

# Penerapan Algoritma *Time-Warped Longest Common Subsequence* dalam Pengaksesan Berkas Musik

Vicky Nugraha Putra<sup>1</sup>, Raymond Ralibi<sup>2</sup>

Laboratorium Ilmu dan Rekayasa Komputasi  
Departemen Teknik Informatika, Institut Teknologi Bandung  
Jl. Ganेशha 10, Bandung

E-mail : [if14129@students.if.itb.ac.id](mailto:if14129@students.if.itb.ac.id)<sup>1</sup>, [if14153@students.if.itb.ac.id](mailto:if14153@students.if.itb.ac.id)<sup>2</sup>

## Abstrak

Pada saat ini musik menjadi elemen yang tidak dapat dipisahkan dari kehidupan sebagian besar manusia. Bersamaan dengan itu pula, teknologi yang berhubungan dengan musik berkembang sangat pesat. Sekarang, pengaksesan terhadap berkas musik atau audio dapat dilakukan hanya dengan menyanyikannya melalui *microphone*. Namun, kesalahan saat menyanyikan lagu atau saat perekaman suara sering kali menyebabkan ketidaksesuaian dengan berkas lagu yang sebenarnya. Oleh karena itu, perlu adanya suatu algoritma yang dapat diterapkan untuk menyelaraskan kesalahan-kesalahan tersebut. Dalam makalah ini, akan dibahas algoritma yang dapat diterapkan dalam permasalahan ini, yaitu *Time-Warped Longest Common Subsequence*.

**Kata kunci:** pengaksesan berkas musik, algoritma, *time-warped longest common subsequence*

## 1. Pendahuluan

Saat ini, berkas musik cukup mudah untuk didapatkan. Hal ini disebabkan banyaknya berkas musik yang dapat ditemui di dunia internet dengan berbagai jenis, ukuran, dan format. Perkembangan ini juga menuntut kemudahan dalam pengaksesan berkas musik. Sekarang ini, kita dapat mengakses berkas musik dengan mengetikannya melalui *keyboard*, atau bahkan menyanyikannya melalui *microphone* [1,4]. Namun, fasilitas ini kadang mengalami beberapa kendala, seperti kesalahan perekaman, ketidaktepatan irama yang dinyanyikan, atau perbedaan tempo atau kecepatan. Untuk itu diperlukan suatu algoritma yang dapat memperhitungkan kesalahan-kesalahan tersebut. Algoritma yang akan diterapkan menggunakan dasar pencocokan string untuk menentukan kemiripan dari dua buah berkas musik.

Musik disusun oleh serangkaian simbol. Dalam permasalahan ini, kita menggunakan pendekatan dengan musik *monophonic*, yang notabene telah berbentuk sebuah string yang dibentuk dari sederetan alfabet yang mewakili nada-nada dalam musik tersebut [3]. Dengan menganggap potongan musik sebagai string dari simbol-simbol, maka kita dapat memanfaatkan dasar algoritma pencocokan string untuk menentukan kemiripan berkas musik [5].

Dalam makalah ini, akan dibahas beberapa algoritma yang dapat diterapkan dalam pengaksesan berkas musik. Masing-masing algoritma tersebut adalah *Dynamic Time Warping*, *Longest Common Subsequence*, dan kombinasi dari keduanya *Time-Warped Longest Common Subsequence*.

## 2. Algoritma *Dynamic Time Warping*

*Dynamic Time Warping* (DTW) adalah sebuah algoritma yang dapat menentukan kemiripan dua buah rangkaian waktu, klasifikasinya, serta menemukan bagian-bagian yang sesuai. Algoritma ini banyak digunakan untuk mengenali apakah dua buah frase yang diucapkan sama atau tidak. DTW memiliki waktu kuadratik serta kompleksitas yang membatasi penggunaannya hanya untuk rangkaian atau himpunan waktu yang kecil.

Algoritma *Dynamic Time Warping* mencari jalur optimal melalui sebuah matriks *point* yang menunjukkan kemungkinan penyesuaian waktu antara dua buah rangkaian waktu (*time series*). DTW bekerja sebagai berikut. Diberikan dua buah rangkaian waktu,

$$X = (x_1, x_2, \dots, x_m),$$

$$Y = (y_1, y_2, \dots, y_n)$$

yang mengisi sebuah matriks  $m \times n$  yang menunjukkan jarak dari jalur yang paling mungkin dengan menggunakan prinsip rekursif.

$$D(i, j) = d(i, j) + \min \begin{cases} D(i, j - 1) \\ D(i - 1, j) \\ D(i - 1, j - 1) \end{cases} \quad (1)$$

dimana  $1 \leq i \leq m$ ,  $1 \leq j \leq n$ ,  $d(i, j)$  menunjukkan jarak antara  $x_i$  dan  $y_j$ .

Algoritma *Dynamic Time Warping* dapat menangani perubahan berkas, baik pengembangan maupun pengompresan. Dengan demikian, penggunaan algoritma DTW dapat menanggulangi permasalahan tentang tempo serta kecepatan nada yang dinyanyikan, sehingga dapat disesuaikan dengan berkas musik aslinya.

### 3. Algoritma Longest Common Subsequence

Algoritma *Longest Common Subsequence* (LCS) mencari upa-rangkaian (*subsequence*) yang sama pada dua buah rangkaian (*sequence*), tanpa mempedulikan panjang dan jumlah simbol yang tidak sesuai. Contoh : "abcdefg" dan "axbdyeyzzzz" memiliki panjang empat *sequence*, yaitu "abde" sebagai *subsequence* sama terpanjang.

Algoritma *Longest Common Subsequence* bekerja sebagai berikut. Diberikan dua buah *sequence*,

$$\begin{aligned} X &= (x_1, x_2, \dots, x_m), \\ Y &= (y_1, y_2, \dots, y_n) \end{aligned}$$

Untuk mencari sebuah *sequence* Z, yang merupakan *subsequence* sama terpanjang dari kedua *sequence* tersebut, kita mendefinisikan,

$$Z = (z_1, z_2, \dots, z_k)$$

dimana terdapat  $(i_1, i_2, \dots, i_k)$  yang merupakan indeks dari X, sehingga untuk semua  $j = 1 \dots k$ ,  $x_{i_j} = z_j$  [2].

Solusi permasalahan LCS menggunakan persamaan rekursif, dimana *cost* untuk hasil operasi disimpan di c.

$$c(i, j) = \begin{cases} 0 & \text{if } i = 0 \text{ or } j = 0 \\ c(i - 1, j - 1) + 1 & \text{if } i, j > 0 \text{ and } x_i = Y_j \\ \max[c(i, j - 1), c(i - 1, j)] & \text{if } i, j > 0 \text{ and } x_i \neq Y_j \end{cases} \quad (2)$$

Keuntungan menggunakan algoritma LCS untuk mengukur kemiripan anatar dua *sequence*, yaitu kita dapat membandingkan dua *sequence* yang tidak sama panjang dan memiliki deretan simbol yang tidak sesuai. Dengan menggunakan algoritma LCS, kita dapat menyanyikan hanya sebagian lagu saja dan gangguan saat perekaman dapat ditanggulangi.

### 4. Algoritma Time-Warped Longest Common Subsequence

Ketika berkas musik mengalami perubahan, baik dikembangkan maupun dikompresi, kita menginginkan berkas tersebut tetap dikenali sebagai berkas yang sama seperti sebelumnya. Contoh : kasus I adalah jika *sequence* "44556677" dibandingkan dengan *sequence* "4567", maka algoritma LCS akan memberikan nilai 4, karena kedua *sequence* tersebut sama-sama memiliki *subsequence* "4567". Kasus II adalah jika yang dibandingkan adalah *sequence* "42536172" dan *sequence* "4567", maka algoritma LCS tetap memberikan hasil 4, karena *subsequence* yang sama-sama dimiliki adalah "4567". Namun, karena pada kasus I masing-masing *sequence* adalah pengembangan atau pengompresan dari satu sama lain, seharusnya pada kasus I memiliki tingkat kemiripan yang lebih tinggi dibandingkan dengan kasus II. Algoritma LCS tidak dapat mengenali perbedaan ini.

Seperti yang diketahui sebelumnya, bahwa algoritma *Dynamic Time Warping* (DTW) dapat menangani berkas yang telah mengalami perubahan, baik pengembangan maupun pengompresan, tapi tidak mampu menangani berkas yang telah mengalami penyisipan maupun penghapusan sebagian berkas yang sangat mungkin terjadi saat perekaman musik, sehingga berkas musik yang digunakan harus utuh. Sebaliknya, algoritma LCS dapat menangani berkas yang tidak utuh, tapi tidak mampu menangani berkas yang telah mengalami pengembangan atau pengompresan sebelumnya.

Dengan mengambil kelebihan dari masing-masing algoritma tersebut, dibuat sebuah algoritma baru, yaitu *Time-Warped Longest Common Subsequence* (T-WLCS) yang dapat menangani kedua permasalahan tersebut. Algoritma T-WLCS bekerja sebagai berikut.

$$c(i, j) = \begin{cases} 0 & \text{if } i = 0 \text{ or } j = 0 \\ \max[c(i, j - 1), c(i - 1, j), c(i - 1, j - 1)] + 1 & \text{if } i, j > 0 \text{ and } x_i = Y_j \\ \max[c(i, j - 1), c(i - 1, j)] & \text{if } i, j > 0 \text{ and } x_i \neq Y_j \end{cases} \quad (3)$$

Disini  $c(i, j)$  merupakan *cost* minimum dari jalur T-WLCS dengan masukan  $i, j$  dalam tabel T-WLCS yang berukuran  $n \times m$ , yang menyimpan jalur dari *cost* minimum yang menunjukkan pada *alignment* yang memungkinkan sejauh ini. Jalur optimal dalam T-WLCS adalah yang menerima *cost alignment* minimum. Nilai ini disimpan dalam  $c(m, n)$ .

## 5. Uji Kasus dan Perbandingan

**Kasus I :** Diberikan dua buah string,  $S1 = "41516171"$ , dan  $S2 = "4567"$ . Jika kedua string tersebut dibandingkan dengan algoritma LCS, maka  $LCS(S1,S2) = 4$ , jika diterapkan algoritma T-WLCS, maka  $T-WLCS(S1,S2) = 4$ . Dalam kasus ini, kedua algoritma tersebut memberikan hasil yang sama, karena kedua string tersebut memiliki *subsequence* yang sama, yaitu "4567". Tabel pencarian solusi T-WLCS dapat dilihat pada Gambar 1(a).

**Kasus II :** Diberikan dua buah string,  $S1 = "44556677"$ , dan  $S2 = "4567"$ . Jika kedua string tersebut dibandingkan dengan algoritma LCS, maka  $LCS(S1,S2) = 4$ , jika diterapkan algoritma T-WLCS, maka  $T-WLCS(S1,S2) = 8$ . Dalam kasus ini, algoritma T-WLCS memberikan hasil yang lebih baik dibandingkan dengan kasus I, pada kasus ini kemiripan dinilai lebih tinggi karena  $S1$  hanya merupakan penggandaan dari elemen-elemen  $S2$ . Tabel pencarian solusi T-WLCS dapat dilihat pada Gambar 1(b).

**Kasus III :** Diberikan dua buah string,  $S1 = "4455661111177"$ , dan  $S2 = "4567"$ . Jika kedua string tersebut dibandingkan dengan algoritma LCS, maka  $LCS(S1,S2) = 4$ , jika diterapkan algoritma T-WLCS, maka  $T-WLCS(S1,S2) = 8$ . Kasus ini menunjukkan bahwa algoritma T-WLCS mengadopsi kelebihan algoritma LCS, yaitu dapat mengatasi gangguan penyisipan atau penghapusan. Tabel pencarian solusi T-WLCS dapat dilihat pada Gambar 1(c).

		Query Sequence						
		4	4	5	6	7	7	7
Stored Sequence	4	1	1	1	1	1	1	1
	5	1	1	2	2	2	2	2
	6	1	1	2	2	3	3	3
	7	1	1	2	2	3	3	4

(a)

		Query Sequence						
		4	4	5	5	6	6	7
Stored Sequence	4	1	2	2	2	2	2	2
	5	1	2	3	4	4	4	4
	6	1	2	3	4	5	6	6
	7	1	2	3	4	5	6	7

(b)

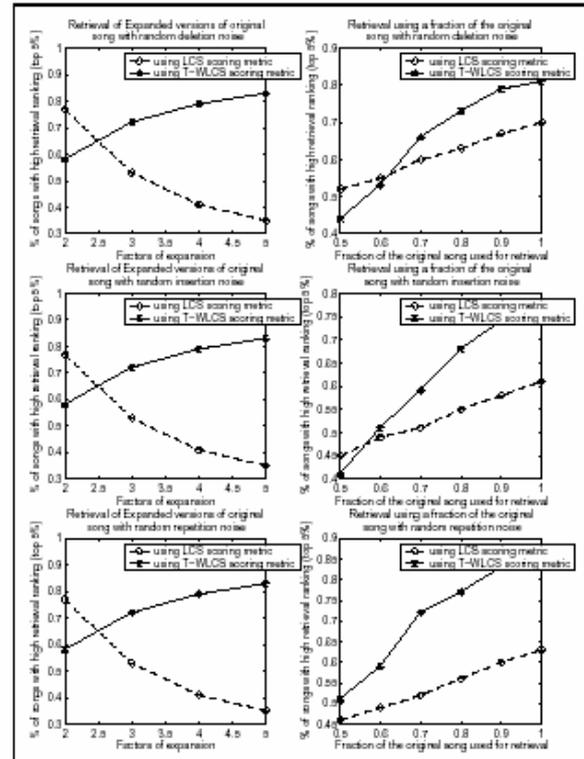
		Query Sequence											
		4	4	5	5	6	6	1	1	1	1	7	7
Stored Sequence	4	1	2	2	2	2	2	2	2	2	2	2	2
	5	1	2	3	4	4	4	4	4	4	4	4	4
	6	1	2	3	4	5	6	6	6	6	6	6	6
	7	1	2	3	4	5	6	6	6	6	6	7	8

(c)

**Gambar 1.** Tabel T-WLCS yang berisi  $c(i,j)$  untuk kasus I, II, III

Uji perbandingan dengan melakukan percobaan memberikan hasil bahwa pengaksesan berkas musik dengan menggunakan algoritma *Time-Warped Longest Common Subsequence* (T-WLCS) untuk menghitung tingkat kemiripan sebuah potongan lagu dengan setiap lagu yang terdapat dalam *database* memiliki efektivitas

yang lebih tinggi dibandingkan dengan penggunaan algoritma *Longest Common Subsequence* (LCS) biasa [6].



**Gambar 2.** Uji perbandingan antara algoritma LCS dan algoritma T-WLCS dalam pengaksesan berkas musik

## 6. Kesimpulan

Pengaksesan berkas musik dapat dilakukan dengan menghitung serta membandingkan kemiripan dua buah berkas musik yang direpresentasikan dalam *sequence* atau string.

Algoritma *Dynamic Time Warping* (DTW) dapat digunakan untuk menyesuaikan berkas musik yang telah mengalami pengembangan atau pengompresan dengan berkas aslinya.

Algoritma *Longest Common Subsequence* (LCS) dapat digunakan untuk menghitung kemiripan dua buah potongan berkas musik, serta dapat menangani penyisipan atau penghapusan beberapa elemen berkas yang mungkin terjadi saat perekaman berkas musik.

Algoritma *Time-Warped Longest Common Subsequence* (T-WLCS) merupakan algoritma yang mengadopsi kelebihan algoritma DTW dan LCS dan menggabungkannya. Dalam persoalan pengaksesan berkas musik, algoritma T-WLCS memiliki efektivitas lebih tinggi daripada algoritma LCS.

## Daftar Pustaka

- [1] H. Sakoe & S. Chiba. *Dynamic Programming Algorithm Optimization for Spoken Word Recognition*. *IEEE Trans. Acoustics, Speech and Signal Processing*, ASSP-26(1):43–49, 1978.
- [2] T. Cormen, C. Leiserson, & R. Rivest. *Introduction To Algorithms*. The MIT Press, Cambridge, MA, 1990.
- [3] D. Gusfield. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, 1997.
- [4] T. Kageyama, K. Mochizuki, and Y. Takashima. *Melody retrieval with humming*. In *ICMC Proceedings 1993*, 1993.
- [5] K. Lemstrom. *String Matching Techniques for Music Retrieval*. PhD thesis, University of Helsinki, 2000.
- [6] Guo, AnYuan, Hava Siegelmann. *Time-Warped Longest Common Subsequence Algorithm for Music Retrieval*. University of Massachusetts, Amherst Department of Computer Science, 2003.