

# Aplikasi Representasi Graf

Deasy Ramadiyan Sari

Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung  
Jalan Ganessa 10 Bandung 40132

Email: [if13008@students.if.itb.ac.id](mailto:if13008@students.if.itb.ac.id)

## Abstrak

Representasi graf sederhana dapat dilakukan dengan minimal tiga jenis representasi, yaitu matriks ketetanggaan, matriks kebersisian, dan senarai ketetanggaan. Banyak hal yang dapat direpresentasikan dalam graf sebenarnya, contohnya adalah XML. Basisdata *Oracle 10g* juga menggunakan representasi graf dalam fiturnya. Paper ini membahas mengenai representasi graf dalam XML, serta penerapan representasi graf pada fitur di dalam *Oracle 10g* dan penerapannya di dalam aplikasi berbasis basisdata *Oracle 10g*. Disamping itu, paper ini juga menyediakan contoh program untuk merepresentasikan graf berbobot kedalam tiga jenis representasi graf minimal diatas.

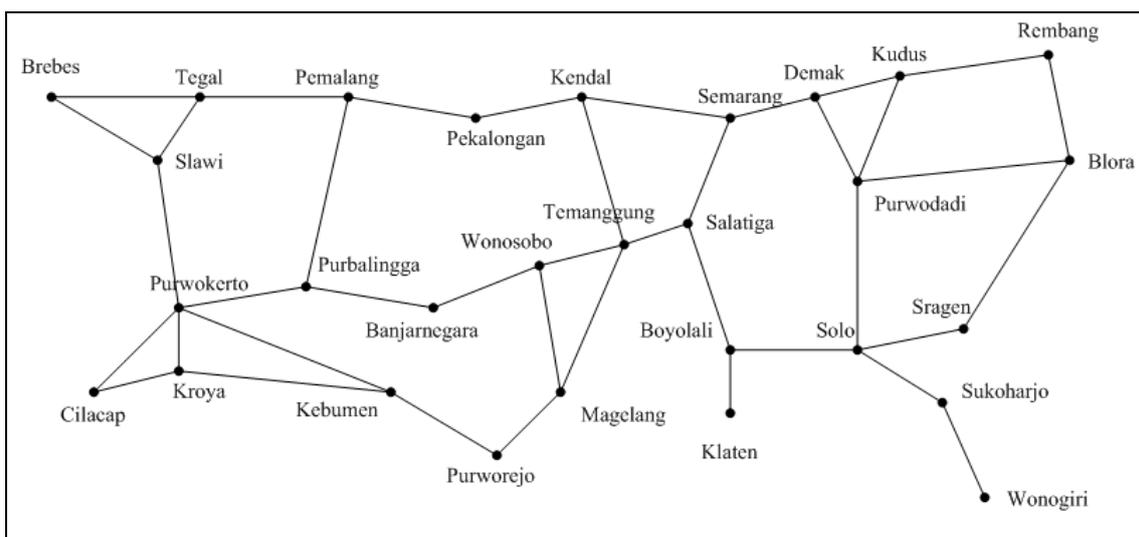
**Kata Kunci:** representasi graf, xml, *Oracle 10g*, graf berbobot

## 1 Pendahuluan

### 1.1 Teori Graf

Graf digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut. Representasi visual dari graf adalah dengan menyatakan objek sebagai noktah, bulatan, atau titik, sedangkan

hubungan antara objek dinyatakan dengan garis. Sebagai contoh, Gambar 1 di bawah ini merupakan sebuah graf yang menyatakan peta jaringan jalan raya yang menghubungkan sejumlah kota di Provinsi Jawa Tengah. Dengan diberikan graf tersebut maka kita dapat mengetahui apakah ada lintasan jalan antara dua buah kota.



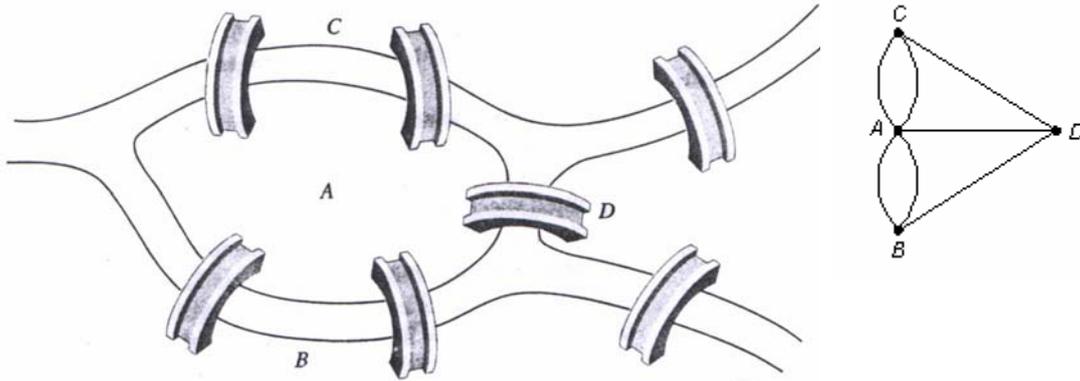
Gambar 1 Peta Jaringan Jalan Raya

## 1.2 Sejarah Graf

Menurut catatan sejarah, masalah jembatan Königsberg (tahun 1736) adalah masalah pertama kali yang menggunakan graf. Di kota Königsberg, terdapat sungai Pregal yang mengalir mengitari pulau Kneiphof lalu bercabang lagi menjadi dua buah anak sungai. Ada tujuh buah jembatan yang menghubungkan daratan yang dibelah oleh sungai tersebut. Masalah jembatan ini adalah,

mungkinkah melalui seluruh jembatan tepat satu kali, dan kembali lagi ke tempat semula? Seorang matematikawan Swiss, L. Euler menemukan jawaban mengenai masalah tersebut dengan pembuktian sederhana. Dengan memodelkan dalam graf. Graf yang merepresentasikan jembatan Königsberg:

1. Simpul (*vertex*) : { a, b, c, d } daratan
2. Sisi (*edge*) : { (a,b), (b,a), (a,c), (c,a), (a,d), (b,c), (c,d) } menyatakan jembatan



Gambar 2 Jembatan Königsberg

Jawaban yang diperoleh saat itu adalah, orang tidak mungkin melalui ketujuh jembatan tersebut masing-masing satu kali dan kembali lagi ke tempat asal keberangkatan jika jumlah derajat setiap simpul tidak genap.

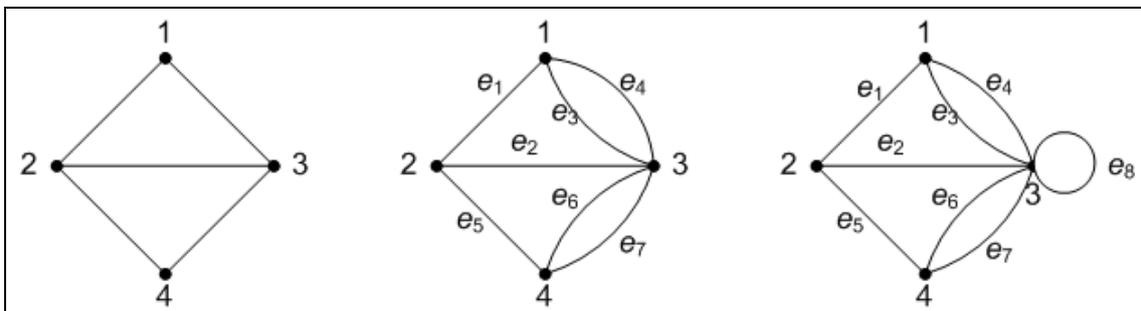
$E$  = himpunan sisi (*edges*) yang menghubungkan sepasang simpul  
 $= \{e_1, e_2, \dots, e_n\}$   
 Atau dapat ditulis dengan notasi  $G = (V, E)$

## 1.3 Definisi Graf

Graf  $G$  didefinisikan sebagai pasangan himpunana  $(V, E)$  yang dalam hal ini:

$V$  = himpunan tidak-kosong dari simpul-simpul (*vertices*)  
 $= \{v_1, v_2, \dots, v_n\}$

Simpul pada graf dapat dinomori dengan huruf seperti  $v, w, \dots$ , dengan bilangan asli 1, 2, 3, ... atau gabungan keduanya. Sedangkan sisi yang menghubungkan simpul  $v_i$  dengan  $v_j$  dinyatakan dengan pasangan  $(v_i, v_j)$  atau dengan lambang  $e_1, e_2, \dots$ . Dengan kata lain, jika  $e$  adalah sisi yang menghubungkan simpul  $v_i$  dengan  $v_j$ , maka  $e$  dapat ditulis sebagai  $e = (v_i, v_j)$ . Gambar 3 menunjukkan beberapa contoh graf.



Gambar 3 Contoh Graf

## 1.4 Jenis-Jenis Graf

Graf dapat dikelompokkan menjadi beberapa jenis bergantung pada sudut pandang pengelompokkannya. Pengelompokkan graf dapat dipandang berdasarkan ada tidaknya sisi ganda atau sisi kalang, berdasarkan jumlah simpul, atau berdasarkan orientasi arah pada sisi.

Berdasarkan ada tidaknya gelang atau sisi ganda pada suatu graf, maka graf digolongkan menjadi dua jenis:

### 1. Graf sederhana (*simple graph*)

Graf yang tidak mengandung gelang maupun sisi-ganda dinamakan graf sederhana.  $G_1$  pada Gambar 3 adalah contoh graf sederhana

### 2. Graf tak-sederhana (*unsimple-graph*)

Graf yang mengandung sisi ganda atau gelang dinamakan graf tak-sederhana (*unsimple graph*).  $G_2$  dan  $G_3$  pada Gambar 3 adalah contoh graf tak-sederhana

Berdasarkan jumlah simpul pada suatu graf, maka secara umum graf dapat digolongkan menjadi dua jenis:

### 1. Graf berhingga (*limited graph*)

Graf berhingga adalah graf yang jumlah simpulnya  $n$  berhingga.

### 2. Graf tak-berhingga (*unlimited graph*)

Graf yang jumlah simpulnya  $n$  tidak berhingga

Berdasarkan orientasi arah pada sisi, maka secara umum graf dibedakan atas 2 jenis:

### 1. Graf tak-berarah (*undirected graph*)

Graf yang sisinya tidak mempunyai orientasi arah disebut graf tak-berarah. Tiga buah graf pada Gambar 2 adalah graf tak-berarah.

### 2. Graf berarah (*directed graph* atau *digraph*)

Graf yang setiap sisinya diberikan orientasi arah disebut sebagai graf berarah. Dua buah graf pada Gambar 3 adalah graf berarah.

## 1.5 Representasi Graf

Untuk maksud pemrosesan graf dengan program komputer, graf harus direpresentasikan di dalam memori. Terdapat beberapa representasi yang mungkin untuk graf. Di sini hanya diberikan tiga macam

representasi yang sering digunakan pada graf sederhana, yaitu matriks ketetanggaan, matriks bersisian, dan senarai ketetanggaan.

### 1.5.1 Matriks Ketetanggaan (*Adjacency Matrix*)

Untuk mempermudah perhitungan pada program komputer, graf dapat direpresentasikan dengan menggunakan matriks. Salah satunya adalah matriks ketetanggaan.

Misalkan  $G = (V, E)$  adalah sebuah graf sederhana dimana  $|V| = n$ ,  $n > 1$ . Misalkan simpul dari  $G$  adalah  $v_1, v_2, \dots, v_n$ . Maka, matriks ketetanggaan  $A$  dari  $G$  adalah  $n \times n$  matriks dimana:

$$A = [a_{ij}], \quad a_{ij} = \begin{cases} 1, & \text{jika simpul } i \\ & \text{dan } j \text{ bertetangga} \\ 0, & \text{jika simpul } i \\ & \text{dan } j \text{ tidak bertetangga} \end{cases}$$

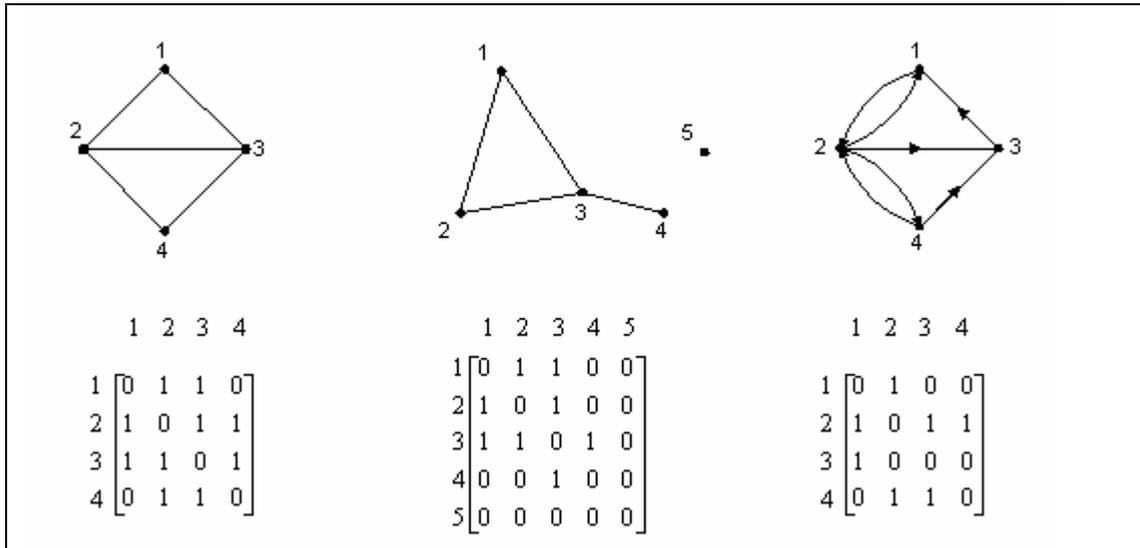
Matriks ketetanggaan dari sebuah graf mengacu pada keterurutan dari simpul. Sehingga, ada sebanyak  $n!$  keterurutan yang berbeda yang terbentuk dari  $n$  simpul.

Matriks ketetanggaan merupakan graf sederhana yang simetris, yaitu  $a_{ij} = a_{ji}$ . Hal ini disebabkan oleh kedua-duanya adalah 1 ketika  $v_i$  dan  $v_j$  mempunyai sisi, dan adalah 0 jika tidak ada sisi diantaranya, dinamakan juga **matriks nol-satu**, sedangkan untuk graf berarah, matriks ketetanggaannya belum tentu simetri (akan berupa simetri jika berupa graf berarah lengkap). Selain itu, graf sederhana tidak mempunyai gelang, sehingga diagonal utamanya selalu 0 karena  $a_{ii} = 0$ ,  $i = 1, 2, 3, \dots, n$  adalah 0.

Sayangnya, matriks ketetanggaan nol-satu tidak dapat digunakan untuk merepresentasikan graf yang mempunyai sisi ganda. Untuk mengatasinya, maka elemen  $a_{ij}$  pada matriks ketetanggaan sama dengan jumlah yang berasosiasi dengan  $(v_i, v_j)$ . Matriks ketetanggaannya bukan lagi matriks nol-satu. Untuk graf semu, gelang pada simpul  $v_i$  dinyatakan dengan nilai satu pada posisi  $(i, i)$  di matriks ketetanggaannya.

Keuntungan representasi dengan matriks adalah, elemen matriksnya dapat diakses langsung melalui indeks. Selain itu, kita dapat

mengetahui secara langsung apakah simpul  $i$  dan simpul  $j$  bertetangga.



Gambar 4 Graf dengan Matriks Ketetangaan

Derajat tiap simpul  $i$ :

(a) Untuk graf tak-berarah

$$d(v_i) = \sum_{j=1}^n a_{ij}$$

(b) Untuk graf berarah,

$d_{in}(v_j)$  = jumlah nilai pada kolom  $j$

$$= \sum_{i=1}^n a_{ij}$$

$d_{out}(v_i)$  = jumlah nilai pada baris  $i$

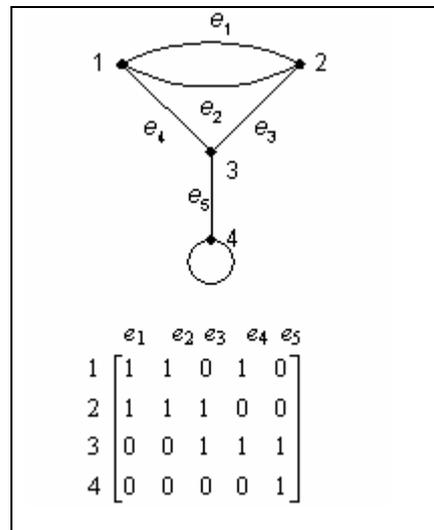
$$= \sum_{j=1}^n a_{ij}$$

$$A = [a_{ij}],$$

$$a_{ij} = \begin{cases} 1, & \text{jika simpul } i \text{ bersisian dengan sisi } j \\ 0, & \text{jika } i \text{ tidak bersisian dengan sisi } j \end{cases}$$

### 1.6 Matriks Bersisian (Incidency Matrix)

Bila matriks ketetangaan menyatakan ketetangaan simpul-simpul di dalam graf, maka matriks bersisian menyatakan kebersisian simpul dengan sisi. Misalkan  $G = (V, E)$  adalah graf dengan  $n$  simpul dan  $m$  buah sisi, maka matriks kebersisian  $A$  dari graf  $G$  adalah matriks yang berukuran  $m \times n$ . Baris menunjukkan label sisi, sementara kolom menunjukkan label sisinya. Gambar 5 menunjukkan representasi graf dalam bentuk matriks bersisian.



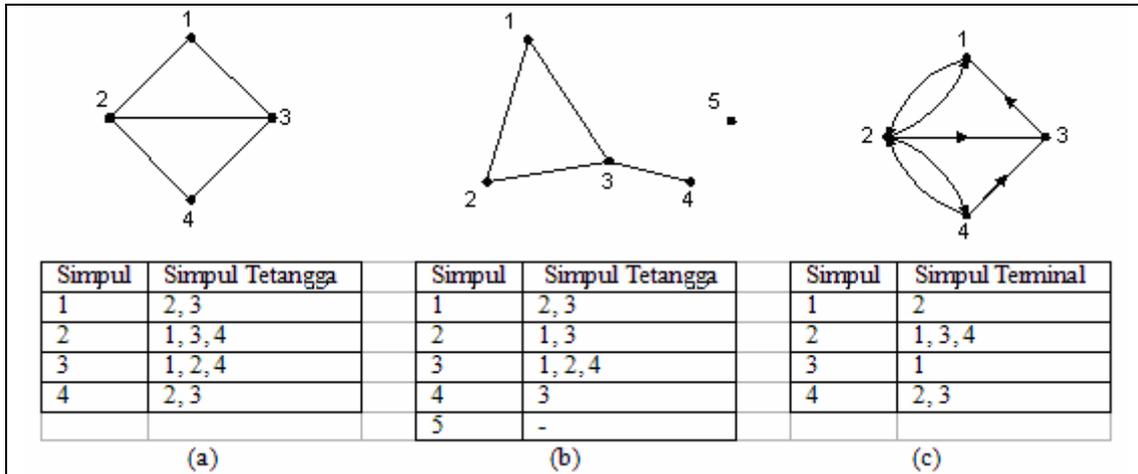
Gambar 5 Graf dengan Matriks Bersisian

Matriks bersisian dapat digunakan untuk merepresentasikan graf yang mengandung sisi ganda atau sisi gelang. Derajat tiap simpul  $i$  dapat dihitung dengan menghitung jumlah seluruh elemen pada baris  $i$  (kecuali pada graf yang mengandung gelang).

## 1.7 Senarai Ketetangaan (Adjacency List)

Kelemahan matriks ketetangaan adalah bila graf memiliki jumlah sisi relatif sedikit, karena matriksnya bersifat jarang, yaitu banyak mengandung elemen nol, sedangkan elemen yang bukan nol sedikit. Ditinjau dari segi implementasi, kebutuhan ruang memory untuk

matriks jarang, boros, karena komputer menyimpan banyak elemen nol yang tidak perlu. Untuk mengatasi masalah ini, senarai ketetangaan digunakan. Senarai ketetangaan mengenumerasi simpul-simpul yang bertetangga dengan setiap simpul di dalam graf. Contoh dari senarai ketetangaan dapat dilihat dari Gambar 6.



Gambar 6 Graf dengan Senarai Ketetangaan

## 1.8 Representasi Graf Lain

Disamping ketiga macam representasi graf diatas, ada banyak hal yang dapat dengan mudah untuk di ubah menjadi bentuk graf. Diantaranya adalah file XML. Mengubah sebuah file XML menjadi bentuk graf sebenarnya tidaklah sesulit yang kita

bayangkan. Sebenarnya, bentuk dari XML itu sendiri sudah merupakan srtuktur pohon, *tag-tag* dalam XML adalah *nodes* atau simpul dari pohon, sehingga memudahkan bagi kita untuk dapat mengubah file XML menjadi graf. Lihatlah contoh di bawah ini untuk membuktikannya:

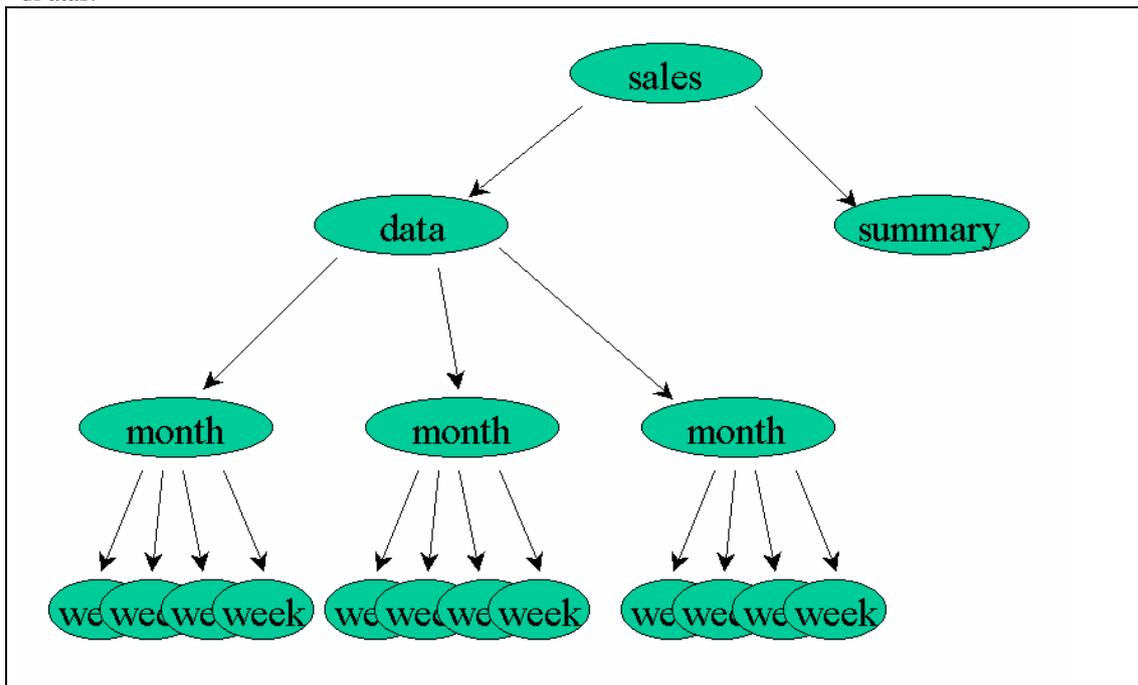
```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="mydvd.xsl"?>
<sales>
  <summary>
    <heading>MyDVD Rental Store</heading>
    <subhead>Periodical Sales Report</subhead>
    <description>Sales Report for January,
      February, and March of 2001</description>
  </summary>
  <data>
    <month>
      <name>January 2001</name>
      <week number="1" dvds_rented="12000" />
      <week number="2" dvds_rented="15000" />
      <week number="3" dvds_rented="18000" />
      <week number="4" dvds_rented="11800" />
    </month>
    <month>
      <name>February 2001</name>
      <week number="1" dvds_rented="11500" />
      <week number="2" dvds_rented="12390" />
      <week number="3" dvds_rented="19050" />
    </month>
  </data>
</sales>
```

```

    <week number="4" dvds_rented="11200" />
  </month>
  <month>
    <name>March 2001</name>
    <week number="1" dvds_rented="15300" />
    <week number="2" dvds_rented="12390" />
    <week number="3" dvds_rented="10050" />
    <week number="4" dvds_rented="11230" />
  </month>
</data>
</sales>

```

Berikut adalah hasil representasi dari dokumen di atas:



Gambar 7 Representasi Graf dari XML

## 2 Graf Berbobot

### 2.1 Pengertian

Graf yang memiliki nilai disetiap sisinya dinamakan graf berbobot. Graf berbobot biasanya digunakan untuk memodelkan jaringan komputer. Biaya komunikasi, waktu respon dari komputer melalui jalur ini, atau jarak antar komputer, semuanya dapat dipelajari melalui graf berbobot.

### 2.2 Representasi Graf Berbobot

Dari beberapa representasi graf sederhana yang disebut diatas, representasi tersebut dapat juga diterapkan dalam graf berbobot. Sebuah program dibuat untuk merepresentasikan graf

berbobot ini kedalam tiga jenis representasi yaitu :

1. Matrix Ketetangaan (*Adjacency Matrix*)
2. Senarai Ketetangaan (*Adjacency List*)
3. Matrix Bersisian (*Incidency Matrix*)

### 2.3 Program Representasi Graf Berbobot

Program ini merepresentasikan graf berarah berbobot dan graf tidak berarah berbobot.

Program dikembangkan dalam bahasa pemrograman Java versi 1.5. Program terdiri dari empat buah file *class*, yaitu:

1. RepresentasiGrafGui.java: *class* antarmuka yang menggabungkan ketiga *class* representasi graf

2. AdjecencyMatrix.java: *class* representasi graf dalam bentuk *Adjecency Matrix*
3. AdjecencyList.java: *class* representasi graf dalam bentuk *Adjecency List*
4. IncidencyMatrix.java: *class* representasi graf dalam bentuk *Incidency Matrix*

### 2.3.1 Input

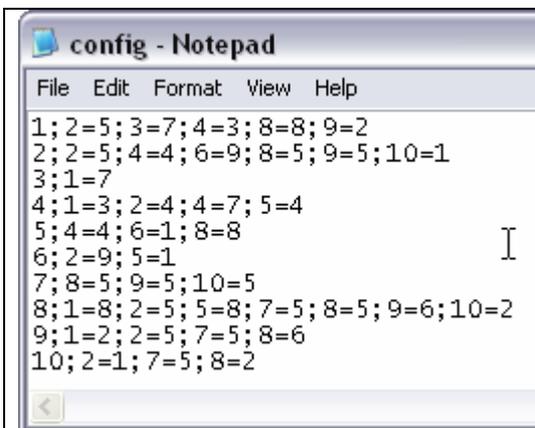
Program membaca graf dari file konfigurasi yang dibuat oleh pengguna, sehingga jika pengguna ingin merepresentasi graf yang dimilikinya dalam program komputer ini, maka graf harus ditransfer dalam bentuk file konfigurasi.

Representasi graf dalam bentuk *Adjecency Matrix* dan *Adjecency List* akan menggunakan file konfigurasi dengan nama config.txt.

Format penulisan file konfigurasi adalah sebagai berikut:

1. Setiap baris melambangkan sebuah simpul pada graf
2. Setiap simpul dilengkapi dengan keterangan simpul-simpul tetangganya yang dipisahkan dengan “;” dan disertai bobotnya masing-masing, yang ditandai dengan lambang “=”

Contoh bentuk file konfigurasi config.txt :



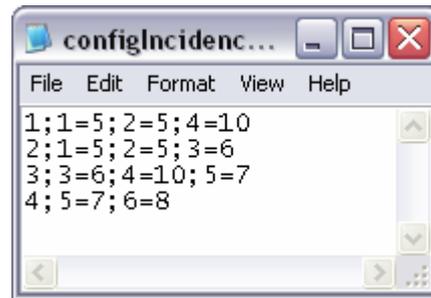
Gambar 8 config.txt

Pada file konfigurasi tersebut dapat terlihat bahwa simpul satu bertetangga dengan simpul 2 dengan bobot 5, bertetangga dengan simpul 3 berbobot 7, dan seterusnya. Untuk representasi *Adjecency List*, ketetanggan tersebut melambangkan simpul keluar dari suatu simpul. Artinya, simpul keluar dari simpul satu adalah simpul 2, 5, 4, 8, dan 9.

Representasi graf dalam bentuk *Incidency Matrix* memiliki file konfigurasi sendiri, yaitu configIncidency1.txt dan configIncidency2.txt. Format penulisannya kedua file konfigurasi sama, yaitu:

1. Setiap baris melambangkan sebuah simpul pada graf
2. Setiap simpul dilengkapi dengan keterangan sisi-sisi tetangganya yang dipisahkan dengan “;” dan bobot yang ditandai dengan lambang “=”

Contoh bentuk file konfigurasi configIncidency1.txt:



Gambar 9 configIncidency1.txt

Bentuknya file konfigurasi configIncidency1.txt dan configIncidency2.txt sama dengan config.txt, namun merepresentasikan hal yang berbeda. Jika config.txt merupakan representasi simpul dengan simpul tetangga, configIncidency1.txt dan configIncidency2.txt merepresentasikan simpul dengan sisi yang bersisian.

Untuk kasus graf berbobot tidak berarah dengan representasi *Incidency Matrix*, hanya file configIncidency1.txt saja yang digunakan, sedangkan untuk kasus graf berbobot berarah dengan representasi *Incidency Matrix*, configIncidency1.txt dan configIncidency2.txt digunakan. File configIncidency1.txt digunakan untuk merepresentasikan simpul yang bersisian dengan sisi keluar, sedangkan file configIncidency2.txt digunakan untuk merepresentasikan simpul yang bersisian dengan sisi masuk.

### 2.3.2 Proses

Setelah file konfigurasi dibuat maka pengguna dapat mengeksekusi program dengan melakukan klik ganda pada file repGraf.bat. Antarmuka pertama yang ditampilkan ke pengguna adalah memilih representasi graf yang diinginkan.



Gambar 10 Antarmuka Pertama

Pada umumnya, antarmuka yang ditampilkan pada saat berinteraksi dengan pengguna adalah sama. Perbedaannya adalah dari cara program merepresentasikan graf sesuai dengan pilihan kita. Sama ada adjacency matrix, adjacency list, *Incidency Matrix*. Pengguna akan diminta untuk memasukkan jumlah simpul maksimum yang dimungkinkan.



Gambar 11 Antarmuka Masukan Jumlah Simpul

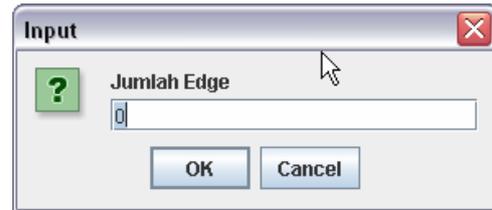
Jumlah simpul yang dimasukkan oleh pengguna harus dipastikan lebih besar atau sama dengan jumlah simpul yang ada di file konfigurasi config.txt jika memilih adjacency matrix atau adjacency list dan configIncidency1.txt jika memilih *Incidency Matrix* yang tidak berarah.

Selanjutnya pengguna akan diminta untuk memasukkan pilihan graf berarah atau tidak berarah sebelum eksekusi dilakukan.



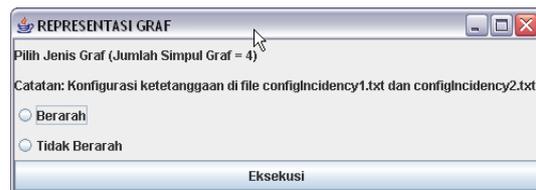
Gambar 12 Antarmuka Pemilihan Jenis Graf Berorientasi Arah

Jika pengguna memilih representasi graf dengan menggunakan *Incidency Matrix*, maka akan diminta untuk memasukkan jumlah edge/sisi maksimum yang dimungkinkan.



Gambar 13 Antarmuka Masukan Jumlah Sisi

Selanjutnya pengguna akan diminta untuk memasukkan pilihan graf berarah atau tidak berarah sebelum eksekusi bisa dilakukan.



Gambar 14 Antarmuka Pemilihan Jenis Graf Berorientasi Arah

### 2.3.3 Output

Output dari program akan ditampilkan dalam console. Output yang dihasilkan oleh program adalah representasi graf dalam bentuk matrix (untuk *Adjacency Matrix* dan *Incidency Matrix*) atau bentuk node beserta tetangganya (untuk *Adjacency List*). Selain itu derajat setiap simpul akan ditampilkan (untuk graf tidak berarah) atau derajat masuk dan keluar setiap simpul (untuk graf berarah). Berikut ditampilkan contoh output program untuk masing-masing representasi.

#### *Adjacency Matrix*

Contoh kasus graf tidak berarah:

```
C:\WINDOWS\system32\cmd.exe
REPRESENTASI GRAF BERARAH DALAM MATRIX
0 5 7 3 0 0 0 8 2 0
5 5 0 4 0 9 0 0 5 5 1
7 0 0 0 0 0 0 0 0 0 0
3 4 0 7 4 0 0 0 0 0 0
0 0 0 4 0 1 0 8 0 0 0
0 9 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 5 5 5 5
8 5 0 0 8 0 5 5 6 2 2
2 5 0 0 0 0 5 6 0 0 0
0 1 0 0 0 0 5 2 0 0 0
Derajat Masuk Simpul ke 1 = 5
Derajat Masuk Simpul ke 2 = 6
Derajat Masuk Simpul ke 3 = 1
Derajat Masuk Simpul ke 4 = 4
Derajat Masuk Simpul ke 5 = 3
Derajat Masuk Simpul ke 6 = 2
Derajat Masuk Simpul ke 7 = 2
Derajat Masuk Simpul ke 8 = 6
Derajat Masuk Simpul ke 9 = 4
Derajat Masuk Simpul ke 10 = 0
Derajat Luar Simpul ke 1 = 5
Derajat Luar Simpul ke 2 = 7
Derajat Luar Simpul ke 3 = 1
Derajat Luar Simpul ke 4 = 4
Derajat Luar Simpul ke 5 = 3
Derajat Luar Simpul ke 6 = 2
Derajat Luar Simpul ke 7 = 3
Derajat Luar Simpul ke 8 = 7
Derajat Luar Simpul ke 9 = 4
Derajat Luar Simpul ke 10 = 3
```

Gambar 15 Hasil Representasi Matriks Ketetanggaan Graf Tidak Berarah

Contoh kasus graf berarah:

```

C:\WINDOWS\system32\cmd.exe
D:\Kuliah\Matematika Informatik\RepGraf>java -cp ./bin com.rep.graf.Representasi
GrafGui
REPRESENTASI GRAF TIDAK BERARAH DALAM Matriks
0 5 7 3 0 0 0 8 2 0
5 5 0 4 0 9 0 0 5 5 1
7 0 0 0 0 0 0 0 0 0 0
3 4 0 7 4 0 0 0 0 0 0
0 0 0 4 0 1 0 0 0 0 0
0 9 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 5 5 5 5
0 5 0 0 8 0 5 5 6 2 2
2 5 0 0 0 0 5 6 0 0 0
0 1 0 0 0 0 5 2 0 0 0

Derajat Simpul ke 1 = 5
Derajat Simpul ke 2 = 7
Derajat Simpul ke 3 = 1
Derajat Simpul ke 4 = 4
Derajat Simpul ke 5 = 3
Derajat Simpul ke 6 = 2
Derajat Simpul ke 7 = 3
Derajat Simpul ke 8 = 7
Derajat Simpul ke 9 = 4
Derajat Simpul ke 10 = 3
    
```

Gambar 16 Hasil Representasi Matriks Ketetangaan Graf Berarah

Adjacency List

Contoh kasus graf tidak berarah:

```

C:\WINDOWS\system32\cmd.exe
Node 1 dengan bobot : 8
Node 2 dengan bobot : 5
Node 5 dengan bobot : 8
Node 7 dengan bobot : 5
Node 8 dengan bobot : 5
Node 9 dengan bobot : 6
Node 10 dengan bobot : 2

Node 9 ,bertetangga dengan :
Node 1 dengan bobot : 2
Node 2 dengan bobot : 5
Node 7 dengan bobot : 5
Node 8 dengan bobot : 6

Node 10 ,bertetangga dengan :
Node 2 dengan bobot : 1
Node 7 dengan bobot : 5
Node 8 dengan bobot : 2

Derajat Simpul ke 1 = 5
Derajat Simpul ke 2 = 6
Derajat Simpul ke 3 = 1
Derajat Simpul ke 4 = 4
Derajat Simpul ke 5 = 3
Derajat Simpul ke 6 = 2
Derajat Simpul ke 7 = 3
Derajat Simpul ke 8 = 7
Derajat Simpul ke 9 = 4
Derajat Simpul ke 10 = 3
    
```

Gambar 17 Hasil Representasi Senarai Ketetangaan Graf Tidak Berarah

Contoh kasus graf berarah:

```

C:\WINDOWS\system32\cmd.exe
Node 9 ,bertetangga dengan :
Node 1 dengan bobot : 2
Node 2 dengan bobot : 5
Node 7 dengan bobot : 5
Node 8 dengan bobot : 6

Node 10 ,bertetangga dengan :
Node 2 dengan bobot : 1
Node 7 dengan bobot : 5
Node 8 dengan bobot : 2

Derajat Masuk Simpul ke 1 = 4
Derajat Masuk Simpul ke 2 = 7
Derajat Masuk Simpul ke 3 = 1
Derajat Masuk Simpul ke 4 = 4
Derajat Masuk Simpul ke 5 = 3
Derajat Masuk Simpul ke 6 = 2
Derajat Masuk Simpul ke 7 = 3
Derajat Masuk Simpul ke 8 = 7
Derajat Masuk Simpul ke 9 = 4
Derajat Masuk Simpul ke 10 = 3
Derajat Keluar Simpul ke 1 = 5
Derajat Keluar Simpul ke 2 = 6
Derajat Keluar Simpul ke 3 = 1
Derajat Keluar Simpul ke 4 = 4
Derajat Keluar Simpul ke 5 = 3
Derajat Keluar Simpul ke 6 = 2
Derajat Keluar Simpul ke 7 = 3
Derajat Keluar Simpul ke 8 = 7
Derajat Keluar Simpul ke 9 = 4
Derajat Keluar Simpul ke 10 = 3
    
```

Gambar 18 Hasil Representasi Senarai Ketetangaan Graf Berarah

Matrix

Contoh kasus graf tidak berarah:

```

C:\WINDOWS\system32\cmd.exe
D:\Kuliah\Matematika Informatik\RepGraf>java -cp
GrafGui
REPRESENTASI GRAF TIDAK BERARAH DALAM INCIDENCY
5 5 0 10 0 0
5 5 6 0 0 0
0 0 6 10 7 0
0 0 0 0 7 8

Derajat Simpul ke 1 = 3
Derajat Simpul ke 2 = 3
Derajat Simpul ke 3 = 3
Derajat Simpul ke 4 = 2
    
```

Gambar 19 Hasil Representasi Matriks Bersisian Graf Tidak Berarah

Contoh kasus graf berarah:

```

C:\WINDOWS\system32\cmd.exe
REPRESENTASI GRAF BERARAH DALAM INCIDENCY MATRIX
5 5 0 10 0 0
5 5 6 0 0 0
0 0 6 10 7 0
0 0 0 0 7 8

Derajat Masuk Simpul ke 1 = 2
Derajat Masuk Simpul ke 2 = 2
Derajat Masuk Simpul ke 3 = 2
Derajat Masuk Simpul ke 4 = 2
Derajat Keluar Simpul ke 1 = 3
Derajat Keluar Simpul ke 2 = 3
Derajat Keluar Simpul ke 3 = 3
Derajat Keluar Simpul ke 4 = 2
    
```

Gambar 20 Hasil Representasi Matriks Bersisian Graf Berarah

2.3.4 Evaluasi dan Keterbatasan

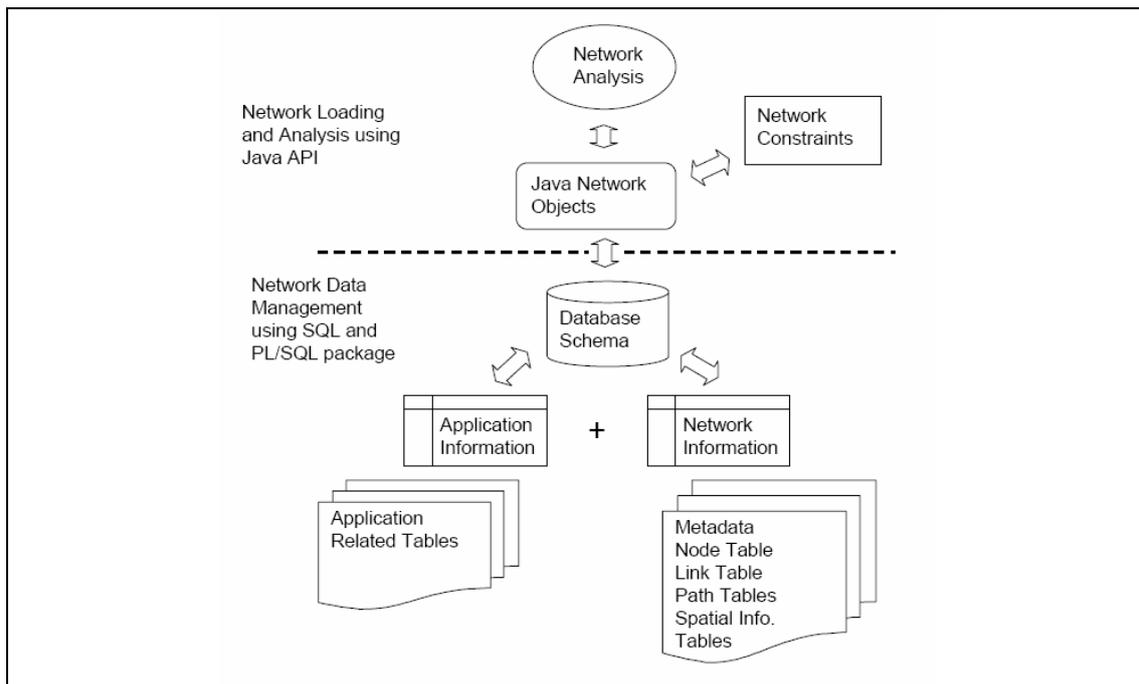
Pada program ini, jumlah simpul maksimum yang dimasukkan harus lebih besar atau sama dengan jumlah simpul yang ada pada file konfigurasi. Jika pengguna memasukkan jumlah simpul yang kurang dari simpul yang ada pada file konfigurasi, maka pesan error akan muncul.

## 3 Aplikasi Representasi Graf pada Oracle 10g

### 3.1 Pengenalan

Oracle 10g merupakan produk basisdata yang kaya akan fitur dan paling ditunggu dan bermanfaat. *Oracle 10g* mempunyai fungsionalitas untuk memodelkan data dalam bentuk graf, dan terlebih lagi mempunyai kemampuan untuk memfasilitasi pencarian.

Kemampuan 10g untuk memodelkan dan menganalisis data dalam bentuk graf karena 10g menerapkan *Oracle Spatial Network Data Model* (NDM). NDM menyimpan objek sesuai dengan simpulnya, hubungan antar simpul tersebut sebagai link, dan list terurut dari link-link tersebut yang didalamnya tidak ada link yang berulang atau simpul yang berulang dari *path* yang dilalui. NDM dapat digunakan untuk merepresentasikan graf berarah, graf tidak berarah, graf random, graf *scale free*, dan graf hirarki.



Gambar 21 Arsitektur Oracle Spatial Network Model Data

Skema NDM mengandung tabel untuk menyimpan simpul, sisi, dan data path, dan jaringan metadata. Skema NDM juga mengandung paket PL/SQL paket untuk *query* jaringan data, memelihara *referential integrity*, dan memperbarui sisi, simpul, dan table *path*.

NDM juga menyediakan kemampuan analisis untuk melacak (*tracing*), mencari *shortest path*, dan *minimum cost spanning tree*. Selain itu juga, NDM mendukung batasan jaringan yang digunakan untuk membatasi untuk analisis jaringan, termasuk *minimum bounding rectangle*, *path cost*, dan *path depth*.

## 3.2 Studi Kasus Network Data Model (NDM)

### 3.2.1 Analisis Metabolic Pathway

Model data grafik sesuai untuk menyimpan dan meneliti *pathway* metabolisme. Data dari KEGG berkenaan dengan metabolisme *pathway* basisdata dari Universitas Kyoto terisi ke dalam NDM sebagai jaringan berarah. Campuran bahan kimia disimpan sebagai simpul (Tabel 1) dan enzim sebagai link (Tabel 2) dalam Gambar 9. Ketika grafik tersimpan di dalam NDM, Java *visualizer* digunakan untuk melihat *pathway* metabolis dan untuk menanyai data menggunakan analisis yang mencakup *shortest path* dan *all path*.

Table 1: Pathway Data Definition: Node table

NODE_ID	NODE_NAME	ACT	COSTS	SAMPLE_ID	ENTRY_ID
1	C00022	Y	1	Pyruvate	49
2	C00122	Y	1	Fumarate	50
3	C00036	Y	1	Oxaloacetate	51
4	C05379	Y	1	Oxalosuccinate	52
5	C00074	Y	1	Phosphoenolpyruvate	53
6	C00024	Y	1	Acetyl-CoA	54
7	C00149	Y	1	(S)-Malate	55
8	C00311	Y	1	Isocitrate	56
9	C00417	Y	1	cis-Aconitate	57
10	C00042	Y	1	Succinate	58

Table 2: Pathway Data Definition Link table

LINK_ID	LINK_NAME	START_NODE_ID	END_NODE_ID	ACT	COST	SAMPLE_ID
1	1.1.1.42 (m:R00268)	4	19	Y	1	isocitrate dehydrogenase (NADP)
2	1.1.1.42 (m:R00268)	19	4	Y	1	isocitrate dehydrogenase (NADP)
3	1.1.1.42 (m:R00268)	4	20	Y	1	isocitrate dehydrogenase (NADP)
4	1.1.1.42 (m:R00268)	20	4	Y	1	isocitrate dehydrogenase (NADP)
5	4.1.1.49 (m:R00341)	3	19	Y	1	phosphoenolpyruvate carboxykinase (ATP)
6	4.1.1.49 (m:R00341)	3	5	Y	1	phosphoenolpyruvate carboxykinase (ATP)
7	1.1.1.37 (m:R00342)	3	7	Y	1	malate dehydrogenase
8	6.4.1.1 (m:R00344)	1	3	Y	1	pyruvate carboxylase

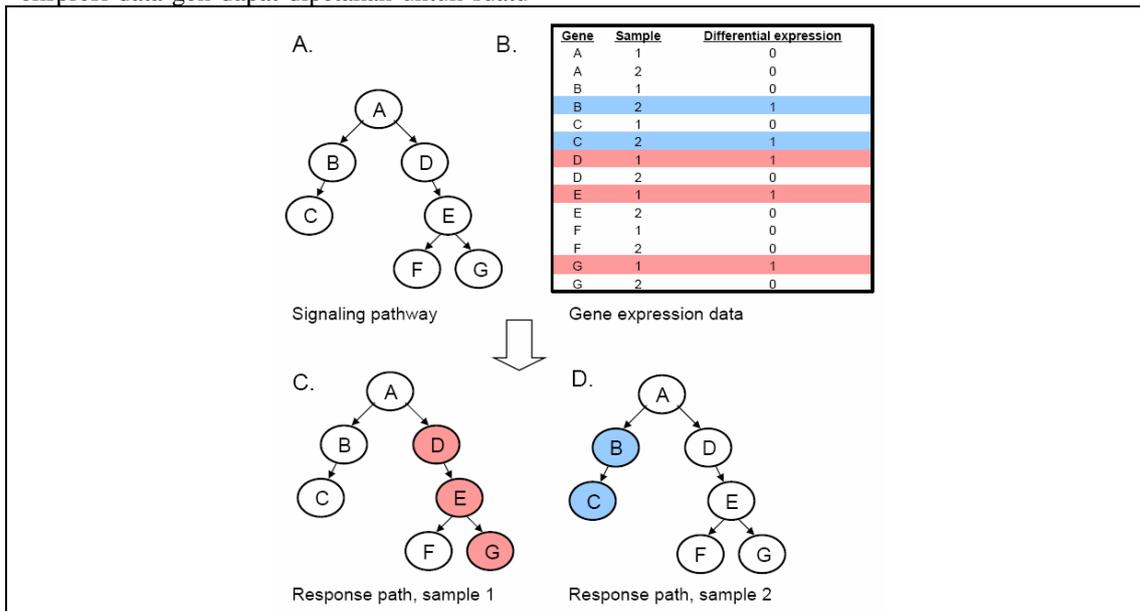
Gambar 22 Simpul dan Link

### 3.2.2 Analisis Jaringan Gen

Jaringan gen mendeskripsikan relasi dalam sebuah grup dari gen. Pendekatan yang umum untuk analisis jaringan gen, termasuk didalamnya menggunakan sisi untuk menciptakan korelasi antara profil ekspresi gen, informasi timbal balik, atau dampak suatu mutasi gen pada ekspresi gen. Contoh lebih rumit meliputi penggunaan kerangka matematik dari model probabilitas, seperti *Bayesian Network*, untuk belajar bagaimana struktur dan parameter suatu jaringan cocok dengan data yang diamati.

Studi kasus ini menguraikan bagaimana ekspresi data gen dapat dipetakan untuk suatu

pemberian isyarat *pathway*, dengan demikian menyediakan pengertian tambahan yang mendalam terhadap komponen. Di dalam contoh ini, pemberian isyarat *pathway* disimpan di dalam NDM dengan gen diwakili dalam simpul dan interaksi antar gen dalam sisi.]. *ekspresi* gen Data dari contoh biologi disimpan di (dalam) suatu relational terpisah. *Query* dapat ditampilkan, yaitu kedua-duanya pemberian isyarat *pathway* dan data *ekspresi* gen, memungkinkan peneliti untuk mengidentifikasi subsets pemberian isyarat *pathway* yang nampaknya akan dipengaruhi oleh perubahan di dalam profil *ekspresi* gen (Gambar 10). Dengan NDM, untuk mendapat kembali connectivas informasi tentang gen menjadi sederhana



Gambar 23 Studi Kasus Analisis Jaringan Gen

### 3.3 Kesimpulan

NDM memungkinkan ilmuwan untuk mengatur dan meneliti data sebagai penyajian grafik di dalam basisdata *Oracle 10g* NDM telah diterapkan oleh *Oracle* sebagai suatu model umum dan terbuka yang memisahkan logika aplikasi dari manajemen data. Pendekatan ini memungkinkan industri berkisar antara telekomunikasi, sistem informasi mengenai ilmu bumi, ilmu yang mempelajari hidup dan elektronika untuk mengambil keuntungan dari fungsionalitasnya.

Di paper ini, studi kasus disajikan seperti bagaimana NDM dapat digunakan untuk analisa dan penyimpanan berkenaan dengan metabolisme *pathway* dan jaringan gen. Seperti meningkatnya volume data di dalam ilmu yang mempelajari hidup yang diwakili sebagai graf, menjadi penting bahwa suatu keamanan, lingkungan yang *scalable* dan dapat dipercaya disediakan untuk data tersebut.

### Referensi

[01] Rosen Kenneth. *Discrete Mathematics and Its Applications, Fifth Edition*. McGraw Hill

[02] Rinaldi Munir. *Diktat Kuliah IF2151 Matematika Diskrit Edisi Keempat*. Program Studi Teknik Informatika ITB

[03]  
[www.oracle.com/technology/industries/life\\_sciences/pdf/ls\\_ieee\\_graph.pdf](http://www.oracle.com/technology/industries/life_sciences/pdf/ls_ieee_graph.pdf). Hak Akses 15 Desember 2006, pukul 16.00