Trajectories - 6

by Mohammad Jamhuri

Submission date: 15-Mar-2021 06:28PM (UTC-0700)
Submission ID: 1534116316
File name: Trayektori_6.pdf (14M)
Word count: 4162
Character count: 23707

SIMILARITY ANALYSIS OF USER TRAJECTORIES BASED ON EUCLIDEAN DISTANCE AND NEEDLEMAN WUNSCH ALGORITHM

(Center, Times New Roman 18 **not Bold**, maximum. **10 words in English**)

Mohammad Jamhuri[1,2], Mohammad Isa Irawan[1]

[1]Departemen Matematika, Fakultas Sains dan Analitika Data, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

[2]Jurusan Matematika, Fakultas Sains dan Teknologi, UIN Maulana Malik Ibrahim, Malang, Indonesia

Email Correspondence: *[1]mii@its.ac.id, [2]m.jamhuri@live.com

Received: Accepted: Published:

Abstrak: Dalam tulisan ini kami membahas cara menghitung persentase kesamaan dari dua buah lintasan menggunakan algoritma Needleman Wunsch. Tahapan-tahapan yang dilakukan adalah menginterpolasi lintasan, menghitung jarak titik-titik koordinat dari kedua lintasan, mengidentifikasi jarak yang ekivalen, mengubah lintasan menjadi sekuens huruf alfabet, menyejajarkan sekuens, dan menentukan besarnya kesamaan berdasarkan hasil penyejajaran. Kesamaan yang diperoleh dari metode penyejajaran dibandingkan secara langsung dengan panjang jalur yang dilalui bersama oleh kedua lintasan, hasil perhitungan menunjukkan bahwa akurasi metode penyejajaran mencapai lebih dari 90%.

Kata kunci—3-5 kesamaan lintasan, interpolasi linier, jarak Haversine, penyejajaran global

Abstract: *In this paper, we discuss how to calculate the similarity of two trajectories using the Needleman Wunsch algorithm. The calculation steps are interpolating the trajectory, calculating the distance between the coordinates of the trajectory, identifying the equivalent distance, transforming trajectories into a sequence of letters of the alphabet, aligning the sequences, and measuring the magnitude of similarity based on the alignment results. The similarity obtained is compared directly to the length of the trajectories shared by the two lines, the calculation results show that the accuracy of the alignment method reaches more than 90%.*

Keywords—*3-5 similarity of trajectories, linear interpolation, Haversine distance, global alignment*

1 PENDAHULUAN

Segala sesuatu yang berubah mengikuti pergerakan waktu, baik posisi ataupun nilainya dapat dinamakan sebagai objek bergerak. Mobil yang berjalan di jalan raya, pesawat yang sedang mengudara, roket yang ditembakkan, perjalanan satelit, perubahan harga saham, kenaikan dan penurunan suhu dalam suatu wilayah tertentu merupakan contoh-contoh objek bergerak. Merekam pergerakan benda tersebut akan menghasilkan sekumpulan data yang berupa titik-titik koordinat ataupun nilai-nilai yang disebut sebagai lintasan. Lintasan merupakan histori pergerakan suatu objek yang dinyatakan dalam himpunan pasangan terurut $\{r_1, r_2, \dots, r_m\}$ yang menyatakan lokasi ataupun nilai pada waktu tertentu $\{t_1, t_2, \dots, t_m\}$, sehingga sebuah lintasan dapat dinyatakan sebagai $r = \{(r_1, t_1), (r_2, t_2), \dots, (r_m, t_m)\}$ dengan m merupakan banyaknya titik yang diobservasi (Toohey & Duckham, 2015).

Besarnya kesamaan suatu lintasan menjadi masalah penting dalam beberapa domain aplikasi. Misalnya, analisis lintasan pelanggan di supermarket dapat digunakan untuk mengatur peletakan produk dalam rak. Contoh lain adalah untuk menemukan pola migrasi yang tak biasa dari sekelompok burung yang mencurigakan atau keberadaan pola lintasan yang langka. Domain lain seperti analisis stok dan data, sistem rekomendasi rute perjalanan, sistem rekomendasi untuk teman berdasarkan minat, hoby, lokasi yang dikunjungi, dll. Kesamaan lintasan dapat juga digunakan untuk memprediksi fenomena di masa mendatang seperti arah mata angin, badai, dan gempa bumi (Wang et al., 2013).

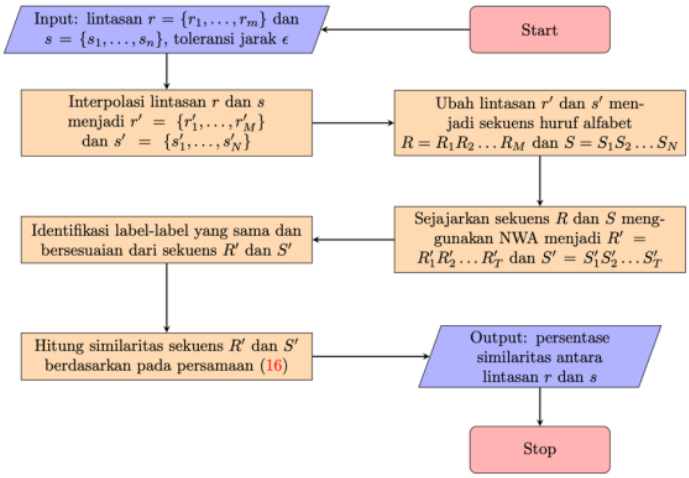
Salah satu cara untuk menghitung kesamaan dari dua buah lintasan adalah dengan menggunakan metode penyejajaran sebagaimana yang dilakukan oleh Cavojsky dalam (Čavojský et al., 2020). Cavojsky dkk menggunakan algoritma Needleman Wunsch untuk menyejajarkan lintasan yang diubah kedalam bentuk sekuens huruf alfabet terlebih dahulu. Sabarish dkk, mengubah lintasan kedalam bentuk graph dan mengukur kesamaan lintasan berdasarkan edge dan vertex (Sabarish et al., 2020). Perhitungan kesamaan lintasan dapat dilakukan dengan menggunakan persamaan hampiran yang dibangun menggunakan model regresi, ataupun interpolasi. Magdy dkk, menggunakan tiga pendekatan untuk mengukur kesamaan dari lintasan-lintasan, yaitu regresi, interpolasi, dan curve barcoding (Magdy et al., 2018).

Sebelumnya Garhwal dan Yan (Garhwal & Yan, 2019) menggunakan algoritma Needleman Wunsch untuk mendeteksi watermark dalam gambar, mereka melaporkan bahwa metode tersebut dapat mengenali watermark dalam beberapa data gambar dengan akurasi sampai100%. Ju dkk (Ju et al., 2018) menggunakan algoritma tersebut untuk mengidentifikasi penggunaan kartu pintar dan menghitung skor kesamaan transaksi untuk mencari hubungannya dengan prestasi siswa. Chua and Foo (Chua & Foo, 2015) menggunakan algoritma tersebut untuk mendeteksi aktifitas mencurigakan pada perangkat rumah pintar dan membandingkan hasilnya dengan hasil yang diperoleh dari decision tree. Cavojsky and Drozda (Čavojský & Drozda, 2019) mengatakan bahwa sebenarnya algoritma Needleman Wunsch (Needleman & Wunsch, 1970) ditujukan untuk penyejajaran DNA, RNA, dan sekuens protein, tetapi dapat pula digunakan untuk penyejajaran lintasan.

Algoritma Needleman-Wunsch awalnya di desain untuk penyejajaran sekuens yang berupa huruf abjad (Beretta, 2018), sehingga untuk menyejajarkan lintasan yang berupa titik-titik koordinat (sekuens numerik) perlu dilakukan penyesuaian terlebih dahulu dengan cara mentrasnformasi kedalam bentuk sekuens huruf alfabet. Sedangkan untuk mengubah sekuens numerik kedalam bentuk huruf abjad diperlukan perhitungan jarak, diantaranya dapat menggunakan jarak Euclid (Čavojský et al., 2020), jarak Manhattan (Chen et al., 2005) ataupun jarak Haversine (Anisya & Swara, 2017). Perhitungan jarak diperlukan untuk mengidentifikasi titik-titik koordinat yang ekivalen pada kedua lintasan. Berikutnya dilakukan pelabelan terhadap titik-titik koordinat menggunakan karakter yang sama jika koordinat ekivalen dan dengan karakter yang berbeda jika koordinat tidak ekivalen. Untuk meningkatkan akurasi dari kesamaan lintasan, dapat dilakukan interpolasi terhadap koordinat-koordinat dari kedua lintasan menggunakan interpolasi linier sebelum dilakukan transformasi.

2 METODE

Kajian utama dalam tulisan ini adalah bagaimana menentukan tingkat kesamaan antara dua buah lintasan. Misalkan r dan s adalah dua buah lintasan yang direkam berdasarkan posisi pengguna dalam bentuk himpunan titik-titik koordinat *latitude longitude*, maka r dan s dapat di tuliskan sebagai himpunan terurut $r = \{r_1, \dots, r_m\}$ dan $s = \{s_1, \dots, s_n\}$, dengan $r_i = (r_{i,x}, r_{i,y})$ dan $s_j = (s_{j,x}, s_{j,y})$ untuk $i = 1, \dots, m$ dan $j = 1, \dots, n$. Untuk menghitung similaritas kedua lintasan tersebut, langkah-langkah penyelesaiannya secara garis besar dapat dilihat pada diagram alir dalam Gambar 1 berikut:



Gambar 1. Diagram Alir untuk Menentukan Similaritas Dua Buah Lintasan

Langkah-langkah penyejajaran lintasan r dan s pada Gambar 1 diatas adalah 1) interpolasi; dilakukan untuk menambahkan titik-titik koordinat baru diantara titik-titik koordinat yang telah ada pada lintasan, 2) transformasi; mengubah barisan numerik titik-titik koordinat dalam lintasan menjadi barisan huruf abjad, 3) penyejajaran; menyusun karakter-karakter yang bersesuaian atau menyisipkan karakter pengganti sedemikian hingga kolom-kolom dari masing-masing sekuens memuat karakter yang identik, (4) menghitung jumlah karakter yang bersesuaian atau identik, (5) menghitung similaritas lintasan. Konsep penting dalam interpolasi dan transformasi lintasan adalah jarak antar titik, yang digunakan untuk menentukan banyaknya titik interpolasi serta untuk mengidentifikasi titik-titik koordinat yang ekivalen.

2.1 Jarak antar Titik

Jarak antara dua buah titik yang berupa koordinat *latitude-longitude* dapat di hitung menggunakan persamaan Haversine yang diperkenalkan oleh James Inman pada tahun 1835 (Anisya & Swan 2017). Yaitu, jika diberikan dua buah pasangan terurut berupa koordinat geografis $A = (x_a, y_a)$ dan $B = (x_b, y_b)$ dengan x_a, x_b dan y_a, y_b secara berturut-turut adalah *latitude* dan *longitude*, maka jarak antara A dan B di definisikan sebagai

$$d(A,B) = 2R \arcsin \sqrt{\sin^2\left(\frac{x_a - x_b}{2}\right) + \cos x_a \cos x_b \sin^2\left(\frac{y_a - y_b}{2}\right)} \qquad (1)$$

dengan R adalah jari-jari bumi, yaitu ± 6.371 km. Selain menggunakan formula Haversine, jarak antara A dan B dapat dihitung menggunakan jarak Euclid sebagaimana pada (Čavojský et al., 2020), yaitu

$$d(A,B) = \sqrt{(x_b - x_a)^2 + (y_b - y_a)^2}. \qquad (2)$$

Jarak yang diperoleh dari persamaan (2) masih dalam bentuk geodesi, sehingga untuk mengubah kedalam satuan kilometer, harus dikalikan dengan jari-jari bumi R. Cara lain yang hampir sama adalah menggunakan jarak Manhattan sebagaimana yang digunakan dalam Chen, dkk (Chen et al., 2005), yaitu

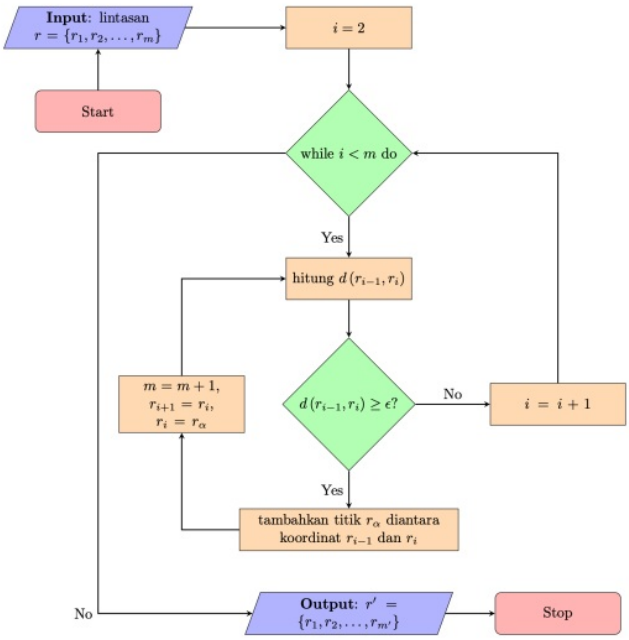
$$d(A,B) = |x_b - x_a| + |y_b - y_a|. \qquad (3)$$

2.2 Interpolasi Lintasan

Interpolasi linier merupakan metode yang digunakan untuk mencari nilai dari sebuah titik yang tidak diketahui dari dua buah titik yang membentuk garis linier yang sudah diketahui terlebih dahulu (Chapra & Canale, 1998). Interpolasi lintasan merupakan interpolasi linier yang dilakukan terhadap dua buah titik koordinat berurutan dalam suatu lintasan yang memiliki jarak lebih dari ϵ. Misalkan $A = (x_a, y_a)$ dan $B = (x_b, y_b)$ adalah titik-titik yang akan diinterpolasi, maka dapat didefinisikan titik interpolasi $C = (x_c, y_c)$ sebagai titik tengah dari A dan B dengan

$$x_c = \frac{x_a + x_b}{2}, \qquad y_c = \frac{y_b - y_a}{x_b - x_a}(x_c - x_a) + y_a \qquad (4)$$

Interpolasi dilakukan secara rekursif, sampai semua titik yang berurutan dalam lintasan tersebut mempunyai jarak kurang dari ϵ. Langkah-langkah interpolasi dalam lintasan yang mempunyai m buah titik koordinat awal secara lebih mendetail dapat dilihat pada diagram alir dalam Gambar 2 berikut:



Gambar 2. Flowchart Interpolasi Lintasan

Pada interpolasi lintasan, terjadi pembaruan terhadap banyaknya titik-titik koordinat, yaitu titik-titik koordinat lama ditambah titik-titik koordinat interpolasi. Banyaknya titik interpolasi yang ditambahkan bisa lebih dari satu dan bergantung pada besarnya jarak antara kedua titik beserta nilai ϵ yang dipilih. Sebagai contoh, pandang sebuah lintasan dengan empat buah titik koordinat $s = \{s_1, s_2, s_3, s_4\}$ sebagaimana dalam Gambar 3 berikut:



Gambar 3. Titik-titik koordinat dalam lintasan s sebelum (kiri) dan sesudah ditambahkan titik-titik interpolasi (kanan)

Titik-titik koordinat lintasan dalam Gambar 3 (kiri) adalah $s_1 = (-7.9389,112.5801)$, $s_2 = (7.9389,112.5799)$, $s_3 = (-7.9388,112.5799)$, dan $s_4 = (-7.9377,112.5799)$. Dengan menggunakan persamaan (1) dapat dihitung jarak antara

titik-titik terdekat dalam lintasan tersebut, yaitu $d(s_1, s_2) = 30.812$ m, $d(s_2, s_3) = 12.1431$ m, dan $d(s_3, s_4) = 119.7142$ m. Untuk melakukan interpolasi, pilih $\epsilon \in \mathbb{R}_+$ sebagai jarak terjauh antara titik-titik koordinat berdekatan setelah lintasan di interpolasi, kemudian terapkan algoritma interpolasi seperti pada bagan alir dalam Gambar 2. Jika di ambil $\epsilon = 20$ akan diperoleh titik-titik koordinat baru dalam lintasan sedemikian hingga semua jarak antar titik-titik yang berdekatan kurang dari 20. Hasil interpolasi lintasan s dapat dilihat pada Tabel 1 berikut:

Tabel 1. Hasil interpolasi lintasan s

s	s'	x	y	Jarak antar titik (meter) $d(s'_{i-1}, s'_i),\ i = 2, \ldots, 12$
s_1	s'_1	-7.9389	112.5801	-
	s'_2	-7.9389	112.5800	15.4060
s_2	s'_3	-7.9389	112.5799	15.4060
s_3	s'_4	-7.9388	112.5799	12.1431
	s'_5	-7.9387	112.5799	14.9643
	s'_6	-7.9385	112.5799	14.9643
	s'_7	-7.9384	112.5799	14.9643
	s'_8	-7.9383	112.5799	14.9643
	s'_9	-7.9381	112.5799	14.9643
	s'_{10}	-7.9380	112.5799	14.9643
	s'_{11}	-7.9379	112.5799	14.9643
s_4	s'_{12}	-7.9377	112.5799	14.9643

Pada Tabel 1 kolom kedua diatas, $s'_2, s'_5, s'_6, s'_7, s'_8, s'_9, s'_{10}, s'_{11}$ adalah titik-titik baru yang ditambahkan pada lintasan s akibat dari interpolasi, dan pada kolom ke-4 dapat dilihat bahwa jarak antara setiap koordinat yang berurutan kurang dari $\epsilon = 20$. Banyaknya titik-titik koordinat pada lintasan s setelah diinterpolasi menjadi 12 titik, yaitu 4 buah titik awal $(s'_1 = s_1, s'_3 = s_2, s'_4 = s_3, s'_{12} = s_4)$ ditambah 8 buah titik hasil interpolasi. Perubahan titik-titik koordinat pada lintasan s dapat dilihat pada Tabel 1.

Pada Gambar 3 (kanan) lingkaran warna kuning adalah titik-titik koordinat asal dalam lintasan s, sedangkan tanda × warna hijau adalah lokasi titik-titik interpolasi yang ditambahkan pada lintasan. Antara titik s_2 dan s_3 tidak ditambahkan titik baru karena $d(s_2, s_3) \geq 20$.

2.3 Transformasi Lintasan

Interpolasi dapat dilakukan langsung pada sebuah lintasan tunggal, tetapi untuk mengubah lintasan menjadi sekuens huruf alfabet diperlukan lintasan lain sebagai pembanding. Salah satu bagian penting dari transformasi lintasan ke bentuk sekuens huruf alfabet adalah mengidentifikasi koordinat-koordinat yang ekivalen pada kedua lintasan, kemudian memberikan label dengan huruf yang sama jika koordinat ekivalen, dan dengan huruf yang berbeda jika koordinat tidak ekivalen.

Misalkan r dan s adalah dua buah lintasan yang akan diubah menjadi sekuens huruf alfabet, maka langkah pertama yang harus dilakukan adalah menghitung jarak antara titik-titik koordinat yang ada di r dengan titik-titik koordinat yang ada di s, kemudian mencari titik-titik koordinat di r yang jaraknya kurang dari ϵ terhadap titik-

titik koordinat di s, setelah itu lakukan pelabelan. Alur untuk mengubah lintasan menjadi sekuens huruf alfabet secara lebih mendetail dapat dilihat pada *function lintasantosequens* dalam Gambar 5 dan diagram alir dalam Gambar .

Sebagai contoh, diberikan dua buah lintasan r dan s dengan elemen berupa titik-titik koordinat geodesi sebagaimana pada Tabel 2 berikut:

Tabel 2. Titik-titik koordinat lintasan r (kiri) dan lintasan s (kanan)

r	x	y
r_1	-7.9389	112.5801
r_2	-7.9389	112.5799
r_3	-7.9382	112.5799
r_4	-7.9378	112.5799
r_5	-7.9376	112.5799

s	x	y
s_1	-7.9389	112.5801
s_2	-7.9389	112.5799
s_3	-7.9388	112.5799
s_4	-7.9377	112.5799

Pertama kita hitung semua jarak antara koordinat yang ada di r dan s menggunakan persamaan (1) dan diubah kedalam satuan meter, adapun hasil perhitungannya dapat dilihat pada Tabel 3 yaitu:

Tabel 3. Jarak antara titik-titik koordinat dalam lintasan r dan s

$d(r,s)$	s_1	s_2	s_3	s_4
r_1	**3.0182**	29.3034	30.6550	137.6043
r_2	28.4627	**3.1641**	**9.9130**	129.6270
r_3	84.4396	78.4808	66.5822	53.1365
r_4	122.6934	118.4714	106.5713	**13.1431**
r_5	153.5585	149.6993	137.8312	**18.1785**

Pada Tabel 3 diatas, jarak antara titik yang nilainya kurang dari $\epsilon = 20$ ditandai dengan huruf merah tebal. Dapat dikatakan bahwa, jarak yang kurang dari ϵ merupakan titik-titik yang ekivalen, yaitu titik-titik yang akan dilabeli menggunakan huruf yang sama. Proses pelabelannya adalah sebagai berikut:

Berdasarkan Tabel 3, lakukan Langkah-langkah dibawah ini:

a) Nilai $d(r_1, s_1) \leq 20$ maka $r_1 = X_1$ dan $s_1 = X_1$.

b) Nilai $d(r_2, s_2) \leq 20$ maka $r_2 = X_2$ dan $s_2 = X_2$.

c) Nilai $d(r_2, s_3) \leq 20$, karena $d(r_2, s_2) < d(r_2, s_3)$ maka $r_2 = X_2$ dan $s_3 = S_1$.

d) Karena nilai $d(r_3, s_j) > 20$, untuk $j = 1, \dots, n$, maka $r_3 = R_1$.

e) Nilai $d(r_4, s_4) \leq 20$ maka $r_4 = X_3$ dan $s_4 = X_3$.

f) Nilai $d(r_5, s_4) \leq 20$, karena $d(r_4, s_4) < d(r_5, s_4)$ maka $r_5 = R_2$ dan $s_4 = X_3$

Dari poin a) sampai f) diatas diperoleh perubahan sebagai berikut

Tabel 4. Hasil transformasi lintasan

r	R
r_1	X_1
r_2	X_2
r_3	R_1

s	S
s_1	X_1
s_2	X_2
s_3	S_1

r_4	X_3	s_4	X_3
r_5	R_2		

Transformasi lintasan r dan lintasan s diatas dilakukan tanpa melakukan interpolasi terlebih dahulu. Proses transformasi lintasan dengan atau tanpa interpolasi caranya sama, yang berbeda adalah banyaknya titik-titik koordinat yang harus di transformasi. Pada Tabel 4 diatas, label huruf yang sama pada kedua lintasan ada sebanyak 3, yaitu X_1, X_2 dan X_3 artinya ada tiga buah titik koordinat di r dan di s yang ekivalen satu sama lain. Sebagai ilustrasi dapat dilihat pada gambar dibawah ini



Gambar 4. Koordinat-koordinat yang ekivalen

Pada Gambar 4 tersebut, koordinat-koordinat yang ekivalen diberi tanda menggunakan lingkaran berwarna merah yang didalamnya memuat lingkaran hijau (bagian dari lintasan r) dan lingkaran kuning (bagian dari lingkaran s). Pada gambar tersebut terlihat bahwa $r_1 \equiv s_1$, $r_2 \equiv s_2$, dan $r_4 \equiv s_4$. Sebenarnya, $r_2 \equiv s_3$ dan $r_5 \equiv s_4$ juag, karena $d(r_2, s_2) \leq d(r_2, s_3)$ dan $d(r_4, s_4) \leq d(r_5, s_4)$ maka s_3 dan r_5 kita lewati dan dianggap tidak ekivalen dengan titik lain seperti titik r_3.

```
function [a, b] = lintasantosekuen (r, s)
24   E = jarak (r, s)
25   a = buatlabelR (E)
26   b = buatlabelS (E)
```

```
function E = jarak (r, s)
1    for i = 1 : m
2        for j = 1 : n
3            d = ||r_i − s_j||_2
4            if d ≤ ϵ
5                E_{i,j} = 1
6            else
7                E_{i,j} = 0
```

Gambar 5. Algoritma Transformasi Lintasan

```
function r = buatlabelR(E)
8   k = 0, l = 0, r' = ""
9   for i = 1 : m
10      if sumrow(E, i) = 0
11          l = l + 1
12          r' = r' + "R_l"
13      else
14          n = caribaris(E_i, 1)
15          for j = 1 : n
16              k = k + 1
17              r' = r' + "X_k"
```

```
function s = buatlabelS(E)
24  k = 0, l = 0, s' = []
25  for j = 1 : n
26      if sumcol(E, j) = 0
27          l = l + 1
28          s' = s' + "S_l"
29      else
30          m = carikolom(E_j, 1)
31          for i = 1 : m
32              k = k + 1
33              s' = s' + "X_k"
```

Gambar 7. Algoritma Transformasi Lintasan ke Bentuk Sekuens Huruf Alfabet

2.4 *Penyejajaran Sekuens*

Setelah lintasan diubah menjadi sekuens huruf alfabet, langkah berikutnya adalah menyejajarkan sekuens sedemikian hingga karakter-karakter yang sama dapat berada pada urutan kolom yang sama. Metode penyejajaran yang digunakan dalam tulisan ini adalah penyejajaran global Needleman Wunsch, sebagai pengingat dapat dilihat pada algoritma *scoring* dan *backtrace* berikut:

```
function p = scoring(a, b, g, match, mismatch)
p_{1,1} = 0
for i = 1 : m
    p_{i,1} = -ig
for j = 1 : n
    p_{1,j} = -ig
for i = 1 : m
    for j = 1 : n
        if a_{i-1} = b_{j-1}
            s(a_i, b_i) = match
        else
            s(a_i, b_i) = mismatch
        p_{i,j} = max{p_{i-1,j-1} + s(a_i, b_j), p_{i-1,j} + g, p_{i,j-1} + g}
```

```
function [a', b'] = backtrace(p, a, b)
atur i = m dan j = n
atur a' = "" dan b' = ""
while (i > 1) or (j > 1)
    if a_{i-1} = b_{j-1}
        s(a_i, b_i) = match
    else
        s(a_i, b_i) = mismatch
    if p_{i,j} = p_{i-1,j-1} + s(a_i, b_i)
        i = i - 1, j = j - 1, a'_i = a_{i-1}, b'_j = b_{j-1}
    elseif p_{i,j} = p_{i-1,j} + gap
        i = i - 1, a'_i = a_{i-1}, b'_j = '-'
    else
        j = j - 1, a'_i = '-', b'_j = b_{j-1}
a' = flip(a'), b' = flip(b')
```

Gambar 8. Algoritma Needleman Wunsch

Pada bagian 2.3 telah diperoleh sekuens hasil transformasi $R = X_1X_2R_1X_3R_2$ dan $S = X_1X_2S_1X_3$ setelah disejajarkan menggunakan algoritma Needleman Wunsch dengan mengambil nilai skor $(match > mismatch > gap)$ diperoleh $R' = X_1X_2R_1X_3R_2$ dan $S' = X_1X_2S_1X_3 -$.

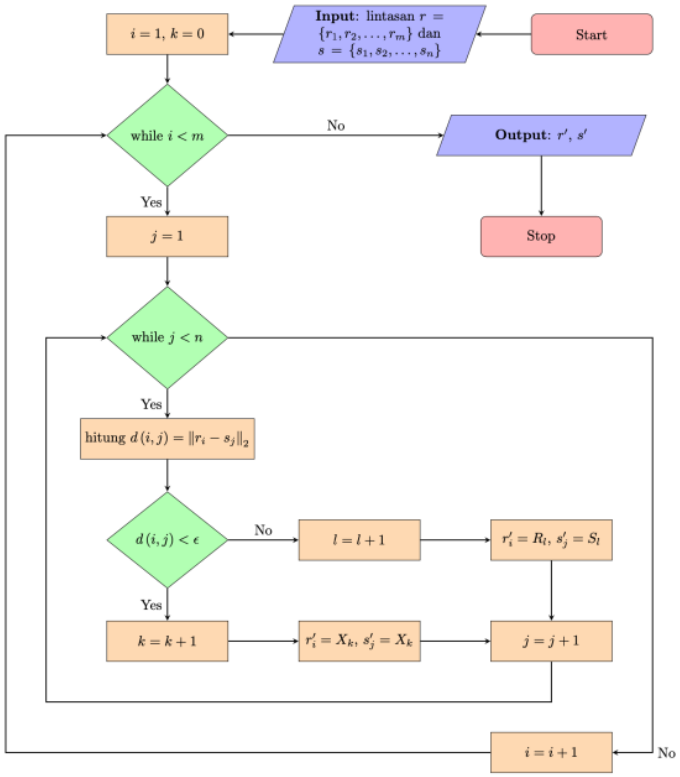
R	=	X_1	X_2	R_1	X_3	R_2		R'	=	X_1	X_2	R_1	X_3	R_2
S	=	X_1	X_2	S_1	X_3		$\Rightarrow$	S'	=	X_1	X_2	S_1	X_3	$-$

Gambar 9. Hasil penyejajaran sekuens dengan algoritma Needleman Wunsch menggunakan skor (match>mismatch>gap)

Dari hasil penyejajaran menggunakan skoring diatas diperoleh 3 lokasi karakter *match*, 1 lokasi karakter *mismatch*, dan 1 lokasi karakter gap. Jika digunakan nilai skor $(match > gap > mismatch)$ diperoleh 3 lokasi memuat karakter *match* dan 3 lokasi memuat karakter *gap,* lihat Gambar 7 berikut:

R	=	X_1	X_2	R_1	X_3	R_2		R'	=	X_1	X_2	R_1	$-$	X_3	R_2
S	=	X_1	X_2	S_1	X_3		$\Rightarrow$	S'	=	X_1	X_2	$-$	S_1	X_3	$-$

Gambar 10. Hasil penyejajaran sekuens dengan algoritma Needleman Wunsch menggunakan penilaian (match>gap>mismatch)



Gambar 11. Bagan alir transformasi lintasan ke bentuk sekuens huruf abjad

2.5 Kesamaan Lintasan

Kesamaan lintasan sebagaimana yang didefinisikan oleh Cavojsky (Čavojský et al., 2020) didasarkan pada banyaknya posisi koordinat yang identik dan banyaknya gap dari dua sekuens yang telah disejajarkan, yaitu:

Definisi 1 (kesamaan). *Lintasan r dan s yang terdiri dari m dan n buah titik koordinat, dianggap sama jika*

$$\frac{match}{\max(m,n)} \geq \alpha \quad dan \quad maxgap \leq \beta \tag{5}$$

dengan $\#match$ adalah banyaknya karakter yang bersesuaian dan $\#maxgap$ adalah jumlah gap terbanya dalam sub himpunan gap.

Menggunakan Definisi 1 diatas, dapat kita definisikan persentase kesamaan dari dua buah lintasan sebagai

$$\text{kesamaan } (\%) = \frac{match}{\max(m,n)} \times 100\% \tag{6}$$

Pada sekuens DNA *match* didefinisikan sebagai basa-basa yang sama dan berada pada posisi yang sama dari dua buah untaian DNA, pada lintasan $match$ adalah koordinat-koordinat yang ekivalen pada kedua lintasan yang diwakili oleh karakter-karakter yang berada pada posisi yang sama dalam kedua sekuens. Parameter m, n pada persamaan (5) secara berturut-turut adalah panjang sekuens R dan sekuens S. Jadi pada Gambar 6 diatas diperoleh $match = 3$, $m = 5$ dan $n = 4$, jadi similaritas dari lintasan R dan S diatas adalah

$$\text{kesamaan } (\%) = \frac{3}{\max(5,4)} \times 100\% = \frac{3}{5} \times 100\% = 60\% \quad (7)$$

3 DISKUSI DAN HASIL

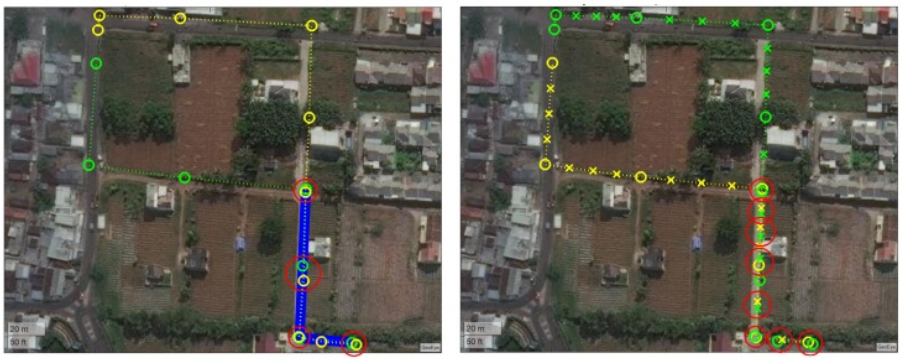
Banyaknya titik interpolasi yang ditambahkan pada sebuah lintasan bergantung pada seberapa besar jarak maksimum (ϵ) yang dinginkan setelah lintasan diinterpolasi. Semakin kecil nilai ϵ yang diberikan, akan semakin banyak titik interpolasi yang dibangkitkan, demikian juga sebaliknya. Interpolasi lintasan berfungsi untuk meningkatkan akurasi kesamaan antar lintasan yang berada pada jalur yang sama, dengan mengestimasi titik-titik menggunakan garis yang dibentuk oleh titik-titik pada lintasan. Interpolasi dapat menegaskan bagian-bagian mana dari suatu lintasan yang memiliki arah dan jalur yang sama.

3.1 Percobaan 1

Perhatikan lintasan A dan B pada Gambar 12, panjang lintasan A (warna hijau) adalah 277.98 meter dan panjang lintasan B (warna kuning) adalah 317.90 meter. Besarnya nilai kesamaan atau similaritas antara lintasan A dan B adalah panjang jalur yang dilalui bersama oleh kedua objek dari masing-masing lintasan dibandingkan dengan lintasan terpanjang. Lintasan A dan B berada pada jalur yang sama sejauh 107.13 meter, jadi nilai similaritasnya adalah

$$\begin{aligned} \text{kesamaan } (\%) &= \frac{\text{panjang irisan lintasan}}{\max(\text{panjang lintasan } A, \text{panjang lintasan } B)} \times 100\ \% \\ &= \frac{107.13}{\max(277.98,317.90)} \times 100\ \% = 33.7\% \end{aligned} \quad (8)$$

Panjang irisan jalur yang dilalui oleh kedua objek yaitu garis tebal berwarna biru, secara visual mudah untuk dikenali dan ditemukan oleh mata manusia, tetapi sulit bagi komputer. Oleh sebab itu diperlukan metode-metode yang mudah dikerjakan oleh program atau komputer sebagaimana yang sudah di jelaskan pada bagian 2.



Gambar 12. Lintasan A dan B Sebelum diinterpolasi (kiri) dan setelah diinterpolasi (kanan)

Pada Gambar 12 (kiri) diatas, koordinat-koordinat yang ekivalen dari kedua lintasan yaitu yang jaraknya (ϵ) kurang dari 15 meter diberi tanda dengan lingkaran berwarna merah, jadi ada 4 buah titik koordinat yang ekivalen. Banyaknya titik pada lintasan A ada 7 buah sedangkan untuk B ada 10 buah, berdasarkan definisi kesamaan pada bab 2.5 nilai kesamaan dari kedua lintasan tersebut adalah

$$\begin{aligned} \text{kesamaan } (\%) &= \frac{\text{banyak titik ekivalen}}{\max(\text{jml titik lintasan } A, \text{jml titik lintasan } B)} \times 100\% \\ &= \frac{4}{\max(7,10)} \times 100\% = \frac{4}{10} \times 100\% = 40\% \end{aligned} \tag{9}$$

Pada Gambar 12 (kanan) lintasan diinterpolasi menggunakan $\epsilon = 20$, sehingga banyaknya titik pada lintasan A menjadi 21 dan banyaknya titik pada lintasan B menjadi 24. Tititk-titik koordinat yang ekivalen menjadi 8 titik. Persentase kesamaaan lintasan A dan B pada Gambar 12 (kanan) menjadi

$$\begin{aligned} \text{kesamaan } (\%) &= \frac{8}{\max(21,24)} \times 100\% \\ &= \frac{8}{24} \times 100\% = 33.33\% \end{aligned} \tag{10}$$

Jika kesamaan yang diperoleh pada persamaan (9) dan (10) dibandingkan dengan kesamaan pada persamaan (8), secara berturut-turut selisihnya adalah 6.3% dan 0.37%. Sedangkan akurasi kesamaan sebelum interpolasi adalah 81%, setelah diinterpolasi akurasinya meningkat menjadi 99%.

3.2 Percobaan 2

Perhatikan lintasan C dan D pada Gambar 13 berikut, panjang lintasan C (warna hijau) adalah 3897.79 meter dan panjang lintasan C (warna kuning) adalah 4370.64 meter. Besarnya nilai kesamaan atau similaritas antara lintasan C dan D adalah panjang jalur yang dilalui bersama oleh kedua objek dari masing-masing lintasan dibandingkan dengan lintasan terpanjang. Lintasan C dan D berada pada jalur yang sama sejauh 2989.65 meter, jadi nilai similaritasnya adalah

$$\text{kesamaan } (\%) = \frac{\text{panjang irisan lintasan}}{\max(\text{panjang lintasan } A, \text{panjang lintasan } B)} \times 100\ \% = \frac{2989.65}{\max(3897.79, 4370.64)} \times 100\ \% = 68.4\% \quad (11)$$

Panjang irisan jalur yang dilalui oleh kedua objek yaitu garis tebal berwarna biru, secara visual mudah untuk dikenali dan ditemukan oleh mata manusia, tetapi sulit bagi komputer. Oleh sebab itu diperlukan metode-metode yang mudah dikerjakan oleh program atau komputer sebagaimana yang sudah di jelaskan pada bagian 2.

Pada Gambar 13 berikut, koordinat-koordinat yang ekivalen dari kedua lintasan yaitu yang jaraknya (ϵ) kurang dari 15 meter diberi tanda dengan lingkaran berwarna merah, jadi ada 10 buah titik koordinat yang ekivalen. Sedangkan titik-titik koordinat pada lintasan C dan D dalam Gambar 13 ada sebanyak 19 dan 42 buah. Menggunakan definisi kesamaan pada bab 2.5 nilai kesamaan dari kedua lintasan tersebut adalah

$$\text{kesamaan } (\%) = \frac{\text{banyak titik ekivalen}}{\max(\text{jml titik lintasan } A, \text{jml titik lintasan } B)} \times 100\% = \frac{10}{\max(19,42)} \times 100\% = 23.81\% \quad (12)$$



Gambar 13. Lintasan C (hijau) lintasan D (kuning), Irisan lintasan C dan D (garis tebal warna biru), titik-titik koordinat yang ekivalen (warna merah)



Gambar 14. Titik-titik koordinat lintasan C (warna hijau) dan lintasan D (warna kuning) setelah di interpolasi, titik-titik koordinat yang ekivalen (warna merah)

Pada Gambar 14, lintasan diinterpolasi menggunakan $\epsilon = 20$ sehingga titik-titik koordinat pada lintasan C menjadi 267 dan banyaknya titik pada lintasan D menjadi 312. Titik-titik koordinat yang ekivalen ada sebanyak 186 titik. Jadi persentase kesamaaan lintasan C dan D pada Gambar 14 menjadi

$$\text{kesamaan } (\%) = \frac{186}{\max(267,312)} \times 100\% = 59.62\% \qquad (13)$$

Jika kesamaan yang diperoleh pada persamaan (12) dan (13) dibandingkan dengan kesamaan pada persamaan (11), secara berturut-turut selisihnya adalah 44.59% dan 8.78%. Akurasi kesamaan sebelum interpolasi adalah 35%, setelah diinterpolasi akurasinya meningkat menjadi 87%.

4 KESIMPULAN

Dari pembahasan dapat disimpulkan bahwa dengan melakukan interpolasi, akurasi nilai kesamaan dari kedua lintasan dapat meningkat secara signifikan. Nilai kesamaan lintasan dapat dihitung tanpa melakukan penyejajaran terlebih dahulu, yaitu dengan cara menghitung banyaknya karakter atau label yang sama setelah sekuens di transformasi, kemudian dibandingkan dengan panjang sekuen terpanjang dari hasil transformasi lintasan. Penyejajaran diperlukan, jika kita ingin menggunakan definisi kesamaan utuk menyatakan dua buah lintasan sama atau berbeda.

5 DAFTAR PUSTAKA

Anisya, A., & Swara, G. Y. (2017). Implementation of Haversine Formula and Best First Search

Method in Searching of Tsunami Evacuation Route. *IOP Conference Series: Earth and Environmental Science*, *97*(1). https://doi.org/10.1088/1755-1315/97/1/012004

Beretta, S. (2018). Algorithms for strings and sequences: Pairwise alignment. In *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics* (Vols. 1–3). Elsevier Ltd. https://doi.org/10.1016/B978-0-12-809633-8.20317-8

Čavojský, M., & Drozda, M. (2019). Comparison of user trajectories with the Needleman-Wunsch algorithm. *International Conference on Mobile Computing, Applications, and Services*, 141–154.

Čavojský, M., Drozda, M., & Balogh, Z. (2020). Analysis and experimental evaluation of the Needleman-Wunsch algorithm for trajectory comparison. *Expert Systems with Applications*, *165*(May 2020). https://doi.org/10.1016/j.eswa.2020.114068

Chapra, S. C., & Canale, R. P. (1998). *Numerical methods for engineers*.

Chen, L., Özsu, M. T., & Oria, V. (2005). Robust and fast similarity search for moving object trajectories. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 491–502. https://doi.org/10.1145/1066157.1066213

Chua, S. L., & Foo, L. K. (2015). Sensor Selection in Smart Homes. *Procedia Computer Science*, *69*, 116–124. https://doi.org/10.1016/j.procs.2015.10.012

Garhwal, A. S., & Yan, W. Q. (2019). BIIIA: a bioinformatics-inspired image identification approach. *Multimedia Tools and Applications*, *78*(8), 9537–9552. https://doi.org/10.1007/s11042-018-6551-y

Ju, S., Park, S., Lim, H., Yun, S. B., & Heo, J. (2018). Spatial-data-driven student characterization: Trajectory sequence alignment based on student smart card transactions. *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Prediction of Human Mobility, PredictGIS 2018*, 1–7. https://doi.org/10.1145/3283590.3283591

Magdy, N., Abdelkader, T., & El-Bahnasy, K. (2018). A comparative study of similarity evaluation methods among trajectories of moving objects. *Egyptian Informatics Journal*, *19*(3), 165–177. https://doi.org/10.1016/j.eij.2018.03.001

Needleman, S. B., & Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, *48*(3), 443–453. https://doi.org/10.1016/0022-2836(70)90057-4

Sabarish, B. A., Karthi, R., & Kumar, T. G. (2020). Graph Similarity-based Hierarchical Clustering of Trajectory Data. *Procedia Computer Science*, *171*(2019), 32–41. https://doi.org/10.1016/j.procs.2020.04.004

Toohey, K., & Duckham, M. (2015). Trajectory similarity measures. *SIGSPATIAL Special*, *7*(1), 43–50. https://doi.org/10.1145/2782759.2782767

Wang, H., Su, H., Zheng, K., Sadiq, S., & Zhou, X. (2013). An effectiveness study on trajectory similarity measures. *Conferences in Research and Practice in Information Technology Series*, *137*(February), 13–22.

Trajectories - 6

ORIGINALITY REPORT

15%	12%	12%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Maroš Čavojský, Martin Drozda, Zoltán Balogh. "Analysis and experimental evaluation of the Needleman-Wunsch algorithm for trajectory comparison", Expert Systems with Applications, 2020 Publication	2%
2	Submitted to Universitas Pendidikan Indonesia Student Paper	1%
3	bdigital.unal.edu.co Internet Source	1%
4	"Trends in Artificial Intelligence Theory and Applications. Artificial Intelligence Practices", Springer Science and Business Media LLC, 2020 Publication	1%
5	journal.hep.com.cn Internet Source	1%
6	Submitted to Universiti Teknikal Malaysia Melaka Student Paper	1%

7	Submitted to Universidad de Medellin Student Paper	1%
8	text-id.123dok.com Internet Source	<1%
9	uri.gbv.de Internet Source	<1%
10	link.springer.com Internet Source	<1%
11	Georg Osang, James Cook, Alex Fabrikant, Marco Gruteser. "LiveTraVeL: Real-time matching of transit vehicle trajectories to transit routes at scale", 2019 IEEE Intelligent Transportation Systems Conference (ITSC), 2019 Publication	<1%
12	Submitted to Universidad San Francisco de Quito Student Paper	<1%
13	vibdoc.com Internet Source	<1%
14	B.A. Sabarish, R. Karthi, T. Gireesh Kumar. "Graph Similarity-based Hierarchical Clustering of Trajectory Data", Procedia Computer Science, 2020 Publication	<1%

15	ejournal.uin-malang.ac.id Internet Source	<1%
16	Bolong Zheng, Han Su, Kai Zheng, Xiaofang Zhou. "Landmark-Based Route Recommendation with Crowd Intelligence", Data Science and Engineering, 2016 Publication	<1%
17	documentop.com Internet Source	<1%
18	www.researchsquare.com Internet Source	<1%
19	Suhendra Suhendra, Feby Nopriandy. "Rancang Bangun dan Uji Kinerja Tugal Semi-Mekanis dengan Sistem Penjatah Berputar untuk Kacang Hijau (Vigna radiata L.)", POSITRON, 2018 Publication	<1%
20	www.tandfonline.com Internet Source	<1%
21	erkanulker.bayebilisim.com Internet Source	<1%
22	Abhimanyu Singh Garhwal, Wei Qi Yan. "BIIIA: a bioinformatics-inspired image identification approach", Multimedia Tools and Applications, 2018 Publication	<1%

23	dergipark.org.tr Internet Source	<1%
24	devound.wordpress.com Internet Source	<1%
25	e-belajaronline2.blogspot.com Internet Source	<1%
26	koreascience.or.kr Internet Source	<1%
27	repository.its.ac.id Internet Source	<1%
28	repository.unair.ac.id Internet Source	<1%
29	www.ship-research.com Internet Source	<1%
30	123dok.com Internet Source	<1%
31	Douglas do Couto Teixeira, Aline Carneiro Viana, Mário S. Alvim, Jussara M. Almeida. "Deciphering Predictability Limits in Human Mobility", Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2019 Publication	<1%

32	docplayer.info Internet Source	<1%
33	pt.scribd.com Internet Source	<1%
34	www.scribd.com Internet Source	<1%
35	Tetsushi Yada. "Genome Alignment", Elsevier BV, 2019 Publication	<1%
36	"Computational Vision and Bio Inspired Computing", Springer Science and Business Media LLC, 2018 Publication	<1%
37	idoc.pub Internet Source	<1%
38	repository.usd.ac.id Internet Source	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		