

Aplikasi Teori Graf dalam Manajemen Sistem Basis Data Tersebar

Arifin Luthfi Putranto (13508050)

Program Studi Teknik Informatika
Institut Teknologi Bandung
Jalan Ganesha 10, Bandung
E-Mail: xenoposeidon@yahoo.com

ABSTRAK

Teori graf merupakan teori yang sejak lama telah ditemukan. Graf digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut. Banyak permasalahan yang dapat diselesaikan dengan menggunakan teori graf. Hingga kini, begitu banyak ditemukan aplikasi dari teori graf disekitar kita guna menyelesaikan masalah yang ada. Pada umumnya, teori graf digunakan untuk memodelkan suatu persoalan yang ada dan model tersebut digunakan untuk mencari solusi dari permasalahan tersebut.

Pada makalah ini, akan dibahas mengenai salah satu aplikasi teori graf dalam kehidupan, yaitu dalam manajemen sistem basis data tersebar (*distributed database*). Sistem basis data tersebar disini akan digambarkan sebagai sebuah graf dimana sisi *Client* dan *Server* digambarkan sebagai simpul (*verteks*), sedangkan kabel yang menghubungkan antara *Server* maupun *Client* yang satu dengan yang lain digambarkan sebagai sisi (*edge*).

Kata kunci: Graf, Simpul (*Verteks*), Sisi (*Edge*), Basis Data, *Client*, *Server*, *Computer Network*, *Site*.

1. PENDAHULUAN

Pada masa kini, jaringan komputer (*computer network*) pasti sudah banyak digunakan dalam berbagai aspek kehidupan di masyarakat. Setiap perusahaan maupun institusi yang ada pastilah memiliki jaringan komputer yang berguna untuk manajemen dan mendukung jalannya lalu lintas informasi data yang ada dalam perusahaan ataupun institusi tersebut.

Sistem basis data tersebar (*distributed database*) adalah sebuah kumpulan koleksi data dimana piranti-piranti penyimpanan data (*data storage devices*) tidak dikendalikan oleh CPU (*central processing unit*) yang sama. Piranti penyimpanan ini bisa saja terletak pada komputer yang berbeda pada lokasi fisik yang sama atau

bisa juga terletak di lokasi fisik yang berbeda. Jika database terletak pada lokasi fisik yang berbeda, maka komputer-komputer tersebut akan saling dihubungkan dengan jaringan komputer. Walaupun secara fisik data yang ada disimpan di tempat-tempat yang berbeda, namun secara logik basis data tersebut adalah satu buah basis data. Data-data yang tersebar dalam lokasi-lokasi tersebut dapat merupakan data yang direplikasi (data yang sama diduplikasi di lebih dari satu lokasi yang berbeda) dan difragmentasi (kelompok data yang sama dipotong-potong dan tiap potongan disimpan di lokasi yang berbeda). Jika seorang pengguna melakukan *query* ke basis data tersebut dari lokasi manapun, maka ia harus mendapatkan semua data secara lengkap tanpa perlu tahu dilokasi mana data tersebut disimpan.

Sistem basis data tersebar sangatlah berguna untuk manajemen data yang harus disimpan dalam suatu perusahaan ataupun institusi. Sistem basis data tersebar baik untuk digunakan karena jika terjadi kerusakan di salah satu server, maka tidak semua data yang ada akan hilang. Selain itu sistem basis data tersebar juga dapat memisahkan lokasi penyimpanan data dengan kategori yang berbeda pada komputer yang berbeda, sehingga lebih termanajemen dengan baik.

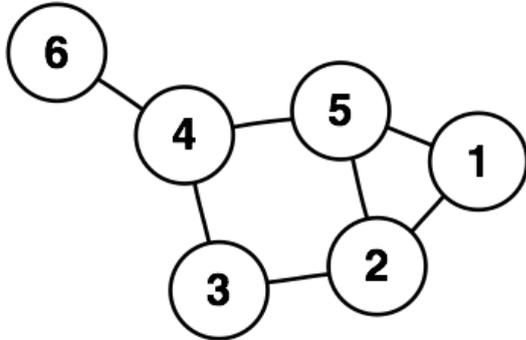
Bagaimana data-data yang tersebar di berbagai tempat itu dapat disatukan kembali? Bagaimana cara agar komputer-komputer yang menyimpan data-data yang tersebar tersebut terhubung satu sama lain? Disinilah akan dibahas salah satu aplikasi teori graf dalam kehidupan sehari-hari, khususnya untuk manajemen sistem basis data tersebar.

2. METODE

Untuk memecahkan persoalan mengenai sistem basis data tersebar, dapat digunakan teori graf. Teori graf dapat digunakan untuk memodelkan persoalan guna menghubungkan komputer-komputer yang satu dengan yang lainnya yang terpisah secara fisik. Dengan kata lain, jaringan komputer itu dapat dimodelkan sebagai suatu graf tak berarah dimana pada satu sisi data dapat berjalan bolak-balik.

2.1 Teori Graf

Secara informal, suatu graf adalah himpunan benda-benda yang disebut simpul (*vertices*) yang terhubung oleh sisi (*edge*). Biasanya graf digambarkan sebagai kumpulan titik-titik (melambangkan simpul) yang dihubungkan oleh garis-garis (melambangkan sisi).



Gambar 2.1.1. Contoh graf dengan 6 simpul dan 7 sisi.

Suatu graf G dapat dinyatakan sebagai $G = \langle V, E \rangle$. Graf G terdiri atas himpunan V yang berisikan simpul (*vertices*) pada graph tersebut dan himpunan dari E yang berisi sisi (*edge*) pada graf tersebut. Himpunan E dinyatakan sebagai pasangan dari *vertices* yang ada dalam V . Sebagai contoh definisi dari graf pada gambar diatas adalah :

$$V = \{1, 2, 3, 4, 5, 6\}$$

$$E = \{(1, 2), (1, 5), (2, 3), (3, 4), (4, 5), (5, 2), (4, 6)\}$$

Sebuah struktur graf bisa dikembangkan dengan memberi bobot pada tiap sisi. Graf berbobot dapat digunakan untuk melambangkan banyak konsep berbeda. Sebagai contoh jika suatu graf melambangkan jaringan jalan maka bobotnya bisa berarti panjang jalan maupun batas kecepatan tertinggi pada jalan tertentu.

2.1.1 Jenis-jenis Graf

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.

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

Graf yang mengandung sisi ganda atau gelang dinamakan graf tak-sederhana (*unsimple graph*).

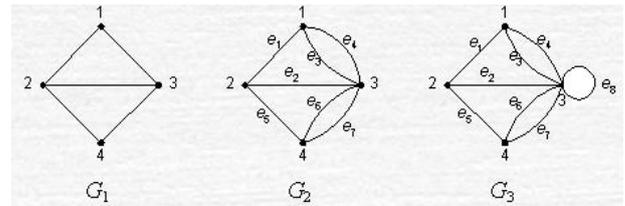
Berdasarkan orientasi arah pada sisi, maka secara umum graf dibedakan atas 2 jenis:

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

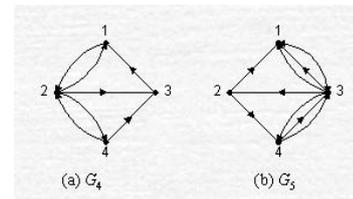
Graf yang sisinya tidak mempunyai orientasi arah disebut graf tak-berarah.

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

Graf yang setiap sisinya diberikan orientasi arah disebut sebagai graf berarah



Gambar 2.1.1.1 (G1) graf sederhana, (G2) graf ganda, dan (G3) graf semu



Gambar 2.1.1.2 (a) graf berarah, (b) graf ganda berarah

2.1.2 Terminologi Dasar Graf

Pada teori graf, banyak ditemukan terminologi (istilah) yang ada dalam persoalan. Terminologi itu antara lain ialah :

A. Bertetangga (*Adjacent*)

Istilah bertetangga merujuk pada dua buah simpul yang keduanya terhubung secara langsung oleh sebuah sisi, baik berarah maupun tidak.

B. Bersisian (*Incident*)

Untuk sembarang sisi, sisi yang dikatakan bersisian dengan simpul a dan b , ialah sisi yang secara langsung terhubung dengan kedua simpul tersebut atau dengan kata lain sisi yang menghubungkan kedua simpul

C. Simpul Terkecil (*Isolated Vertex*)

Simpul terpeleceh ialah simpul yang tidak mempunyai sisi yang bersisian dengannya. Dikatakan pula simpul terpeleceh adalah simpul yang tidak bertetangga satupun dengan simpul lainnya.

D. Graf Kosong (*Null Graph*)

Graf kosong ialah graf dengan n buah simpul yang himpunan sisinya merupakan himpunan kosong.

E. Derajat (*Degree*)

Derajat suatu simpul pada graf tak-berarah adalah jumlah sisi yang bersisian dengan simpul tersebut. Simpul yang memiliki gelang (*loop*), dihitung berderajat dua. Simpul yang berderajat satu disebut simpul anting-anting (*pendant vertex*). Pada graf berarah, derajat simpul dinyatakan dengan derajat masuk yang dihitung berdasarkan jumlah busur yang masuk pada simpul dan derajat keluar yang dihitung berdasarkan jumlah busur yang keluar dari simpul.

F. Lintasan (*Path*)

Lintasan adalah barisan berselang-seling simpul-simpul dan sisi-sisi sedemikian sehingga terbentuk lintasan dari simpul awal ke simpul yang dituju. Bila graf tersebut sederhana, maka cukup dituliskan barisan simpul-simpul yang dilaluinya saja, sebab antara dua buah simpul yang berurutan hanya terdapat satu sisi. Namun bila mengandung sisi ganda, maka perlu dituliskan lintasan antar simpul untuk menghindari kerancuan sisi yang dilewatinya.

G. Siklus (*Cycle*) atau Sirkuit (*Circuit*)

Siklus atau sirkuit ialah lintasan yang berawal dan berakhir pada simpul yang sama. Panjang sirkuit dihitung berdasarkan jumlah sisi yang terdapat pada sirkuit tersebut. Sebuah sirkuit dikatakan sirkuit sederhana (*simple sirkuit*) bila setiap sisi yang dilaluinya berbeda.

H. Terhubung (*Connected*)

Simpul dikatakan terhubung jika terdapat lintasan yang menghubungkan kedua simpul tersebut. Dengan adanya hubungan antar simpul, maka suatu simpul dapat dicapai oleh simpul yang lain. Graf dikatakan graf terhubung jika setiap pasang simpulnya terhubung. Sebagai catatan, graf yang hanya terdiri dari satu simpul saja tetap dikatakan terhubung.

I. Upagraf (*Subgraph*)

Suatu graf dikatakan upagraf atau bagian dari graf lain yang lebih besar jika himpunan simpul pada upagraf tersebut merupakan himpunan bagian dari graf yang lebih besar. Dengan kata lain, bila graf besar tersebut dipecah dengan menghapus beberapa sisinya, maka pecahan yang didapat adalah upagraf-upagraf dari graf besar.

J. Upagraf Merentang (*Spanning Upagraf*)

Upagraf merentang adalah upagraf yang menagndung semua simpul dari graf yang direntangnya dengan jumlah sisi yang lebih sedikit. Upagraf merentang bisa didapatkan dengan cara menghapus sisi namun tetap mempertahankan keterhubungan simpul-simpul pada graf.

K. Cut-Set

Cut-set dari sebuah graf ialah himpunan sisi dari graf yang bila salah satu sisi himpunan tersebut dibuang, akan membuat graf menjadi tidak terhubung. Sehingga cut-set menghasilkan dua buah komponen graf terhubung.

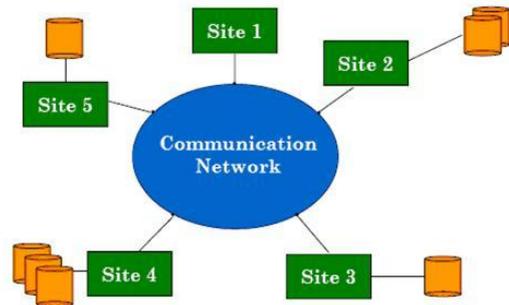
L. Graf Berbobot (*Weighted Graph*)

Graf yang setiap sisinya diberi sebuah nilai atau bobot. Bobot tersebut dapat menyatakan nilai kepentingan sisi yang diwakilinya.

2.2 Representasi Sistem Basis Data Tersebar Sebagai Graf

Dalam sebuah sistem basis data tersebar, database disimpan pada beberapa komputer. Komputer-komputer

dalam sebuah sistem terdistribusi berhubungan satu sama lain melalui bermacam-macam media komunikasi seperti high-speed buses atau telephone line.

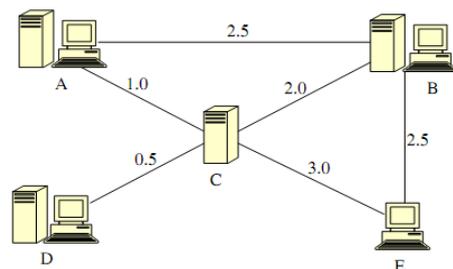


Gambar 2.2.1. Gambaran umum sistem basis data tersebar.

Sebuah sistem basis data tersebar berisikan sekumpulan lokasi (*site*), di mana tiap-tiap site dapat berpartisipasi dalam pengeksesian operasi-operasi yang mengakses data pada satu *site* atau beberapa *site*. Tiap-tiap *site* dapat memproses operasi lokal yaitu sebuah operasi yang mengakses data pada satu *site* di mana operasi telah ditentukan.

Sebuah *site* juga dapat mengambil bagian dalam mengeksekusi operasi global yaitu operasi yang mengakses data pada *site* yang berbeda di mana operasi telah ditentukan, atau operasi yang mengakses data pada beberapa *site* yang berbeda.

Jika sistem basis data tersebar direpresentasikan secara logik sebagai graf, maka simpul (*verteks*) merepresentasikan lokasi (*site*) unit penyimpanan / pemrosesan data. Sedangkan sisi (*edge*) merepresentasikan hubungan / koneksi antara dua buah lokasi. Pada setiap lokasi, dimungkinkan ada sebuah *server* sebagai unit penyimpanan dan pemrosesan data dan sebuah *client*, yaitu aplikasi yang melakukan operasi terhadap data-data yang disimpan. Suatu lokasi bisa saja terdiri dari sebuah *server* dan sebuah *client*, tapi bisa juga hanya terdiri atas sebuah *client* saja ataupun *server* saja. Jaringan yang menghubungkan satu lokasi dengan lokasi lain memiliki laju data transmisi (*bandwith*) yang menyatakan jumlah maksimum data yang bisa dilewatkan dalam jaringan (baik untuk proses pengambilan data (*download*) maupun pengiriman data (*upload*)).



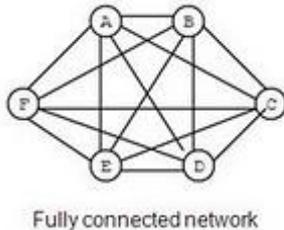
Gambar 2.2.2. Contoh graf dengan yang menggambarkan sistem basis data tersebar.

Pada gambar diatas, ada 5 lokasi yaitu A, B, C, D dan E. Semua lokasi memiliki sebuah *server* dan sebuah *client*. Kecuali lokasi C yang hanya memiliki sebuah *server* dan lokasi E yang hanya memiliki sebuah *client*. Data disimpan pada lokasi yang memiliki *server*, sedangkan untuk melakukan update dan manipulasi data dapat dilakukan di lokasi yang memiliki *client*.

Site-site dalam database terdistribusi dihubungkan secara fisik dengan berbagai cara. Beberapa topologi digambarkan sebagai sebuah graf yang simpul-simpulnya bersesuaian dengan site. Sebuah edge dari simpul A ke simpul B bersesuaian dengan sebuah hubungan langsung antara dua site. Beberapa konfigurasi (bentuk) digambarkan sebagai berikut:

Fully Connected Network :

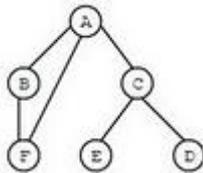
Keuntungan : kalau salah satu node rusak, yang lainnya masih dapat berjalan (tetapi biaya mahal).
Kerugian : control management tidak terjamin.



Gambar 2.2.3. Topologi Jaringan Fully Connected Network

Partially Connected Network :

Keuntungan : reliability rendah, biaya dapat ditekan
Kerugian : control management tidak terjamin

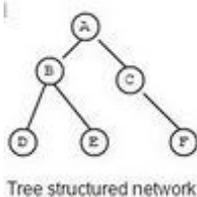


Gambar 2.2.4. Topologi Jaringan Partially Connected Network

Tree Structure Network :

Keuntungan : bersifat sentral, control management lebih terjamin
Kerugian : kalau node pusat (A) rusak, semua akan rusak.

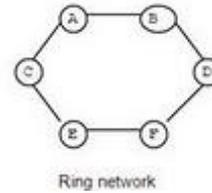
Catatan : setiap proses dimulai dari bawah.



Gambar 2.2.5. Topologi Jaringan Tree Structured Network

Ring Network (LAN) :

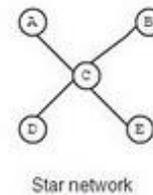
Keuntungan : rusak satu, yang lain masih berjalan
Kerugian : kontrol management kurang terjamin karena bersifat desentralisasi



Gambar 2.2.6. Topologi Jaringan Ring Network

Star Network (LAN) :

Keuntungan : control management lebih terjamin, karena bersifat sentral - reliability rendah
Kerugian : kalau pusat rusak, yang lainnya rusak



Gambar 2.2.7. Topologi Jaringan Star Network

Pada setiap sisi yang menghubungkan antara satu lokasi dengan lokasi yang lainnya terdapat suatu nilai yang merepresentasikan laju data transmisi yang disebut *bandwith*. Dikarenakan *bandwith* yang berbeda-beda, maka data yang lewat antara satu jalur dengan jalur yang lain akan memiliki kecepatan transfer yang berbeda. Perlu dilakukan optimasi disini. Kita dapat melakukan optimasi kecepatan transfer file dengan menggunakan algoritma yang digunakan pada teori graf, algoritma Dijkstra.

2.3 Algoritma Dijkstra untuk Optimasi Kecepatan Transfer Data

Algoritma Dijkstra, dinamai menurut penemunya, Edsger Dijkstra, adalah sebuah algoritma rakus (*greedy algorithm*) dalam memecahkan permasalahan jarak terpendek (*shortest path problem*) untuk sebuah graf dengan bobot-bobot sisi (*edge weights*) yang bernilai tak-negatif. Input algoritma ini adalah sebuah graf berarah yang berbobot (*weighted directed graph*) G dan sebuah sumber vertex s dalam G dan V adalah himpunan semua vertices dalam graph G.

Algoritma Dijkstra dapat digunakan untuk melakukan optimasi kecepatan transfer data pada sistem basis data tersebar karena sistem basis data tersebar dapat dimodelkan dengan teori graf. Dalam hal ini, bobot dari sisi graf menyatakan *bandwith* dari kabel yang menghubungkan satu lokasi dengan lokasi lainnya.

Setiap sisi dari graf ini adalah pasangan vertices (u,v) yang melambangkan hubungan dari simpul u ke simpul v.

Himpunan semua sisi disebut E . Bobot (*weights*) dari semua sisi dihitung dengan fungsi :

$$w: E \rightarrow [0, \infty)$$

Jadi $w(u,v)$ adalah jarak tak-negatif dari simpul u ke simpul v .

Ongkos (*cost*) dari sebuah sisi dapat dianggap sebagai jarak antara dua simpul, yaitu jumlah jarak semua sisi dalam jalur tersebut. Untuk sepasang simpul s dan t dalam V , algoritma ini menghitung jarak terpendek dari s ke t .

```
1  function Dijkstra(G, w, s)
2      for each vertex v in V[G]
3          d[v] := infinity
4          previous[v] := undefined
5      d[s] := 0
6      S := empty set
7      Q := V[G]
8      while Q is not an empty set
9          u := Extract_Min(Q)
10         S := S union {u}
11         for each edge (u,v) outgoing
12             from u
13                 if d[u] + w(u,v) < d[v]
14                     d[v] := d[u] + w(u,v)
15                     previous[v] := u
```

Dengan digunakannya algoritma Dijkstra diatas, dapat dicari jalur tercepat yang dilalui data dalam sistem basis data tersebar. Jalur tersebut dapat digunakan agar kecepatan transfer data menjadi optimal.

2.4 Keuntungan dan Kerugian Penggunaan Sistem Basis Data Tersebar

Setiap pemodelan masalah, pastilah terdapat keuntungan serta kerugiannya masing-masing. Seperti pada pemodelan sistem basis data tersebar, terdapat berbagai keuntungan dan kerugiannya. Berikut keuntungan dan kerugian yang diperoleh dari pemodelan manajemen sistem basis tersebar dengan teori graf :

Keuntungan :

1. Pengawasan distribusi dan pengambilan data

Jika sejumlah *site* yang berbeda dihubungkan satu sama lain, lalu seorang pemakai yang berada pada satu *site* dapat mengakses data yang tersedia pada *site* lain. Sebagai contoh, sistem distribusi pada sebuah bank memungkinkan seorang pemakai pada salah satu cabang dapat mengakses data cabang lain.

2. Reliability dan availability

Sistem distribusi dapat terus menerus berfungsi dalam menghadapi kegagalan dari *site* individu atau mata rantai komunikasi antar *site*. Misalnya jika *site-site* gagal dalam sebuah sistem distribusi, *site-site* lainnya dapat melanjutkan operasi jika data telah direplikasi pada beberapa *site*.

3. Kecepatan pemrosesan query

Jika sebuah query melibatkan data pada beberapa *site*, memungkinkan membagi query ke dalam sub query yang dapat dieksekusi dalam bentuk paralel oleh beberapa *site*. Perhitungan secara paralel mempercepat pemrosesan dari seorang pemakai query

4. Otonomi lokal

Pendistribusian sistem mengizinkan sekelompok individu dalam sebuah perusahaan untuk melatih pengawasan lokal melalui data mereka sendiri. Dengan kemampuan ini dapat mengurangi ketergantungan pada pusat pemrosesan.

5. Efisien dan fleksibel

Data dalam sistem distribusi dapat disimpan dekat dengan titik di mana data tersebut dipergunakan. Data dapat secara dinamik bergerak atau disalin, atau salinannya dapat dihapus.

Kerugian :

1. Harga software yang mahal

Hal ini disebabkan sangat sulit untuk membuat sistem database distribusi.

2. Kemungkinan kesalahan lebih besar

Site-site yang termasuk dalam sistem distribusi beroperasi secara paralel sehingga menjadi lebih sulit untuk menjamin kebenaran dari algoritma. Adanya kesalahan mungkin tak dapat diketahui.

3. Biaya pemrosesan tinggi

Perubahan pesan-pesan dan penambahan perhitungan dibutuhkan untuk mencapai koordinasi antar *site*. Dalam memilih sebuah disain untuk sistem database, perancang harus mengimbangi keuntungan dan kerugian dari database terdistribusi.

IV. KESIMPULAN

Teori graf dapat dijadikan model permasalahan di berbagai bidang karena memiliki ruang lingkup aplikasi yang sangat luas. Salah satu contoh pengaplikasian teori graf ialah pada manajemen sistem basis data tersebar.

Pemodelan sistem basis data tersebar dengan teori graf dapat menyederhanakan cara menganalisis masalah yang kemungkinan timbul. Selain itu, pemodelan dengan teori graf dapat dibuat menjadi berbagai topologi yang masing-masing mempunyai keuntungan dan kerugian tersendiri. Pilihan kita akan menggunakan yang mana tergantung akan kebutuhan.

REFERENSI

- [1] http://id.wikipedia.org/wiki/Teori_graf, waktu akses 20 Desember 2009.

- [2] http://id.wikipedia.org/wiki/Algoritma_Dijkstra,
waktu akses 20 Desember 2009.
- [3] <http://viliaekameyana.blogspot.com/2008/11/basis-data-terdistribusi.html>,
waktu akses 20 Desember 2009.
- [4] <http://kuliah.itb.ac.id/mod/resource/view.php?id=5211>,
waktu akses 20 Desember 2009.
- [5] Munir, Rinaldi, Diktat Kuliah Struktur Diskrit,
Departemen Teknik Informatika, Institut Teknologi Bandung.