Mobile Application

Catatan Kuliah #7

Alauddin Maulana Hirzan, M. Kom
0607069401

Graph Database

Graph Database

Apa itu **Graph Database**? #1

Graph Database adalah jenis basis data yang dirancang untuk menyimpan dan mengelola data sebagai graf, yang terdiri dari simpul (node) dan sisi (edge). Dalam basis data graf, data disimpan sebagai sekumpulan node yang saling terhubung, di mana setiap node mewakili entitas dan setiap sisi mewakili hubungan antar entitas.

Informasi

Contoh penggunaannya nyata dari **Graph Database** seperti aplikasi Sosial Media

Graph Database

Apa itu **Graph Database**? #2

Keuntungan utama dari basis data grafik adalah kemampuannya untuk merepresentasikan hubungan yang kompleks antara titik-titik data secara efisien.

Sebagai contoh, jaringan sosial dapat direpresentasikan sebagai sebuah graf, di mana setiap pengguna adalah simpul dan setiap hubungan antar pengguna adalah sisi. Hal ini memungkinkan kueri yang kompleks dilakukan dengan cepat dan efisien, seperti menemukan semua teman dari pengguna tertentu, atau mengidentifikasi kelompok pengguna dengan minat yang sama.

Graph Database

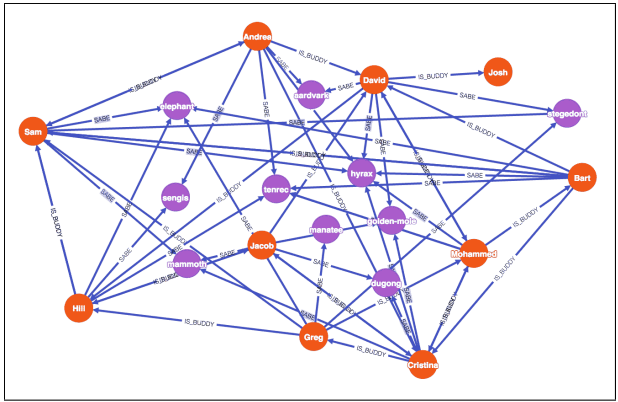
Apa itu **Graph Database**? #3

Pemanfaatan

Basis data grafik sering digunakan dalam aplikasi seperti jejaring sosial, mesin rekomendasi, dan sistem deteksi penipuan. Database ini juga dapat digunakan dalam aplikasi yang membutuhkan pemodelan data yang kompleks, seperti manajemen rantai pasokan, manajemen jaringan, dan bioinformatika.

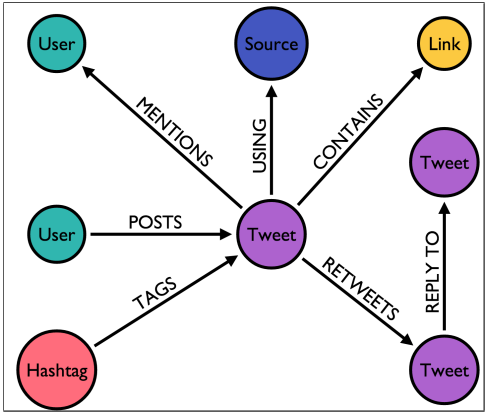
Graph Database

Ilustrasi **Graph Database** #1



Graph Database

Ilustrasi **Graph Database** #2



Graph Database

Komponen **Graph Database** #1

Graph Database memiliki perbedaan bentuk dalam hal menyimpan data, lebih sederhana namun dapat menciptakan jaringan data yang luar biasa kompleks

Secara sederhana, **Graph Database** hanya memiliki 4 komponen utama:

1. Nodes / Objek
2. Edges / Relasi
3. Labels / Penanda
4. Properties / Isi Data

Graph Database

Komponen **Graph Database** #2

Secara keseluruhan, **Graph Database** memiliki 3 tambahan komponen lainnya berupa:

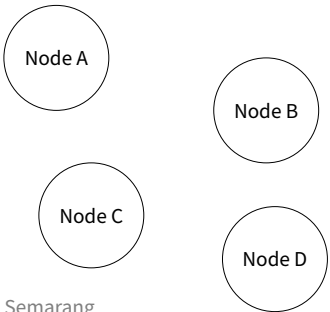
1. Indeks
2. Traversal algorithm / Algoritma Jelajah
3. Query language / Bahasa Kueri

Sehingga data dapat diambil secara cepat tanpa mengganggu kinerja sistem database

Graph Database

Komponen **Graph Database** - Node

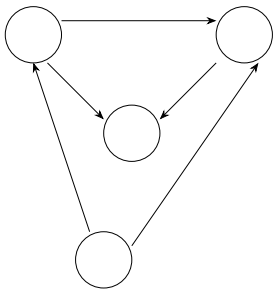
Node → blok bangunan dasar dari basis data graf. Setiap node merepresentasikan sebuah entitas, seperti orang, tempat, atau benda. Node dapat berisi properti yang mendeskripsikan entitas, seperti nama, usia, atau alamat.



Graph Database

Komponen **Graph Database** - Edge

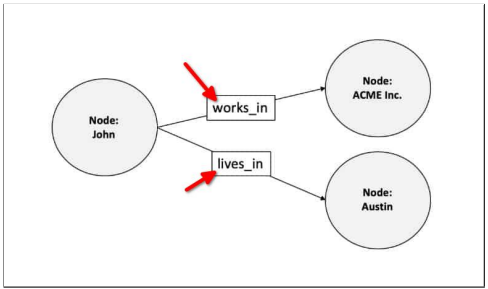
Edge $\rightarrow$ hubungan antara node. Edge mewakili hubungan antara entitas, seperti "berteman dengan", "berlokasi di", atau "dibeli oleh". Edge juga dapat berisi properti yang menggambarkan hubungan, seperti tanggal pertemanan atau harga pembelian.



Graph Database

Komponen **Graph Database** - Label

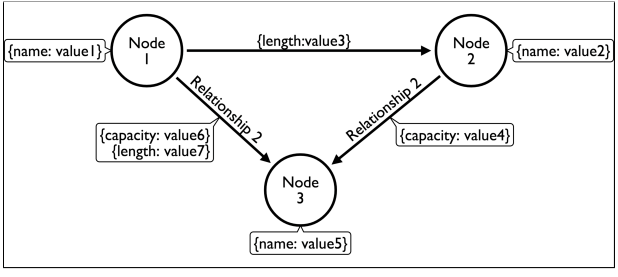
Label digunakan untuk mengelompokkan node ke dalam set berdasarkan karakteristiknya. Sebagai contoh, node yang mewakili orang dapat diberi label "orang", sedangkan node yang mewakili perusahaan dapat diberi label "perusahaan". Label juga digunakan untuk mendefinisikan indeks dan batasan pada data.



Graph Database

Komponen **Graph Database** - Properti

Properti adalah pasangan kunci-nilai yang mendeskripsikan simpul dan tepi. Properti dapat digunakan untuk menyimpan semua jenis data, seperti teks, angka, tanggal, atau larik. Properti juga dapat diindeks untuk pencarian dan pengambilan yang lebih cepat.



Graph Database

Komponen **Graph Database** - Traversal Algorithm #1

Algoritma penjelajahan digunakan untuk menavigasi basis data graf dengan mengikuti sisi-sisi dari satu simpul ke simpul lainnya. Algoritme ini dapat digunakan untuk menemukan jalur antar simpul, menghitung jalur terpendek, atau mengidentifikasi kelompok simpul dengan properti yang sama.

Graph Database

Komponen **Graph Database** - Traversal Algorithm #2

Dalam melakukan pencarian, terdapat banyak cara untuk menjelajahi database seperti:

1. **Depth-First Search**
2. **Breadth-First Search**
3. **Djikstra's Algorithm**
4. **A* Search**
5. **Minimum Spanning Tree**
6. **Page Rank**
7. **Connected Component Labeling**

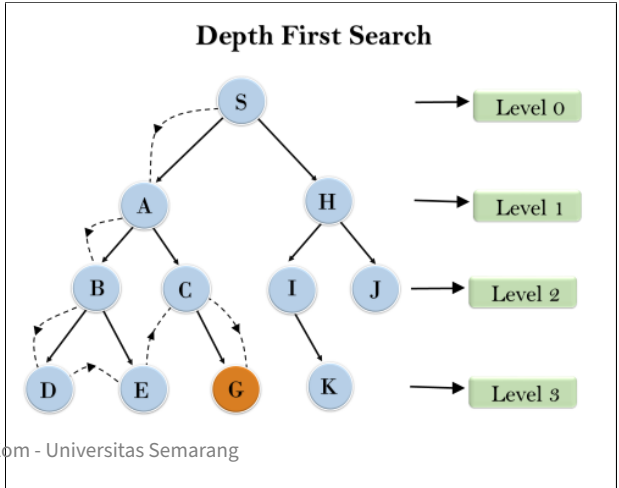
Graph Database

Komponen **Graph Database** - Traversal Algorithm - Depth-First Search #1

DFS dimulai dari sebuah simpul tertentu dan menjelajahi sejauh mungkin di sepanjang setiap cabang sebelum melakukan backtracking. Algoritma ini sering digunakan untuk menemukan jalur antar simpul dalam sebuah graf.

Graph Database

Komponen **Graph Database** - Traversal Algorithm - Depth-First Search #2



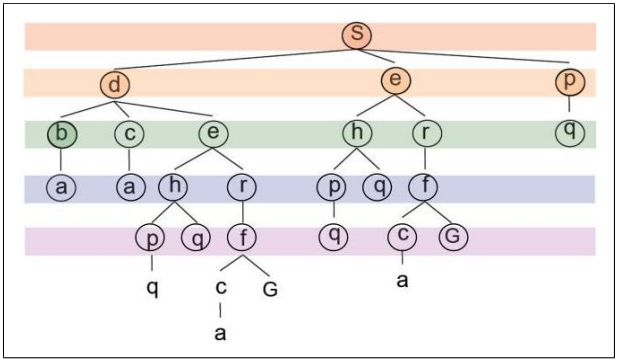
Graph Database

Komponen **Graph Database** - Traversal Algorithm - Breadth-First Search #1

BFS menjelajahi graf dengan mengunjungi semua node pada jarak tertentu dari node awal sebelum pindah ke node yang lebih jauh. Algoritma ini sering digunakan untuk menemukan jalur terpendek antara dua node.

Graph Database

Komponen **Graph Database** - Traversal Algorithm - Breadth-First Search #2



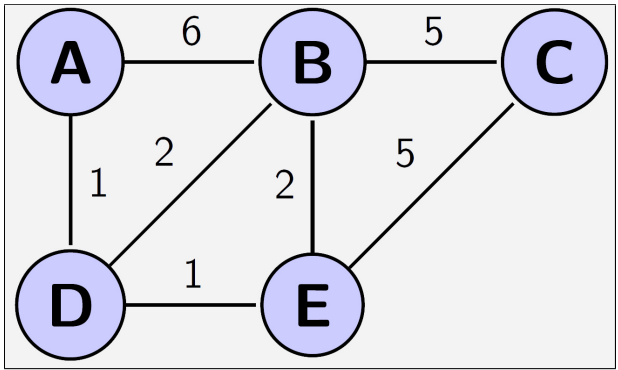
Graph Database

Komponen **Graph Database** - Traversal Algorithm - Djikstra's Algorithm #1

Algoritma Dijkstra digunakan untuk menemukan jalur terpendek antara dua node dalam graf berbobot. Algoritma ini memberikan jarak sementara untuk setiap node, kemudian memperbarui jarak berdasarkan bobot sisi-sisi di antara node.

Graph Database

Komponen **Graph Database** - Traversal Algorithm - Djikstra's Algorithm #2



Graph Database

Komponen **Graph Database** - Query Language

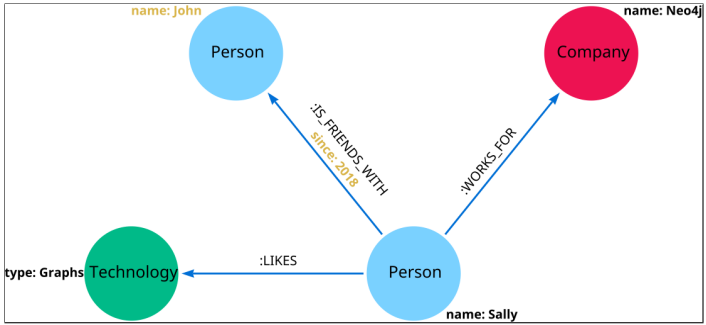
Sebuah basis data grafik biasanya memiliki bahasa kueri sendiri, yang digunakan untuk berinteraksi dengan data. Beberapa bahasa kueri grafik yang populer termasuk Cypher (digunakan oleh Neo4j), Gremlin (digunakan oleh Apache TinkerPop), dan SPARQL (digunakan oleh basis data RDF).

Contoh (Cypher Neo4j):

- CREATE (p:Person)-[:LIKES]->(t:Technology)
- MATCH (p:Person)<-[:LIKES]-(t:Technology)
- MATCH (p:Person)-[:LIKES]-(t:Technology)

Graph Database

Ilustrasi Database Neo4J



Sehingga pola untuk **Sally** adalah

(p:Person name: "Sally")-[rel:LIKES]->(g:Technology type: "Graphs")

Graph Database

Contoh Query Cypher #1

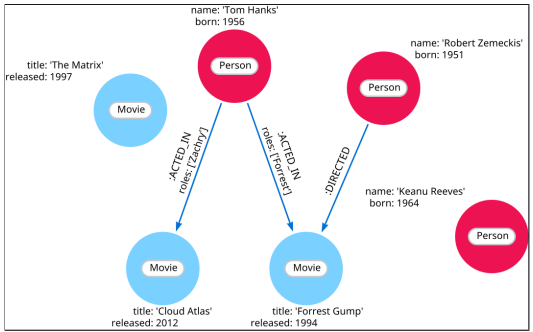
Masukkan Query ke Graph Neo4J:

```
CREATE (matrix:Movie title: 'The Matrix', released: 1997)
CREATE (cloudAtlas:Movie title: 'Cloud Atlas', released: 2012)
CREATE (forrestGump:Movie title: 'Forrest Gump', released: 1994)
CREATE (keanu:Person name: 'Keanu Reeves', born: 1964)
CREATE (robert:Person name: 'Robert Zemeckis', born: 1951)
CREATE (tom:Person name: 'Tom Hanks', born: 1956)
CREATE (tom)-[:ACTED_IN roles: ['Forrest']]->(forrestGump)
CREATE (tom)-[:ACTED_IN roles: ['Zachry']]->(cloudAtlas)
CREATE (robert)-[:DIRECTED]->(forrestGump)
```

Graph Database

Contoh Query Cypher #2

Akan menghasilkan **Graph Database** sebagai berikut:




THANK YOU