**Implementasi VoG (Vocabulary based summarization of Graph) Pada Web Graph**

***Implementation of VoG (Vocabulary based summarization of Graph) on the Web Graph***

**Tugas Akhir**

**Kelompok Keahlian : SIDE**

**Satrio Adityo Hartomo**

**1103120029**

****

**Program Studi Teknik Informatika**

**Fakultas Informatika**

**Universitas Telkom**

**Bandung**

**2015**

# LEMBAR PERNYATAAN

Dengan ini saya menyatakan bahwa tugas akhir dengan judul **“Implementasi VoG (Vocabulary based summarization of Graph) pada Web Graph”** dan seluruh isinya benar-benar merupakan karya saya sendiri dan saya tidak melakukan penjiplakan atau pengutipan dengan cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan. Atas pernyataan ini, saya siap menanggung resiko/sanksi yang dijatuhkan kepada saya apabila kemudian ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya ini, atau ada klaim dari pihak lain terhadap karya keaslian saya ini.

Bandung, Desember 2015

Yang membuat pernyataan,

Satrio Adityo Hartomo

# LEMBAR PENGESAHAN

**Implementasi VoG (Vocabulary based summarization of Graph) pada Web Graph**

***Implementation of VoG (Vocabulary based summarization of Graph) on the Web Graph***

**Satrio Adityo Hartomo**

**1103120029**

Telah disetujui dan disahkan sebagai tugas akhir

Program Studi Teknik Informatika

Fakultas Informatika Universitas Telkom

Bandung, Desember 2015

Menyetujui

|  |  |  |
| --- | --- | --- |
| Pembimbing I |  | Pembimbing II |
|  |  |  |
|  |  |  |
| Kemas Rahmat Saleh W, S.T., M.Eng. NIP. 06830335-1 |  | Siti Sa'adah, S.T., M.T.  NIP. 13861143-1 |

|  |
| --- |
| Ketua Program Studi  Sarjana Teknik Informatika |
|  |
|  |
| Kemas Rahmat Saleh W, S.T., M.Eng. NIP. 06830335-1 |

# Abstrak

# Abstract

# Lembar Persembahan

# Kata Pengantar

# Daftar Isi

# Daftar Gambar

# Daftar Tabel

# Daftar Istilah

# Pendahuluan

## Latar Belakang

Dengan adanya teknik penyimpanan data yang baru yaitu *graph database* [1], cara untuk menyimpan data tidak lagi terbatas pada *relational database* yang struktural. Data yang mempunyai perbedaan atribut (semi terstruktur) dapat disimpan ke database dengan representasi graph [1], seperti Facebook dan Twitter yang juga menggunakan *graph database* untuk menyimpan data-datanya [2] [3]. *Website* yang lain pun memiliki data yang unik, berjumlah banyak dan semakin bertambah hingga saat ini [4]. *Website* yang dulunya hanya menampilkan data statis kini telah berevolusi menjadi *website* yang dapat menampilkan data secara dinamis dan dapat direpresentasikan sebagai graph (web graph) [1].

Diberikan web graph yang besar yang semakin bertambah jumlahnya, dengan ribuan *node* dan *edge* [5], lalu apa yang dapat dikatakan mengenai struktur web graph yang besar tersebut? Apakah strukturnya *random*? Jika tidak *random*, bagaimana kita dapat meringkas web graph tersebut untuk mendapatkan informasi dari strukturnya? [6]

VoG (Vocabulary based summarization of Graph) adalah metode untuk meringkas graph yang besar secara efisien [6]. Dalam tugas akhir ini akan dilakukan implementasi web graph *summarization* menggunakan metode VoG untuk meringkas web graph menjadi subgraph-subgraph yang lebih kecil ukuran atau jumlah nodenya dan mendapatkan informasi dari subgraph-subgraph yang dihasilkan.

## Perumusan Masalah

Dari latar belakang yang telah diuraikan, permasalahan yang akan dibahas dalam tugas akhir ini adalah sebagai berikut:

1. Bagaimana mengimplementasikan VoG untuk *summarize* web graph?
2. Bagaimana menganalisis struktur subgraph yang dihasilkan oleh VoG untuk memperoleh minimum satu kalimat informasi?

## Tujuan

Adapun tujuan dalam tugas akhir ini adalah sebagai berikut:

1. Mengimplementasikan metode VoG untuk men-*summarize* web graph.
2. Menganalisis struktur subgraph yang dihasilkan VoG untuk memperoleh minimum satu kalimat informasi.

## Batasan Masalah

Mengenai batasan permasalahan dalam tugas akhir ini adalah sebagai berikut:

1. Struktur yang dapat diidentifikasi oleh metode VoG dalam tugas akhir ini terdiri dari *Perfect Clique, Perfect Star, Perfect Bipartite, Perfect Chain, Near Clique, Near Star, Near Bipartite,* dan *Near Chain.*
2. Metode VoG hanya dapat melakukan identifikasi struktur setelah memiliki keluaran dari algoritma *graph decomposition* yang dalam tugas akhir ini adalah algoritma Slashburn.
3. Menggunakan dataset dari hasil crawling web menggunakan aplikasi buatan sendiri.

## Metodologi Penyelesaian Masalah

Metode penyelesaian masalah yang diterapkan dalam tugas akhir ini adalah sebagai berikut:

1. Identifikasi permasalahan

Pada tahap ini dilakukan studi literatur dan diskusi untuk mengidentifikasi permasalahan yang akan dibahas pada tugas akhir ini. Identifikasi meliputi observasi terkait fenomena yang terjadi di bidang graph database, metode yang digunakan untuk solusi permasalahan, dan batasan masalah yang ada.

1. Studi literatur

Pada tahap ini dilakukan pengumpulan dan pemahaman terkait materi-materi yang dibutuhkan untuk menyelesaikan permasalahan yang tertera di perumusan masalah. Referensi yang dicari antara lain materi mengenai metode VoG, metode *graph decomposition* untuk menghasilkan graph summarization.

1. Perancangan dan pembuatan sistem

Pada tahap ini dilakukan pemodelan dataset ke dalam bentuk graph. Kemudian membuat rancangan sistem untuk memberikan solusi dari permasalahan. Dari rancangan sistem yang sudah dibuat dilakukan pembuatan sistem dengan menggunakan bahasa pemrograman Java.

1. Pengujian dan analisis

Pada tahap ini dilakukan pengujian terhadap dataset yang sudah dimodelkan dalam bentuk graph. Pengujian dilakukan sesuai dengan skenario yang ada pada bagian strategi pengujian. Kemudian dilakukan analisis terhadap subgraph-subgraph hasil ringkasan graph.

1. Penyusunan laporan

Pada tahap ini dilakukan dokumentasi dan pelaporan hasil sesuai dengan aturan dan sistematika penulisan yang ditetapkan institusi.

## Sistematika Penulisan

Sistematika penulisan buku tugas akhir ini adalah sebagai berikut:

* 1. Bab I Pendahuluan

Bab ini berisi latar belakang, perumusan masalah, tujuan, batasan masalah, metodologi penyelesaian masalah dan sistematika penulisan buku tugas akhir.

* 1. Bab II Landasan Teori

Bab ini berisi landasan teori yang menjadi dasar dilakukannya penelitian pada tugas akhir ini.

* 1. Bab III Perancangan Sistem

Bab ini berisi perancangan sistem yang akan diimplementasikan pada tugas akhir ini.

* 1. Bab IV Pengujian dan Analisis

Bab ini berisi pengujian dan analisis hasil dari sistem yang telah dibuat.

* 1. Bab V Kesimpulan dan Saran

Bab ini berisi kesimpulan yang didapat dari permasalahan, pengujian dan analisis hasil, serta saran untuk penelitian selanjutnya.

# Landasan Teori

## 2.1 Teori Graph

*Graph*, secara definisi adalah sekumpulan *nodes* () dan *edges* (). Sekumpulan *nodes* bisa saja tidak terbatas atau bisa disebut ***infinite graph***, sedangkan yang sekumpulan *nodes*nya terbatas disebut ***finite graph***. Dalam tugas akhir ini hanya akan fokus pada *finite graph*. *Graph* dapat dinotasikan sebagai . adalah sekumpulan *nodes* dan adalah sekumpulan *edges* yang menghubungkan *vertices* [7].

|  |  |  |  |
| --- | --- | --- | --- |
| Gambar 2‑1: Directed Graph | | Gambar 2‑2: Undirected Graph | |
| Gambar 2‑3: Simple Graph | Gambar 2‑4: Multigraph | | Gambar 2‑5: Pseudograph |
| Gambar 2‑6: Graph Berlabel | | Gambar 2‑7: Graph Tak Berlabel | |

Perbedaan jenis *graph* dapat dilihat berdasarkan ada atau tidaknya arah *edge*, *graph* jenis ini terdiri dari ***directed graph*** dan ***undirected graph***. Berdasarkan *edge* yang menghubungkan *nodes* terdiri dari ***simple graph*** (tidak memiliki loop dan multiple edge), ***multigraph*** (tidak memiliki loop edge tetapi memiliki multiple edge) dan ***pseudograph*** (memiliki loop edge). Berdasarkan ada atau tidaknya label terdiri dari ***graph* berlabel** dan ***graph* tak berlabel**. Contoh jenis-jenis *graph* dapat dilihat pada gambar 2-1 sampai 2-7 di bawah ini [7].

Sekumpulan *nodes* dan *edges* yang ada di *graph* dapat digunakan untuk memodelkan permasalahan yang ada pada dunia nyata, sebagai contoh *graph* jaringan komputer. Namun, dalam kondisi tertentu untuk menyelesaikan permasalahan yang ada, tidak seluruh *nodes* digunakan. Sehingga yang digunakan untuk menyelesaikan masalah adalah bagian kecil dari *graph*, atau bisa disebut dengan *subgraph*. Secara definisi, *subgraph* dari *graph* adalah *graph* dimana dan dan [7].

|  |  |
| --- | --- |
| Gambar 2‑8: Graph G | Gambar 2‑9: Graph H (subgraph G) |

Cara merepresentasikan *graph* ada beberapa cara, diantaranya yaitu menggunakan *Adjacency Matrix* dan *Adjacency List*. Berikut contoh cara merepresentasikan graph [7]:

Dari gambar 2-8 dapat direpresentasikan *adjacency list* seperti gambar 2-10 di bawah ini :



Gambar 2‑10 Representasi graph - adjacency list [8]

Atau sebagai *adjacency matrix* seperti di bawah ini :

Angka 1 merepresentasikan adanya *edge* antar*nodes* [7].

## 2.2 Graph Database

*Graph Database Management System* adalah *database management system* yang memiliki metode CRUD (*Create*, *Read*, *Update*, dan *Delete*) untuk memaparkan model data *graph*. Properti model *graph* dapat dideskripsikan sebagai berikut :

* Terdiri dari *nodes* dan *relationship/edges*.
* Setiap *nodes* terdiri dari pasangan *key-value*.
* *Edge* mempunyai nama dan arah, jika tidak memiliki arah disebut *undirected graph.*
* *Edge* dapat juga mempunyai properti.

Dengan abstraksi sederhana dari *nodes* dan *edges* kedalam struktur yang saling terhubung, *graph database* memungkinkan untuk dibangunnya model yang mirip dengan permasalahan yang ada di dunia nyata [1].



Gambar 2.2‑1 Model graph database

## 2.3 Web Graph

## 2.4 Graph Decomposition

## 2.5 VoG

# Daftar Pustaka

# References

|  |  |
| --- | --- |
| [1] | I. Robinson, J. Webber dan E. Eifrem, Graph Databases, Sebastopol: O'Reilly Media, Inc, 2013. |
| [2] | P. Gupta, A. Goel, J. Lin, A. Sharma, D. Wang dan R. Zadeh, “WTF: The Who to Follow Service at Twitter,” dalam *International World Wide Web Conference Committee (IW3C2)*, Rio de Janeiro, 2013. |
| [3] | M. Curtiss, I. Becker, T. Bosman, S. Doroshenko, L. Grijincu, T. Jackson, S. Kunnatur, S. Lassen, P. Pronin, S. Sankar, G. Shen, G. Woss, C. Yang dan N. Zhang, “Unicorn: A System for Searching the Social Graph,” dalam *Proceedings of the VLDB Endowment*, Trento, 2013. |
| [4] | B. H. Murray dan A. Moore, “Sizing The Internet,” Cyveillance, 2000. |
| [5] | N. Shervashidze, S. Vishwanathan, T. H. Petri, K. Mehlhorn dan K. M. Borgwardt, “Efficient graphlet kernels for large graph comparison,” dalam *International Conference on Artificial Intelligence and Statistics (AISTATS)*, Florida, 2009. |
| [6] | D. Koutra, U. Kang, J. Vreeken dan C. Faloutsos, “VoG : Summarizing and Understanding Large Graphs,” dalam *Proceedings of the 2014 SIAM International Conference on Data Mining*, 2014. |
| [7] | K. H. Rosen, Discrete Mathematics and Its Applications, New York: The McGraw-Hill Companies, Inc., 2012. |
| [8] | M. A. Kolosovskiy, Data structure for representing a graph:, Altai State Technical University, Russia. |