BAB II

LANDASAN TEORI

Pada bagian ini akan diuraikan teori-teori dasar yang dijadikan sebagai landasan dalam penulisan tugas akhir ini.

2.1 Ilmu Bioinformatika

Bioinformatika merupakan kajian yang mengkombinasikan disiplin ilmu biologi molekular, matematika, statistika, dan teknik komputasi [1]. Ilmu tersebut digunakan untuk mengolah dan menganalisis data-data biologi molekular. Biologi molekular adalah ilmu yang mempelajari organisme pada tingkat molekular.

Pada awalnya ilmu bioinformatika muncul atas inisiatif para ahli biologi molekular dan ahli statistik. Menurut pendapat para ahli tersebut, semua gejala yang ada di alam dapat dibuat secara *artificial* melalui simulasi dari data-data yang ada [5]. Pada saat ini, bioinformatika mempunyai peranan yang sangat penting, diantaranya adalah untuk manajemen data-data biologi molekular, terutama sekuens DNA dan informasi genetika lainnya. Perangkat utama bioinformatika adalah *software* yang diperlukan untuk pengolahan, penyimpanan dan simulasi serta visualisasi karakteristik data biologi molekular.

Perkembangan ilmu bioinformatika sejak kemunculannya pada pertengahan tahun 1980-an hingga sekarang semakin pesat. Hal ini ditandai dengan kemampuan manusia memahami genom, yaitu cetak biru informasi genetik yang menentukan sifat setiap makhluk hidup yang tersirat dalam rantai *deoxyribonucleic acid* (DNA) [5]. Pada penerapannya data-data biologi molekular yang menjadi obyek penelitian adalah berupa sekuen DNA, RNA, dan protein. Agar dapat dianalisis dengan metode komputasi yang ada, sekuen tersebut direpresentasikan oleh sejumlah terhingga sekuen berupa huruf-huruf alfabet. Sekuen DNA direpresentasikan oleh 4 (empat) huruf yaitu A, C, G, dan T. Setiap hurufnya adalah inisial dari asam nukleat atau nukleotida (*nucleotides*) penyusun DNA, yaitu *adenine* (A)*, cytosine* (C)*, guanine* (G), dan *thymine* (T) [1].

Molekul DNA umumnya terdapat pada kromosom yang terletak pada nukleus atau inti sel dari sebuah organisme hidup. Sebuah molekul DNA terdiri dari dua buah rantai nukleotida yang berpilin (*double helix*). Pada susunan *double helix* tersebut, setiap nukleotida pada rantai yang satu berpasangan dengan nukleotida pada rantai pasangannya. Nukleotida *adenin* (A) berikatan dengan *thymine* (T) dan *cytosine* (C) berikatan dengan *guanine* (G) [1].

Berbeda dengan DNA, sekuen protein memiliki lebih banyak unsur penyusunnya yaitu terdiri dari 20 asam amino yang berbeda. Setiap asam aminonya direpresentasikan oleh sebuah huruf sehingga terdapat 20 huruf yang berbeda pada satu sekuen protein, yaitu A, C, D, E, F, G, H, I, K, L, M,

N, P, Q, R, S, T, V, W, dan Y. Sedangkan *ribonucleic acid* (RNA) hampir mirip dengan DNA, tersusun dari 4 nukleotida yang direpresentasikan oleh huruf A, C, G, dan U, yang membedakannya dengan DNA adalah RNA hanya terdiri dari satu rantai nukleotida. RNA berfungsi untuk membantu proses pembuatan protein [1]. Sekuen-sekuen DNA, RNA, dan protein merupakan obyek yang akan dianalisis dengan menggunakan berbagai metode untuk mencapai tujuan tertentu. Banyak hal yang bisa diperoleh dengan menganalisis sekuens di atas, karena informasi genetik yang berada pada sekuen biologi molekular tersebut sangatlah banyak dengan fungsi dan sifat yang berbeda-beda. Pada tugas akhir ini secara tidak langsung sekuen di atas akan digunakan sebagai obyek penelitian untuk membentuk suatu pohon kekerabatan dengan tujuan untuk melihat kekerabatan dari beberapa spesies .

2.2 *Tree*

Tree merupakan bagian terpenting dalam penulisan tugas akhir ini. Tetapi sebelum membahas definisi *tree*, ada beberapa hal yang terkait dengan definisi *tree* tersebut. Berikut adalah beberapa definisi yang perlu diketahui sebelum membahas mengenai *tree*.

Definisi 1.1. Graf sederhana G = (V,E) terdiri dari V, suatu himpunan tidak kosong yang unsur-unsurnya disebut *node*, dan E, suatu himpunan dari *unordered pairs* dari unsur-unsur terpisah V, disebut *edges* [6].

Definisi 1.2. *Node* adalah titik terminal atau titik perpotongan dari sebuah graf [6].

Definisi 1.3. *Edge* adalah garis yang memenghubungkan *node* [6].

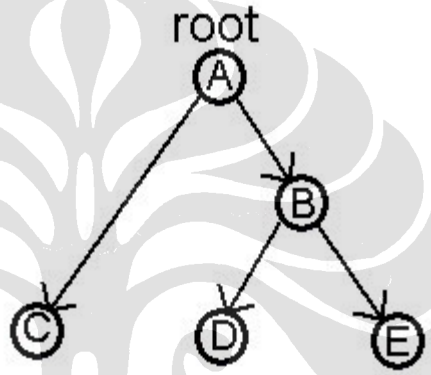
Definisi 1.4. Jalur (*path*) adalah barisan *node* sedemikian sehingga untuk setiap *node*, terdapat *edge* yang terhubung dengan *node* berikutnya di dalam barisan [6].

Definisi 1.5. *Cycle* adalah jalur yang titik awal dan titik akhirnya sama, dengan syarat *edge* tidak boleh dilalui lebih dari satu kali [6].

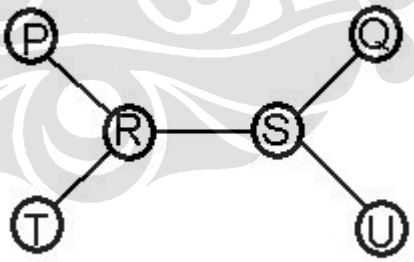
Sekarang akan dibahas mengenai *tree*, yaitu sebuah graf yang setiap dua *node*-nya dihubungkan oleh tepat satu *path* [7]. Setiap *tree* yang merupakan graf sederhana (G) yang pasti memenuhi setiap kondisi berikut.

- G terhubung dan tidak memiliki *cycle*
- G tidak memiliki *cycle*, tetapi jika ditambahkan sebuah *edge* maka G akan memiliki *cycle*.
- G terhubung dan menjadi tak terhubung jika sebuah *edge* dihilangkan dari G.
- Setiap dua *node* pada G dapat dihubungkan secara unik oleh sebuah *path*.

Secara umum *tree* dibagi menjadi dua jenis yaitu *unrooted tree* (tidak memiliki root) dan *rooted tree* (memiliki root). *Rooted tree* adalah *tree* berarah yang memiliki *root*, suatu *root* bisa dikatakan sebagai nenek moyang (*ancestor*) dari semua *node* pada *tree* tersebut. Sedangkan *unrooted tree* adalah *tree* yang tidak berarah yang tidak memiliki *node* yang bertindak sebagai nenek moyang untuk semua *node* [2].



Gambar 1. *Rooted tree*



Gambar 2. *Unrooted tree*

Sebagai contoh, pada Gambar 1 dan 2, *node* A bertindak sebagai *root* yang memiliki anak (*child node*) yaitu *node* B dan C. Node C, D, E, P, Q, T,

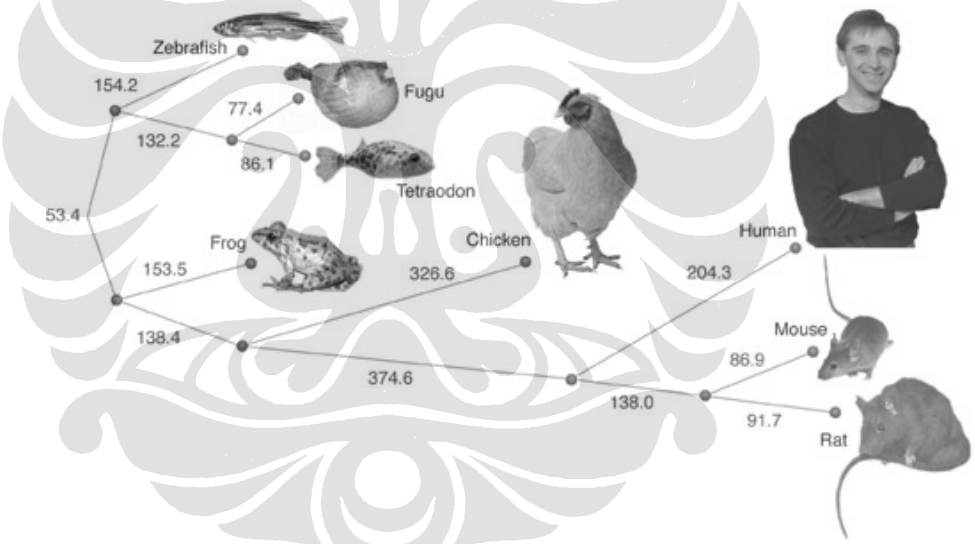
dan U bertindak sebagai terminal *node* atau *leaf node*, yaitu semua *node* yang hanya terhubung oleh satu *edge* dan tidak memiliki *child node*. Sedangkan *node* B, R, dan S bertindak sebagai *internal node*, yaitu semua *node* yang memiliki *child node* tetapi bukan *leaf node* [2].

2.3 *Phylogenetic Tree*

Phylogenetic adalah ilmu yang mempelajari hubungan evolusi organisme yang hidup di muka bumi ini. Hubungan evolusi organisme tersebut diperoleh dengan menganalisis struktur genetik dan melihat similiaritas yang ada pada organisme tersebut . Hubungan evolusi yang terjadi antar organisme direpresentasikan oleh sebuah *tree* yang disebut sebagai *phylogenetic tree.* Pada dasarnya pembentukan *tree* tersebut dilakukan dengan melihat similiaritas yang ada pada genetik organisme yang dibandingkan. Semakin tinggi tingkat similiaritas genetiknya, maka semakin dekat pula hubungan evolusinya dan kekerabatannya. Organisme-organisme yang akan dibentuk *tree* nya, haruslah organisme yang memiliki kemiripan (similaritas) secara genetik [1].

Pada mulanya para ahli biologi membandingkan sifat-sifat fisik yang ada pada suatu organisme, seperti morfologi, ukuran, warna, banyaknya kaki dan sebagainya untuk membangun suatu *phylogenetic tree*. Namun saat ini, ilmuwan sudah dapat menggunakan informasi genetik yang ada pada organisme, seperti DNA dan protein untuk membentuk suatu *phylogenetic tree* [5].

Pada *phylogenetic tree*, setiap terminal node atau *leaf* merepresentasikan spesies atau gen yang dibandingkan, atau disebut sebagai *operational taxonomic unit* (OTU). *Internal node* merepresentasikan nenek moyang bersama dari setiap *node* yang terhubung dengannya, selanjutnya disebut sebagai *hypothetical taxonomic units* (HTU) yang tidak dapat diteliti secara langsung, dan *edge* merepresentasikan hubungan genetik yang menentukan kekerabatannya. Gambar berikut adalah contoh sebuah *phylogenetic tree* dari beberapa spesies.



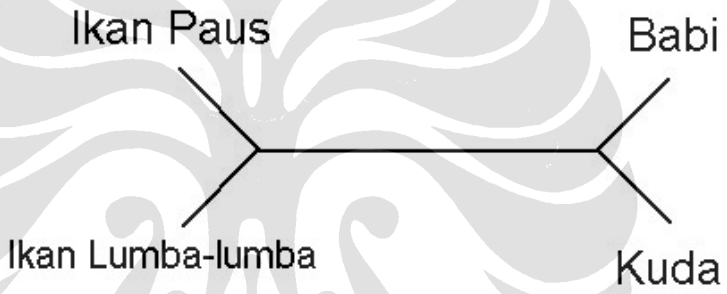
Sumber: http://www.idi.ntnu.no/~helgel/teaching/proposals/phylogenetic.html

Gambar 3. Contoh *Phylogenetic tree*

Phylogenetic tree dapat dikelompokan menjadi beberapa jenis, yaitu *rooted* atau *unrooted*, *binary tree* atau tidak, dan memiliki panjang *edge* atau tidak. *Rooted* atau *unrooted* ditentukan oleh adanya *node* yang menjadi *root*

atau tidak. *Binary tree* adalah *tree* yang setiap *node*-nya memiliki satu sampai tiga *edge* yang terhubung dengannya. Sebuah *tree* dapat memiliki panjang *edge* yang panjangnya merepresentasikan jarak antara *node* yang berhubungan [2].

Pada tugas akhir ini, *phylogenetic tree* yang dibangun merupakan *unrooted tree* dari sebuah *tree* tak berarah yang hanya dibatasi untuk *binary tree* dan memiliki panjang *edge*. Berikut ini adalah contoh *unrooted tree* dari empat spesies mamalia yaitu {Ikan Paus, Babi, Kuda, Ikan Lumba-lumba}.



Gambar 4. *Phylogenetic tree* sederhana

Gambar 4 merupakan contoh *unrooted tree* yang termasuk *binary tree* dan tidak memiliki panjang *edge*. Pada gambar *phylogenetic tree* tersebut, bisa ditarik kesimpulan bahwa spesies Ikan Paus lebih dekat kekerabatannya dengan Ikan Lumba-lumba dari pada dengan Kuda atau Babi karena Ikan Paus dan Ikan Lumba-lumba terhubung oleh *internal node* yang sama yang menjadi nenek moyang bersama (*common ancestor*) [1].

Terdapat dua metode dalam pembentukan *phylogenetic tree*, yaitu metode berdasarkan karakter (*character based methods*) dan metode

berdasarkan jarak (*distance based methods*). Pada penulisan tugas akhir ini, metode yang dipakai untuk membentuk *phylogenetic tree* adalah *distance methods* yang akan dijelaskan pada subbab berikutnya.

2.4 Metode Jarak *(Distance Methods)*

Metode jarak atau *distance methods* adalah salah satu metode pembentukan *tree* dari sekumpulan jarak antar setiap pasang sekuen (*sequence*) yang telah disejajarkan (*alignment*) [1]. Penulisan tugas akhir ini hanya difokuskan pada masalah pembentukan *phylogenetic tree* dari sebuah matriks jarak yang telah diberikan. Penulis tidak akan membahas masalah proses pembentukan matriks jarak dan proses penyejajaran (*alignment*). Namun, penulis memberikan gambaran umum mengenai alur pembentukan matriks jarak hingga terbentuknya *phylogenetic tree* pada Diagram 1. Definisi matriks jarak lebih lanjut akan dijelaskan kemudian.

Diagram tersebut menunjukan alur pembentukan suatu matriks jarak dari sehimpunan sekuen yang selanjutnya digunakan untuk membentuk *phylogenetic tree*. Pada diagram tersebut terdapat beberapa tahapan yang harus dilakukan hingga terbentuknya sebuah *phylogenetic* tree. Setiap tahapan memiliki proses dan algoritma yang bervariasi untuk mencapai tahap berikutnya.

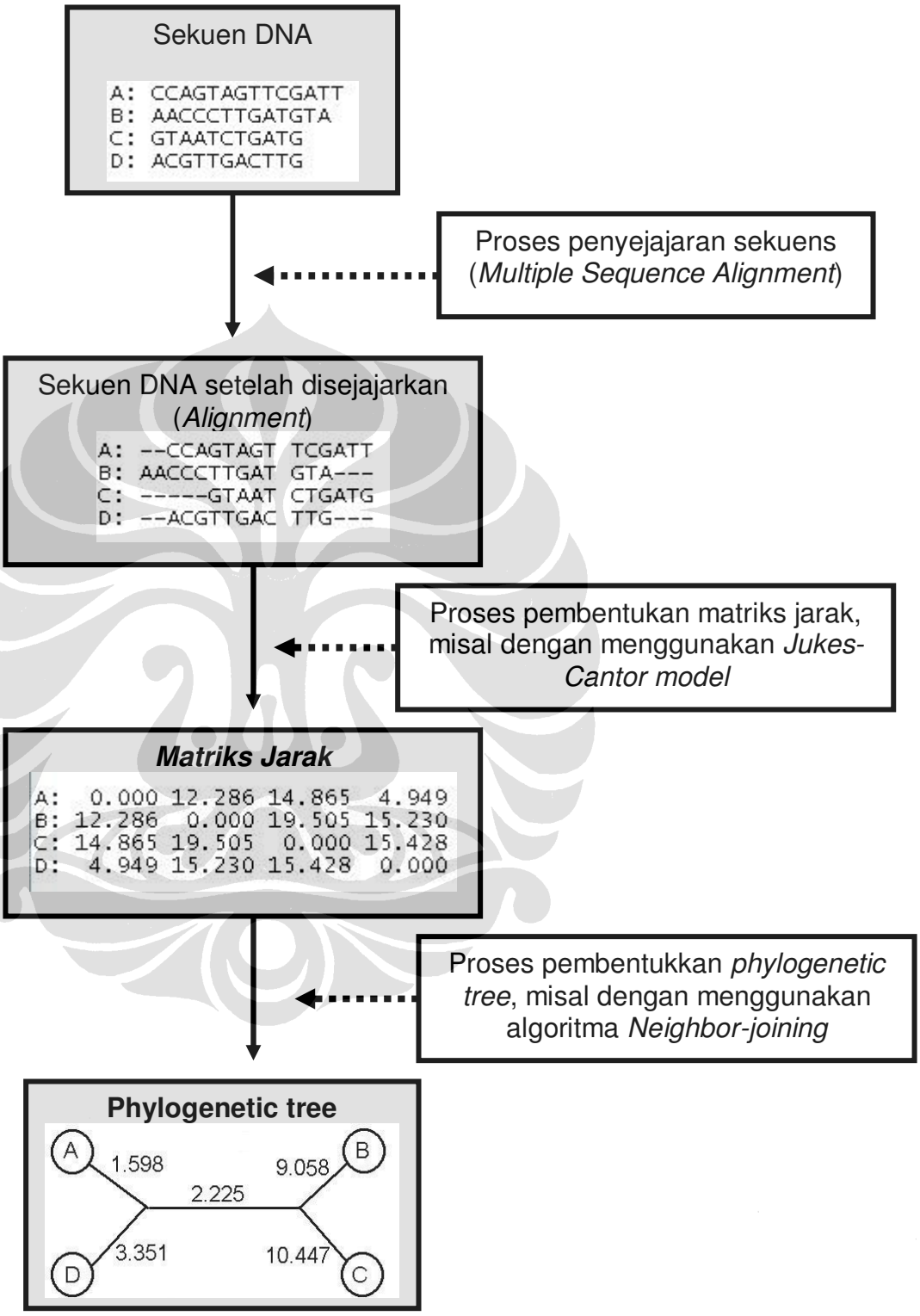


Diagram 1. Alur pembentukan matriks jarak hingga *phylogenetic tree*

Sekumpulan jarak dituliskan dalam bentuk matriks, yang disebut matriks jarak [1]. Perhatikan contoh matrik jarak berikut.

Md	x^1	x^2	x^3	x^4	x^5	x^6
x^1	0	8	3	14	10	12
x^2	8	0	9	10	6	8
x^3	3	9	0	15	11	13
x^4	14	10	15	0	10	8
x^5	10	6	11	10	0	8
x^6	12	8	13	8	8	0

Gambar 5. Matriks jarak 6 OTU

Gambar 5 menunjukan matriks jarak dari enam sekuen (OTU) yang terlibat. $\{x1, x2, x3, x4, x5, x6\}$ merupakan himpunan sekuen yang diperoleh dari enam spesies (OTU) yang berbeda. Setiap elemen matriks di atas merepresentasikan jarak antar OTU yang terlibat. Misalkan, jarak antara OTU $x1$ dan $x2$ adalah 8, angka 8 tersebut menyatakan bahwa perbedaan genetik sekuen $x1$ dan $x2$ adalah 8 satuan. Perbedaan tersebut dapat terjadi karena proses evolusi yang terjadi di dalam struktur genetiknya. Angka-angka tersebut bisa dikatakan sebagai waktu evolusi atau banyaknya perbedaan gen akibat evolusi. Pada tugas akhir ini, tidak dibahas bagaimana cara memperoleh matriks tersebut dan diasumsikan matriks sudah diberikan.

Secara formal, suatu matriks jarak dibentuk berdasarkan *distance function* yang didefinisikan sebagai berikut.

Definisi 1.6. Misalkan A adalah sebuah himpunan dan d : A $\times$ A $\rightarrow$ R adalah sebuah fungsi. *d* dikatakan sebagai *distance function* atau fungsi jarak pada A jika

(i). $d(u,v) > 0$ untuk setiap $u, v \in$ A, $u \neq v$

(ii). $d(u,u) = 0$ untuk setiap $u \in$ A

(iii). $d(u,v) = d(v,u)$ untuk setiap $u, v \in$ A

(iv). Memenuhi ketaksamaan segitiga: $d(u,v) \leq d(u,w) + d(w,v)$ untuk setiap $u, v, w \in$ A

Jika *d* adalah *distance function* pada A, maka untuk $u, v \in$ A bilangan $d(u,v)$ disebut sebagai jarak antara u dan v. Himpunan yang akan dipakai disini adalah himpunan terhingga A = $\{x1, x2, ..., xN\}$ yang merupakan himpunan sekuen (OTU) yang akan dibentuk *phylogenetic tree*-nya. Diasumsikan bahwa *distance function d* terdefinisi di A dan *d* relevan secara biologi, maksudnya adalah *d* sesuai dengan informasi genetik yang ada pada sekuen (OTU) di A. Sebagai contoh, $d(x1, x2) > d(x3, x4)$ berarti OTU $x1$ dengan $x2$ lebih jauh hubungan evolusi atau kekerabatannya dibanding OTU $x3$ dengan $x4$. Untuk menyederhanakan penulisan, $d(xi, xj)$ ditulis sebagai d_{ij} dengan $i, j \in \{1, 2, 3, ..., N\}$. Berdasarkan *distance function* tersebut

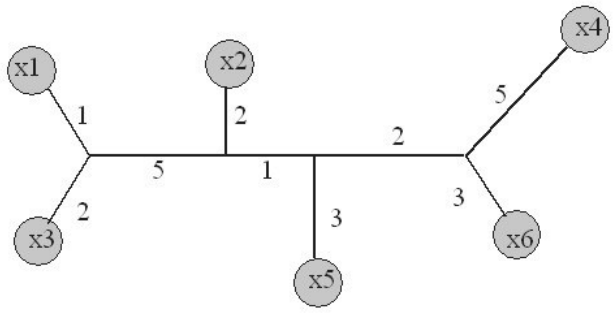
dapat diperoleh matriks jarak (*distance matrix*), $Md = \left(d_{ij}\right)$ dengan definisi formal sebagai berikut.

Definisi 1.7. Misalkan *d* adalah suatu *distance function*, Md disebut sebagai matriks jarak yang didefinisikan oleh

$$Md = \begin{pmatrix} d_{11} & d_{12} & \cdots & d_{1j} & \cdots & d_{1N} \\ d_{21} & d_{22} & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ d_{i1} & \ddots & \ddots & d_{ij} & \ddots & d_{iN} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ d_{N1} & d_{N2} & \cdots & d_{Nj} & \cdots & d_{NN} \end{pmatrix}$$

dengan $i, j = 1, 2, 3, ..., N$ dan N adalah jumlah OTU yang terlibat [1].

Pada tugas akhir ini, dari sebuah matriks jarak *tree* akan dibangun adalah sebuah *unrooted tree* yang representatif untuk menggambarkan matriks jarak dari sekuen (OTU). Jadi tujuan dari metode jarak untuk membangun *phylogenetic tree* ini adalah membangun *tree* yang jarak/panjang *edge* dari node-node yang terhubung, dekat atau sama dengan apa yang ada pada matriks jarak *d.* Sebagai ilustrasi, perhatikan *phylogenetic tree* pada gambar berikut.



Gambar 6. *Phylogenetic tree* dari 5 spesies/OTU

Tree pada Gambar 6 merupakan *tree* yang dibentuk berdasarkan matriks jarak pada Gambar 5. Bisa dilihat bahwa jarak $x1$ dengan $x2$ pada matriks jarak tersebut adalah 8 dan begitu pula dengan jarak $x1$ dengan $x2$ pada *tree* di atas, jika panjang *edge* yang menghubungkan *node* $x1$ dengan $x2$ dijumlahkan yaitu 1 + 5 + 2, maka jumlahnya adalah 8. Jika dibandingkan, semua *node* yang terhubung pada *tree* (Gambar 6) jaraknya sesuai dengan matriks jarak pembangunnya.

Namun tidak semua matriks jarak dapat membangun *tree* seperti ilustrasi di atas. Ada syarat yang harus dipenuhi agar suatu matriks jarak dapat membangun sebuah *tree*, yang dikenal sebagai sifat *additive.*

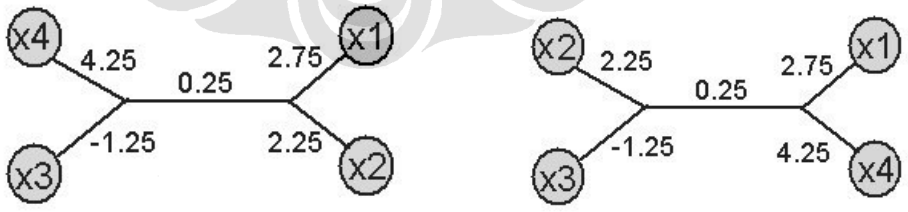
Teorema 1.8. Misalkan *d* adalah suatu *distance function* pada A dan $N \geq 4$. Maka *d additive* jika dan hanya jika untuk setiap himpunan 4 angka berbeda $1 \leq i, j, k, l \leq N$, dua dari penjumlahan $d_{ij} + d_{kl}, d_{ik} + d_{jl}, d_{il} + d_{jk}$ sama dan lebih besar atau sama dengan yang ketiga [1].

Kondisi pada teorema 1.8 disebut *four point condition.* Matriks jarak yang memenuhi kondisi di atas pasti dapat membangun sebuah *tree* yang unik, yang merepresentasikan matriks jarak tersebut. Namun, jika matriks jarak tidak bersifat *additive, tree* yang dihasilkan tidaklah unik dan tidak seutuhnya merepresentasikan matriks pembangunnya [1]. Perhatikan matriks jarak berikut.

Md	$x1$	$x2$	$x3$	$x4$
$x1$	0	5	2	7
$x2$	5	0	1	7
$x3$	2	1	0	3
$x4$	7	7	3	0

Gambar 7. Matriks jarak 4 spesies/OTU

Terlihat bahwa matriks jarak pada Gambar 7 tidak bersifat *additive* karena $d_{13}+d_{24}=9,\quad d_{12}+d_{34}=8,\quad d_{14}+d_{23}=8$. Dengan menggunakan algoritma *neighbor-joining* yang akan dijelaskan kemudian, matriks jarak tersebut dapat membentuk dua *tree* yang berbeda, yaitu *tree* pada Gambar 8.



(a) (b)

Gambar 8. *Tree* yang dihasilkan oleh matriks jarak yang sama

Karena tidak *additive, tree* yang dihasilkan dapat memiliki *edge* yang bernilai negatif seperti Gambar 8. Selain itu, *tree* juga tidak merepresentasikan matriks jarak seutuhnya. Pada Gambar 8(a), jarak $x1$ dan $x4$ adalah 4.25 + 0.25 + 2.75 = 7.25, sedangkan jarak $x1$ dan $x4$ pada matriks jarak di Gambar 7 adalah 7 atau $d_{14} = 7$. Hal yang sama pula terjadi pada Gambar 8(b) untuk pasangan $x1$ dan $x2$. Jadi sifat *additive* sangat menentukan *tree* yang dihasilkan dari matriks jarak pembangunnya.

Ada beberapa cara untuk membangun *tree* dari sebuah matriks jarak, diantaranya dengan menggunakan algoritma *neighbor-joining* (NJ) dan UPGMA (*Unweighted Pair Group Method using Arithmetic average*). Algoritma ini akan dijelaskan pada subbab berikutnya.

2.5 Algoritma pada Metode Jarak

Secara umum ada dua cara untuk membangun *tree* dari sebuah matriks jarak yaitu *clustering* dan *optimality*. Terdapat dua metode *clustering* yang paling umum yaitu metode dengan menggunakan algoritma *neighbor-joining* dan UPGMA. Sedangkan pada metode *optimality* terdapat dua metode, yaitu metode dengan menggunakan algoritma *fitch-margoliash* dan *minimum evolution* [2]. Ide dari metode *clustering* adalah menggabungkan sehimpunan node (OTU), pengelompokan didahulukan untuk node-node (OTU) yang terdekat secara genetik (dalam kasus ini jarak yang terpendek), metode ini selesai jika semua node (OTU) telah terhubung, *tree* yang

dihasilkan hanya satu jenis. Sedangkan pada metode *optimality*, semua kemungkinan *tree* akan dicari dan dibandingkan untuk memperoleh *tree* yang terbaik [4].

Pada penulisan tugas akhir ini hanya dibatasi pada metode *clustering* untuk algoritma *neighbor-joining*. Ide dari algoritma *neighbor-joining* ini adalah menemukan sepasang OTU yang terdekat hubungannya tetapi jauh dari OTU yang lainnya, kemudian pasangan tersebut akan dihubungkan ke sebuah *node* yang bertindak sebagai internal *node*, dalam hal ini nenek moyang bersama (*common ancestor*) dari kedua OTU tersebut. Algoritma akan selesai jika semua *node* telah terhubung.