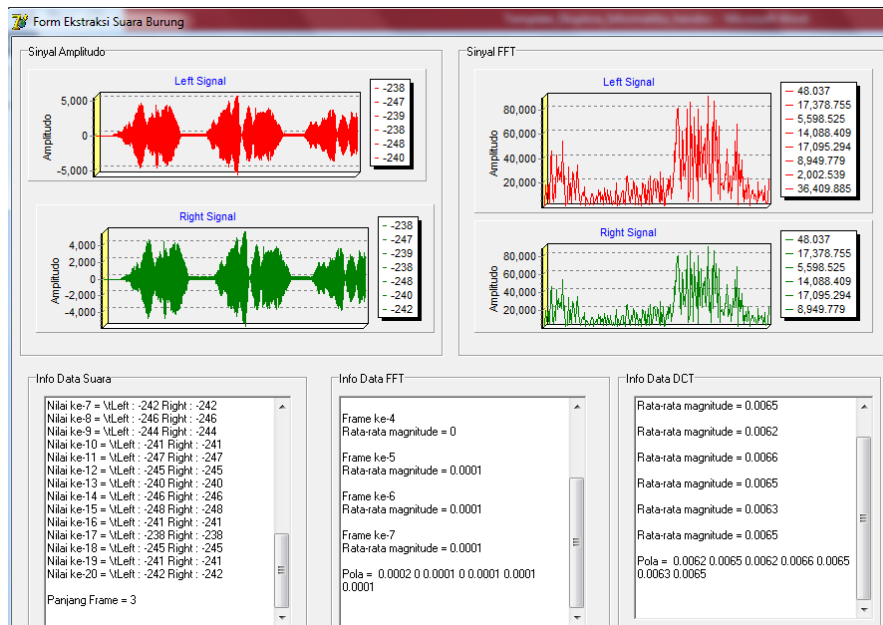
3.1. Hasil Penelitian

Hasil penelitian similaritas suara burung dengan data nilai *DCT* yang pertama dilakukan adalah input suara dengan format *WAV* menjadi nilai *DCT*. Fail suara burung Kenari, Red Lories, dan Beo Merah Meksiko dilakukan proses ekstraksi fitur dengan langkah-langkah berdasarkan Gambar 1.

3.1.1. Ekstraksi Fitur

Ekstraksi fitur suara burung untuk mendapatkan nilai *DCT* dilakukan langkah-langkah yang berdasarkan gambar 1. Input data suara burung Kenari (K), Red Lories (R), dan Beo Merah Meksiko (B) masing-masing data dibagi dengan data S dan data T.

Pembagian data S dan data T berdasarkan waktu suara fail *WAV*, untuk data S dengan waktu satu menit dan data T dengan waktu dua menit. hasil ekstraksi fitur fail suara *WAV* menjadi nilai *DCT* menggunakan aplikasi yang dibuat berdasarkan langkah-langkah pada gambar 1. Untuk mengetahui aplikasi ekstraksi fitur dapat dilihat pada Gambar 3.



Gambar 3. Aplikasi Ekstraksi Fitur Fail Suara

Data-data suara burung yang di-*input*-kan pada aplikasi ekstraksi fitur menghasilkan nilai-nilai *DCT* dengan panjang *frame* berbeda pada fail suara burung yang dua menit (data T) dan satu menit (data S). Untuk data T panjang *string* nilai *DCT* lebih panjang dari pada panjang *string* nilai *DCT* data S. Hasil pada data nilai *DCT* pada masing-masing suara burung dapat dilihat pada tabel 2 untuk data T

Tabel 2. Nilai *DCT* Data T

N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
K	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	06	06	06	06	06	06	06	06	06	06	06	06	06	05	05	06	06	06	06
	2	5	2	6	5	3	5	4	7	8	2	8	7	6	6	4	7	8	8
R	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	05	06	06	06	06	05	05	05	05	05	05	05	05	05	06	06	06	06	07
	1	3	5	7	2	5	1	0	1	0	2	1	1	0	2	3	6	1	1

B	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
	04	06	06	06	06	06	06	05	05	05	05	05	05	05	05	06	06	06	
	8	4	7	8	7	0	5	5	4	3	4	3	2	1	1	2	3	6	8

Pada Tabel 2 sebagai data T yang memiliki panjang *frame* 19. Data T sebagai data target pada metode *levenshtein distance* untuk dibandingkan dengan data S. Hasil perbandingan antara data T dan data S menghasilkan nilai similaritas dalam bentuk persentase. Semakin tinggi nilai persentasi similaritasnya maka dinyatakan data T dan data S dinyatakan mirip atau sama. Hasil ekstraksi fitur suara burung untuk data S dapat dilihat pada tabel 3. Pada tabel 3 panjang *frame* pada setiap jenis suara burung memiliki panjang *frame* yang berbeda-beda. Hal ini dikarenakan dalam waktu satu menit terdapat *noise* (nilai nol) yang terpotong pada proses ekstraksi fitur. Pemotongan nilai ekstraksi fitur disebabkan metode *DCT* bernilai nol akan dihilangkan.

Tabel 3. Nilai *DCT* Data S

N	1	2	3	4	5	6	7	8	9	10	11	12
K1	0.0062	0.0065	0.0062	0.0066	0.0065	0.0063	0.0065					
K2	0.0059	0.0067	0.0064	0.0067	0.0068	0.0056	0.0070					
R1	0.0051	0.0063	0.0065									
R2	0.0051	0.0063	0.0065	0.0067	0.0062	0.0055	0.0058	0.0062	0.0065			
R3	0.0057	0.0065	0.0067	0.0062								
B1	0.0048	0.0064	0.0067	0.0068	0.0067	0.0060	0.0065	0.0055	0.0054	0.0053	0.0063	0.0066
B2	0.0053	0.0063	0.0066	0.0068								

Setelah dilakukan hasil ekstraksi pada data suara T dan data suara S, maka dilakukan perhitungan similaritas menggunakan metode *levenshtein distance*.

3.1.2. Hasil Similaritas

Untuk mendapatkan nilai similaritas suara burung pada masing jenis suara burung yang diteliti antara burung Kenari (K) dengan burung Kenari K1, K2 dan juga suara burung Red Lories dengan suara burung Red Lories R1, R2, R3 sama halnya dengan suara burung Beo Merah Meksiko dengan suara burung B1 dan B2. Untuk mencari nilai similaritas langkah-langkah yang dilakukan sama dengan gambar 2.

Proses metode *levenshtein distance* yang dilakukan pada data T dan data S, menggunakan tiga operasi, untuk penggunaan operasi tersebut dilihat pada panjang *frame string* data T dan data S. Panjang *frame string* data T > panjang data S, maka menggunakan operasi yaitu (1) tidak ada proses penghapusan, (2) operasi pemindahan (PN), dan (3) operasi penambahan (PM). Untuk lebih jelas menggunakan metode *levenshtein distance*, maka dicontohkan pada tabel 4.

Tabel 4. Penggunaan Metode *Levenshtein Distance*

N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
B	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	04	06	06	06	06	06	06	05	05	05	05	05	05	05	05	05	06	06	06
	8	4	7	8	7	0	5	5	4	3	4	3	2	1	1	2	3	6	8
B	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0							
I	04	06	06	06	06	06	05	05	05	05	06	06							
	8	4	7	8	7	0	5	5	4	3	3	6							
O							PN				P	P	P	P	P	P			P
											M	M	M	M	M	M			M

Hasil perhitungan similaritas pada tabel 4 menggunakan persamaan 4, dicari nilai $dis(x,y)$ dari nilai jarak data T dan data S dari jumlah operasi pemindahan (PN) dan penambahan (PM). Pengisian nilai panjang data T dan panjang data S. Hasil perhitungan similaritas untuk semua pengujian dapat dilihat pada tabel 5.

Tabel 5. Hasil Similaritas

Data T/S	Panjang T	Panjang S	Dis(x,y)	Max(S,T)	Similaritas
B/B1	19	12	8	19	58%
B/B2	19	4	16	19	16%
K/K1	19	7	12	19	37%
K/K2	19	7	13	19	32%
R/R1	19	3	16	19	16%
R/R2	19	9	13	19	32%
R/R3	19	4	15	19	21%

3.1.3. Grafik Similaritas Data T dan Data S

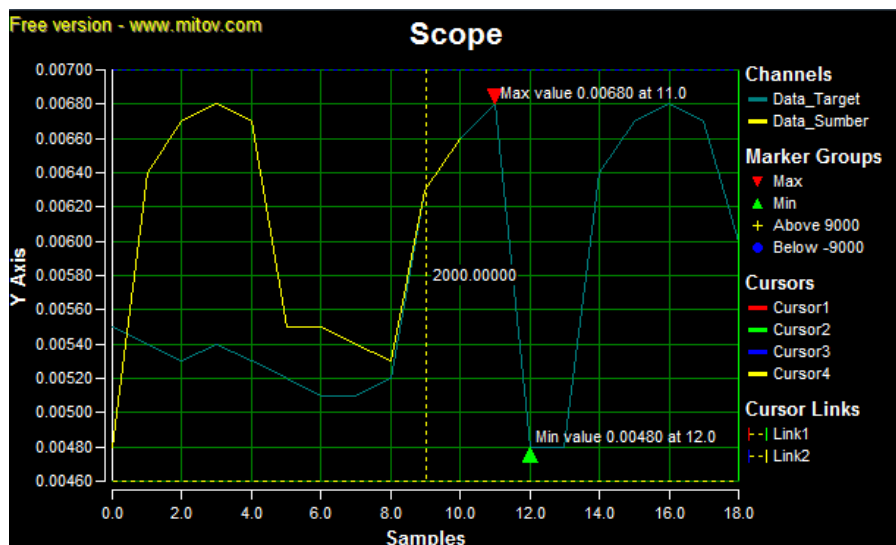
Hasil yang didapat di dalam penelitian yang sudah ditampilkan pada tabel 5, maka untuk jelasnya ditampilkan dalam bentuk grafik. Hasil yang didapat pada tabel 5 similiaritas suara burung dengan nilai yang tidak ada di atas 60%, akan tetapi memiliki masih memiliki similaritas pada suara yang ada pada data T dan data S. Untuk mengetahui similaritas suara burung yang ada pada *string* nilai *DCT* dilihat pada pola grafik.

Pola grafik nilai *DCT* pada similiaritas suara burung menggunakan sebuah aplikasi grafik yang menggunakan grafik garis (*line graphic*). Di dalam aplikasi grafik terdapat dua grafik yaitu (1) grafik garis data T dan (2) grafik garis data S. Grafik similaritas di contohkan dengan data pada tabel 4 burung Beo Merah Meksiko. Hasil pada aplikasi grafik similaritas dapat dilihat pada gambar 4.



Gambar 4. Aplikasi Grafik Similaritas Pada Data T dan Data S

Pada Gambar 4. menunjukkan adanya pola yang sama pada suara burung Beo Merah Meksiko (B) pada data S dengan panjang *string* 12 *frame* dengan suara burung Beo Merah Meksiko (B1) pada data T dengan panjang *string* 19 *frame*. Kesamaan pola grafik pada *frame* 1 sampai dengan 6 pada data S. Dan similaritas pola suara burung juga ditampilkan *frame* 11 sampai dengan 12 pada data S yang dapat dilihat pada Gambar 5.



Gambar 5. Similaritas Pola Grafik Suara Burung

3.2. Pembahasan Hasil

Hasil yang telah didapat dari penelitian similaritas suara burung dengan data ekstraksi fitur menggunakan *DCT*, maka terdapat nilai similaritas suara burung data T dan data S pada masing-masing jenis suara burung Kenari, Red Lories dan Beo Merah Meksiko. Pada data jenis suara burung Kenari dengan panjang *string* nilai *DTC 19 frame* di data T dibandingkan dengan suara burung kenari dengan panjang *string* nilai *DTC 7 frame* di data S didapat hasil similaritas 37% dan 32%, maka hasil tersebut terdapat pola yang sama pada suara burung tersebut yang bisa menunjukkan bahwa ada similaritas suara burung Kenari pada data T. Kesamaan pola tersebut berdasarkan urutan data nilai *DCT* yang menunjukkan pola grafik yang sama.

Hal tersebut juga bisa dilihat pada contoh Gambar 4 dan Gambar 5 yang menunjukkan suatu pola similaritas suara burung yang dicontohkan pada suara burung Beo Merah Meksiko. Pada data T untuk suara burung Beo Merah Meksiko dengan panjang *string* nilai *DTC 19 frame*, ada saatnya di posisi *frame* tersebut memiliki similaritas pola yang sama dengan data S suara burung Beo Merah Meksiko.

Hasil similaritas yang tidak bisa mencapai di atas 60% dikarenakan panjang *string* pada data T dan data S tidak sama, atau waktu data suara burung yang tidak sama. Hasil ini yang dapat mempengaruhi hasil similaritas. Akan tetapi tingkat keberhasilan dalam penelitian ini adalah dapat mengetahui pola similaritas suara burung pada tiap-tiap panjang *frame* yang ditunjukkan di dalam grafik.

4. Kesimpulan

Pada penelitian similaritas suara burung dengan menggunakan metode *levenshtein distance* pada ekstraksi fitur suara burung menggunakan *DCT*, dihasilkan similaritas pada masing-masing jenis suara burung yang dibagi menjadi data T dan data S. Hasil yang didapat similaritas pada jenis suara burung Kenari pada data T dan suara burung Kenari pada data S memiliki similaritas 37% dan 32%. Jenis suara burung Red Lories pada data T dan suara burung Red Lories pada data S memiliki nilai similaritas 16%, 32% dan 21% dan Suara burung Beo Merah Meksiko tingkat similaritas pada data T dan data S memiliki nilai 58% dan 16%. Hasil yang didapat penelitian ini bisa untuk ditingkatkan pada penelitian selanjutnya seperti klasifikasi atau klastering pada jenis suara burung.

Daftar Pustaka

- [1] C. Jarne, "MethodsX A method for estimation of fundamental frequency for tonal sounds inspired on bird song studies," *MethodsX*, vol. 6, pp. 124–131, 2019.
- [2] H. Nugroho, W. Widodo, and A. Rachman, "Pattern recognition bird sounds based on their type using discrete cosine transform and gaussian methods," vol. 4, no. 3, pp. 233–240, 2019.
- [3] J. Xie and M. Zhu, "PT US CR," *Ecol. Inform.*, 2019.
- [4] T. Tuncer, E. Akbal, and S. Dogan, "Multileveled ternary pattern and iterative ReliefF based bird sound classification," *Appl. Acoust.*, vol. 176, p. 107866, 2021.
- [5] X. Zhang, A. Chen, G. Zhou, Z. Zhang, X. Huang, and X. Qiang, "Ecological Informatics Spectrogram-*frame* linear network and continuous *frame* sequence for bird sound classification," *Ecol. Inform.*, vol. 54, no. August, p. 101009, 2019.
- [6] R. Pahuja and A. Kumar, "Sound-spectrogram based automatic bird species recognition using MLP classifier," *Appl. Acoust.*, vol. 180, p. 108077, 2021.
- [7] M. Ben Lazreg, M. Goodwin, and O. Granmo, "Computer Speech & Language Combining a context aware neural network with a denoising autoencoder for measuring *string* similarities," *Comput. Speech Lang.*, vol. 60, p. 101028, 2020.
- [8] N. Fadhillah, "Validasi Pencarian Kata Kunci Menggunakan Algoritma *Levenshtein Distance* Berdasarkan Metode Approximate *String* Matching," vol. 3, no. 2, pp. 3–7, 2018.
- [9] *Pengenalan Pola , Aplikasi untuk pengenalan wajah, analisis tekstur objek, pengenalan pola plat nomor kendaraan, dan segmentasi pembuluh darah*. Yogyakarta: Graha Ilmu.
- [10] J. Abreu and J. R. Rico-juan, "Characterization of contour regularities based on the *Levenshtein edit distance*," *Pattern Recognit. Lett.*, vol. 32, no. 10, pp. 1421–1427, 2011.
- [11] B. Indonesia, M. Metode, and N. *Levenshtein*, "Identifikasi Kesalahan Penulisan Kata (Typographical Error) pada Dokumen Identifikasi Kesalahan Penulisan Kata (Typographical Error) pada Dokumen Berbahasa Indonesia Menggunakan Metode N-gram dan *Levenshtein Distance*," no. August, 2017.
- [12] J. L. B. Y.R. Prayogi, "Identifikasi parameter optimal Gaussian Mixture Model pada Identifikasi pembicara di lingkungan berdearau menggunakan residu deteksi Endpoin," *J. Ilm. Teknol. Inf.*,

-
- vol. 13, no. 2, pp. 198–206, 2015.
- [13] Manan Vyas, “A Gaussian Mixture Model Based Speech Recognition System Using Matlab,” *Signal Image Process. J. (SIPJ)*, vol. 4, no. 4, pp. 109–118, 2013.
- [14] R. D. A. Donanda Khabi Putra, Iwan Iwut. T, “Simulasi dan Analisis speaker recognition menggunakan Metode *Mel* Frequency Cepstrum Coefficient (MFCC) dan Gaussian Mixtrue Model (GMM),” *e-proceeding Eng.*, vol. 4, no. 2, pp. 1766–1772, 2017.
- [15] P. V. P. Upadhyaya, O. Farroq, M.R Abidi, “Comparative Study of Visual Feature for Bimodal Hindi Speech Recognition,” *Arch. Accoustics PAN*, vol. 40, no. 4, pp. 609–619, 2015.
- [16] X. Cheng and Q. Duan, “Speech Emotion Recognition Using Gaussian Mixture Model,” in *2012 The 2nd International Conference on Computer Application and System Modeling*, 2013, pp. 1222–1225.
- [17] Sukhdeep Kaur, ER. Gurwinder Kaur, “Enhancement of Speech Recognition Algorithm Using *DCT* and Inverse *WAVE* Transformation,” *ournal Eng. Res. Appl.*, vol. 3, no. 6, pp. 749–754, 2013.
- [18] F. Suherdiansyah, “Klasisfikasi gerak bibir berdasarkan pola suara menggunakan metode *Mel*-Frequency Cepstrum Coefficients(MFCC) dan Hiden Markov Model (HMM) untuk Mengenal kata sederhana,” 2019.