

Analisa dan Implementasi *Graph Summarization* dengan Metode CANAL

Wisnu Riyan Pratama Putra¹, Kemas Rahmat S. W. S.T., M.Eng.², Alfian Akbar Ghazali S.T., M.T.³
^{1,2,3}Prodi S1 Teknik Informatika,

Fakultas Teknik,
 Telkom University, Indonesia

¹riyan.pratama@outlook.com, ²bagindokemas@telkomuniversity.ac.id, ³alfian@tass.telkomuniversity.ac.id

Abstract— Pemodelan data menggunakan *graph* telah diterapkan oleh banyak aplikasi dan sistem berskala besar dalam berbagai bidang. Data tersebut direpresentasikan sebagai *graph* dengan *node* yang mewakili sebuah objek dan *edge* menandakan hubungan antara dua objek. Untuk memahami karakteristik *graph*, maka dibutuhkan teknik *graph summarization*.

Pada penelitian ini digunakan metode CANAL (*Categorization of Attributes with Numerical Values based on Attribute Values and Link Structures of Nodes*) untuk meringkas *graph*. Metode ini merupakan pengembangan dari metode *Aggregation-Based Graph Summarization* yang melakukan peringkasan dengan mengelompokkan serta menggabungkan *node* kedalam sebuah super *node* kemudian menggali pengetahuan dari data untuk menemukan *cutoff* yang digunakan dalam pengelompokan *node* secara otomatis. Metode CANAL memperbaiki metode *graph summarization* SNAP dan k-SNAP yang masih mempunyai kelemahan dalam menangani data dengan atribut numerik[2]. Kedua metode tersebut hanya dapat menangani *categorical node attribute*, sehingga ketika dihadapkan dengan atribut numerik pengguna masih harus melakukan pengelompokan secara manual berdasarkan pengetahuan mereka terhadap data yang digunakan.

Hasil dari sistem yang akan dibangun merupakan sebuah *graph summary* yang merepresentasikan *pattern* hubungan antar kelompok dalam ringkasan. *Pattern* tersebut dapat digunakan untuk membantu memahami informasi yang tersembunyi didalam *graph* asli. Dari *summary* yang dihasilkan oleh metode CANAL kemudian dinilai kualitasnya dan dibandingkan dengan kualitas *summary* dengan *cutoff* manual. Perbandingan tersebut menunjukkan bahwa kualitas *summary* dari CANAL memiliki kualitas baik yang setara dengan kualitas *summary* dengan *cutoff* manual.

Keywords—*graph summarization, Aggregation-Based Graph Summarization, node attribute, link structure, interestingness measure.*

I. PENDAHULUAN

Perkembangan informasi telah mengalami peningkatan yang signifikan. Hal tersebut berdampak pada peningkatan jumlah data yang dihasilkan oleh berbagai sistem dan aplikasi. Untuk itu penggunaan *graph* sebagai pemodelan data telah dilakukan oleh banyak pihak dalam menangani data dengan jumlah yang sangat besar[3]. Data tersebut dimodelkan dalam *graph* dengan *node* yang merepresentasikan objek dan *edge* merepresentasikan hubungan antar objek. Oleh karena itu, untuk mengetahui karakteristik dari *graph* dan mendapatkan informasi berharga yang tersembunyi di dalam data tersebut dibutuhkan teknik *graph summarization* yang efektif.

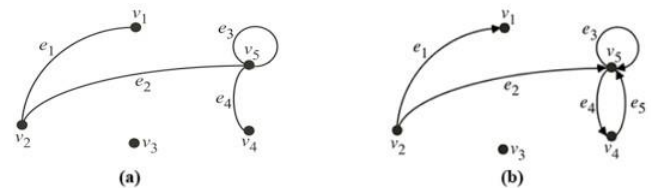
Salah satu teknik *graph summarization* yang ada adalah metode SNAP (*Summarization by Grouping Nodes on Attributes and Pairwise Relationships*) dan k-SNAP. Kedua metode tersebut masih terdapat kekurangan dalam menangani atribut numerik pada *node*. Pengguna harus mengelompokkan data secara manual berdasarkan pengetahuan pengguna tentang data tersebut. Namun untuk data dengan ukuran yang sangat besar serta semakin bertambahnya data, pengelompokan *node* berdasarkan atribut numerik tidak dapat dilakukan dengan mudah.

Untuk itu dibutuhkan metode yang dapat menangani atribut numerik untuk melakukan peringkasan *graph*. Sehingga pengguna dapat mendapatkan ringkasan informasi maupun wawasan dari sebuah *graph* yang besar, tanpa harus mengolah data terlebih dahulu.

II. LANDASAN TEORI

2.1. Graph

Graph merupakan kumpulan simpul atau *node* yang terhubung oleh sisi yang disebut *edge*[10]. Secara formal, *graph* yang dinotasikan dengan $G = (V, E)$ dimana V merupakan himpunan tak beraturan dari himpunan dari *node* dapat dinotasikan dengan simbol $\{v_1, v_2, \dots, v_n\}$.



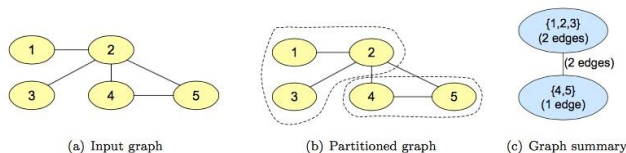
Gambar 1 (a) Undirected Graph (b) Directed Graph[4]

Gambar (a) merupakan *graph* tak berarah (*undirected graph*) dimana setiap sisinya tidak mempunyai arah sehingga $((v_i, v_j) \in E) \Leftrightarrow (v_j, v_i) \in E$. Sedangkan gambar (b) merupakan *graph* berarah (*directed graph*) dimana setiap sisinya mempunyai arah. Setiap pasangan *node* yang terhubung oleh *edge* ditentukan oleh arah *edge* yang ada sehingga pasangan *node* yang terhubung *edge* tidak sama dengan $((v_i, v_j) \in E) \Rightarrow (v_j, v_i) \notin E$. Dalam *graph* berarah, titik awal *node* dari sebuah sisi disebut *initial node*, sedangkan titik akhir dinamakan *terminal node*.

2.2. Graph Summarization

Graph summarization merupakan salah satu teknik yang digunakan untuk mengolah data dalam jumlah besar untuk mendapatkan informasi serta memahami karakteristik *graph*[4]. Dari *graph* dengan ukuran yang

sangat besar, terdiri dari jutaan *node* dan *edge* akan dihasilkan *summary* yang dapat dipahami dengan lebih mudah dengan ukuran yang lebih kecil. Tujuan dari *graph summarization* sendiri ada beberapa hal tergantung oleh kepentingan pengguna dalam upaya untuk menggali informasi dari *graph*, misalnya untuk menciptakan easily interpretable visualizations dari sebuah *graph*. Salah satu teknik *graph summarization* yang ada adalah Aggregation-Based Graph Summarization[7]. Teknik tersebut dapat dianalogikan dengan gambar dibawah ini.

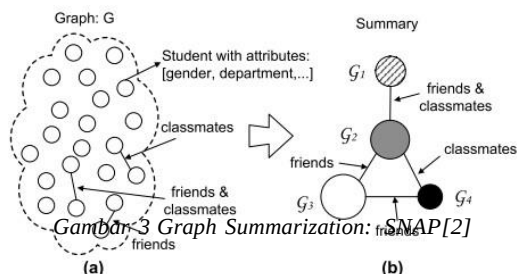


Gambar 2 Aggregation-Based Graph Summarization[4]

Teknik tersebut meringkas *graph* dengan melakukan pengelompokan *node* berdasarkan atribut yang terdapat pada *node*. Tujuan dari teknik Aggregation-Based Graph Summarization sendiri adalah menemukan pattern hubungan antar kelompok yang tercipta dalam hasil *summary* yang berupa *graph*. Pattern tersebut memberikan informasi yang mewakili informasi di *graph* asli. *Graph summarization* sendiri sangat berguna dalam hal mempercepat analisis data *graph* dengan menciptakan lossy concise representation dari *graph* asli dengan ukuran sangat besar.

2.3.! SNAP

Salah satu pengembangan teknik *graph summarization* yang ada adalah SNAP (Summarization by Grouping Nodes on Attributes and Pairwise Relationships). Metode ini akan mengolah data *graph* dan menghasilkan ringkasan berupa *graph* dengan ukuran yang lebih kecil dan informatif melalui *homogeneous grouping*[2]. SNAP termasuk dalam Aggregation-Based Graph Summarization yang memanfaatkan atribut dari setiap *node* dan struktur dari *edge* yang menghubungkan *node* dalam *graph* untuk mengelompokkan *node* dalam *graph* asli. Metode SNAP dapat digambarkan sebagai berikut.



Gambar 3 Graph Summarization: SNAP[2]

Pada gambar diatas, *graph* terdiri dari *node* yang merepresentasikan seorang murid. Setiap murid memiliki atribut gender, department, dll. Dengan menggunakan atribut gender yang merupakan *categorical node*, serta *edge* dengan tipe friends and classmate, SNAP akan memproduksi *graph* (b). Hasil dari SNAP merupakan sebuah *graph* yang terdiri dari beberapa *node* yang disebut

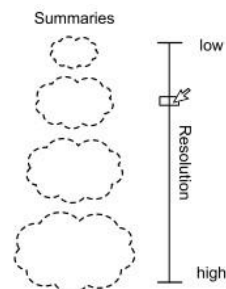
group, dan *edge* yang disebut *relationship*. Pada contoh di gambar tersebut, *group* 0_1 , 0_2 , dan 0_3 masing masing merupakan sebuah

node yang mewakili hasil pengelompokan *node* pada *graph* (a). Kumpulan *node* yang berada dalam *group* tersebut pasti memiliki kesamaan dari salah satu atribut yang ditentukan diawal untuk pengelompokan. Dari beberapa *node* tersebut, dihubungkan oleh *edge*

yang disebut *relationship*. Dengan mengamati *graph summary*, pengguna mendapatkan informasi mengenai pola hubungan antar *group* secara cepat.

2.4.! k-SNAP

k-SNAP merupakan metode yang melengkapi metode SNAP, dengan memberikan kemampuan kepada pengguna untuk mengontrol hasil ringkasan yang akan dihasilkan oleh metode SNAP. Metode ini memberikan kemampuan *drill down* dan *roll up*, yaitu kemampuan untuk mengontrol ukuran *summary* yang ingin dihasilkan[2]. Fitur ini dapat dianalogikan dengan gambar sebagai berikut.



Gambar 4 Fitur Zoom-in dan Zoom-Out pada k-SNAP[2]

Kemampuan *drill down* dan *roll up* lebih mirip pada analogi zoom in dan zoom out. Hasil *summary* dapat ditentukan dengan memberikan parameter inputan *k*. Nilai *k* akan merepresentasikan jumlah dari *group* yang dihasilkan pada *summary*. Jumlah *group* dalam *summary* selanjutnya disebut dengan *resolutions*. Dengan menggunakan fitur ini pengguna dapat mencari *summary* yang paling bagus dengan menjelajahi berbagai macam ukuran *summary*, mulai dari yang *summary* dengan *node* relatif sedikit kemudian bertambah sesuai dengan kebutuhan.

2.5.! Participation Ratio

Participation ratio merupakan nilai yang merepresentasikan hubungan antar dua *group*. Semakin besar nilai participation ratio, maka semakin kuat hubungan antar keduanya. Untuk menghitung rasio tersebut, terlebih dahulu didefinisikan himpunan *node* dari *group* 0_1 yang berpartisipasi di *group relationship* $(0_1, 0_3)$ sebagai $S_{0_1, 0_3}$ dan *relationship* $(0_1, 0_3)$ didefinisikan sebagai berikut.

$$B_{1,3} = \frac{C_{0_1, 0_3} \cdot F_{0_1, 0_3}}{C_{0_1} \cdot C_{0_3}} \quad (1)$$

Rasio partisipasi *group* $(0_1, 0_3)$ akan melihat anggota dari masing masing *group* yang saling berpartisipasi kepada *group* lawannya. Jika nilai participation ratio $< 50\%$, maka hubungan kedua *group* tersebut disebut *weak group relationship*. Sebaliknya jika $\geq 50\%$ disebut *strong group relationship*.

2.6.! SimLink

SimLink merupakan nilai yang merepresentasikan link structure dari sebuah pasangan *group*. Dari *group relationship* $(0_1, 0_3)$, SimLink didefinisikan dengan persamaan sebagai berikut.

$$(\quad) \Sigma$$

$$H \parallel K I = L \quad 0_1, 0_3 = \frac{1}{1,3OM} |B_{1,MN} B_{3,M}| \quad (2)$$

Nilai SimLink dihitung dari penjumlahan seluruh participation ratio dari *group relationship* 0_1 dan 0_3 dengan *group* yang lain dalam *graph*. Semakin kecil nilai SimLink, maka *group* 0_1 dan 0_3 semakin mirip dari link structure-nya.

2.7.1 Δ -measure

Nilai Δ -measure digunakan untuk mengukur kualitas hasil ringkasan *graph* dengan memeriksa perbedaan ringkasan tersebut terhadap ringkasan yang secara teori merupakan ringkasan yang ideal. Ringkasan yang ideal sendiri merupakan ringkasan dengan rasio partisipasinya 100% atau 0%. Yang berarti seluruh *relationship* yang menghubungkan *group* harus merupakan sebuah *strong group relationship*. $Graph, \Phi$, yang terdiri dari beberapa *group*, dimana $\Phi = \{O_1, O_2, O_3, \dots, O_M, O_N\}$, dapat dihitung Δ -measure dengan persamaan sebagai berikut.

$$\Delta \Phi = \sum_{G_i, G_j \in U} (S_{G_i}(O_1) + S_{G_j}(O_3)) \quad (3)$$

dimana,

$$S_{G_i}(O_1) = \begin{cases} |S_{G_i}(O_1)|, & \text{if } |S_{G_i}(O_1)| \leq 0.5 \\ |O_1| - |S_{G_i}(O_1)|, & \text{if } |S_{G_i}(O_1)| > 0.5 \end{cases} \quad (4)$$

Persamaan Δ -measure diatas melihat seluruh kemungkinan pasangan *group* yang ada dalam *graph* tersebut. Jika pasangan *group* tersebut merupakan *weak group relationship* maka yang diambil adalah kardinalitas dari himpunan $S_{G_i}(O_1)$. Sedangkan apabila pasangan *group* tersebut merupakan *strong group relationship*, maka nilai S nya merupakan nilai kardinalitas dari himpunan *group* O_1 dikurangi dengan kardinalitas himpunan $S_{G_i}(O_1)$.

2.8.1 Interestingness Measure

Interestingness measure merupakan tolak ukur yang digunakan untuk menilai kualitas dari *summary graph*[1]. Metode ini memungkinkan pengguna untuk memilih *summary* yang lebih berkualitas secara otomatis. *Interestingness* terdiri dari tiga parameter penilaian.

a.1 Diversity

Parameter ini didapat dengan melihat banyak hubungan yang kuat antara *group* yang terdiri atas *node* dengan nilai atribut berbeda, sehingga membuat ringkasan dalam *graph* menjadi lebih beragam. Nilai *diversity* didapat dengan persamaan sebagai berikut.

$$Diversity(H) = \frac{|C(H)|}{|H|} \quad (5)$$

Nilai C merupakan jumlah kategori yang terdapat di ringkasan *graph*. Kemudian H merupakan himpunan pasangan dari kategori dalam ringkasan yang terhubung oleh satu atau lebih *strong group relationship*. Semakin tinggi nilai *diversity* dari sebuah *summary*, berarti semakin banyak terdapat *strong group relationship* pada *summary* tersebut.

b.1 Coverage

Coverage menghitung *strong group relationship* yang terdapat pada *summary*, dimana semakin banyak *node* yang berpartisipasi dalam hubungan tersebut maka hasil dari ringkasan akan semakin komprehensif. Nilai dari *coverage* dihitung berdasarkan persamaan berikut.

$$Coverage(H) = \sum_{(G_i, G_j) \in E(a)} \frac{|C_{G_i}(G_j)|}{|G_i|} \quad (6)$$

$Hf(H)$ merupakan himpunan dari *strong group relationship* di ringkasan, $\#$ merupakan himpunan dari *node* di *graph* awal dan

$S_{G_i}(O_1)$ merupakan himpunan dari *node* di O_1 yang berpartisipasi dalam *group relationship* (O_1, O_3) .

c.1 Conciseness

Conciseness merepresentasikan kepadatan dari sebuah *graph summary*. Kepadatan sebuah *graph* dihitung dari banyak jumlah *group* dan *strong group relationship* dalam sebuah ringkasan *graph*.

$$Conciseness(H) = \frac{|H|}{|G| + |Hf(H)|} \quad (7)$$

Nilai $|G|$ merupakan jumlah *node* dalam *graph* dan $Hf(H)$ merupakan himpunan dari *strong group relationship* pada *graph*. Nilai *conciseness* berbanding terbalik dengan kepadatan sebuah *graph summary*. Semakin rendah nilai *conciseness*, maka semakin padat *graph* tersebut yang berarti semakin sedikit jumlah *group*.

Dari ketiga parameter diatas, kemudian dihitung nilai *interestingness*

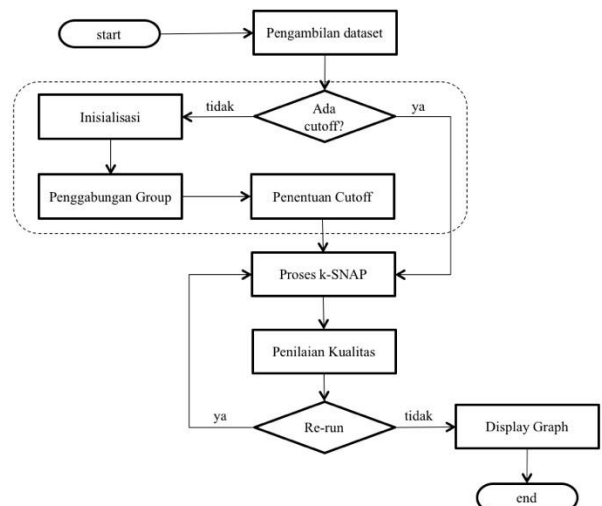
nya dengan persamaan sebagai berikut.

$$Interestingness(H) = \frac{Diversity(H) \times Coverage(H)}{Conciseness(H)} \quad (8)$$

Sebuah *summary* termasuk dalam kategori ringkasan yang informatif jika nilai *Diversity* dikali dengan *Coverage* nya semakin besar, namun ringkasan tersebut jarang digunakan karena ukuran ringkasan yang terlalu besar. Untuk itu, nilai *interestingness* membagi hasil perkalian kedua poin tersebut dengan nilai *Conciseness*. Semakin tinggi nilai *interestingness*nya, maka kualitas dari *summary* tersebut adalah semakin baik.

III.1 PERANCANGAN SISTEM.

Pada penelitian ini dibangun sistem *graph summarization* dengan metode CANAL. Sistem yang akan dibangun tergambar pada diagram berikut.



Berdasarkan diagram blok dari sistem yang akan dibangun, mula mula dilakukan pengambilan data dari database. Selanjutnya data tersebut dimodelkan kedalam *graph* yang akan digunakan sebagai input dalam proses selanjutnya. *Graph* tersebut kemudian akan memasuki proses pencarian *cutoff* dengan metode CANAL. Dari *cutoff* yang dihasilkan, selanjutnya *graph* tersebut akan diringkas dengan algoritma k-SNAP. Hasil ringkasan selanjutnya akan dinilai berdasarkan *interestingness measure* untuk mendapatkan *graph summary* dengan kualitas yang baik. Penilaian *interestingness* ini merupakan penilaian

kualitas ringkasan *graph* dari SNAP yang melihat sebuah *summary* dari tiga aspek, yaitu *diversity*, *coverage*, dan *conciseness*. Dataset yang

digunakan merupakan collaboration network dari Stanford Large Network Dataset Collection. Hasil akhir dari sistem merupakan sebuah *graph summary* yang mengandung karakteristik dasar dari *graph* asli. Dari *graph summary* tersebut, pengguna dapat melihat pola hubungan antar kelompok yang dihasilkan.

Inti dari sistem ini adalah penerapan metode CANAL yang digunakan untuk mengelompokkan atribut numerik pada *node*. CANAL sendiri terbagi menjadi tiga tahap, yaitu.

a.! Tahap Inisialisasi

Tahap 1: Inisialisasi

- 1.! **INPUT:** Graph G, Atribut a, dan Kategori C
- 2.! Kelompokkan *node* di G berdasarkan atribut a
- 3.! Urutkan *group* berdasarkan range atribut untuk dapatkan value adjacent *group* (gi, gj)
- 4.! Untuk setiap pasang value adjacent *group*, hitung nilai SimLink. Masukkan SimLink, gi, gj kedalam heap SimLink
- 5.! Hitung Δ -measure untuk hasil grouping sementara

Tahapan pertama dari CANAL merupakan tahap inisialisasi *graph* sebagai parameter inputan hingga penggabungan beberapa *node* menjadi sebuah *group*. Tujuan dari tahap inisialisasi adalah untuk menemukan sepasang value *adjaent group* yang dikelola dalam sebuah

heap.

b.! Tahap Penggabungan Group

Tahap 2: Penggabungan Group

- 1.! **WHILE** heap SimLink **NOT EMPTY DO:**
- 2.! **POP** heap dengan nilai SimLink paling rendah (*min sort*)
- 3.! Merge gi dan gj, kemudian hitung μ_p
- 4.! **PUSH** μ_p , gi, gj kedalam heap μ_p (*max sort by μ_p*)
- 5.! **END WHILE**

Pada penggabungan *group* dilakukan pengambilan elemen heap simlink hasil dari tahapan pertama dengan nilai simlink paling rendah. Untuk setiap elemen yang diambil (*pop*), didapatkan dua *group* O_1 dan O_3 yang merupakan pasangan value adjacent *group*. Untuk pertama kali penggabungan, setiap *group* tersebut memiliki anggota *node* yang mempunyai nilai atribut yang sama. Selanjutnya, masing masing *group*

memiliki batas atas dan batas bawah masing masing terhadap atribut yang digunakan. Kedua anggota tersebut kemudian digabung untuk membentuk sebuah *group* baru yang mempunyai anggota dari kedua *group* tadi. Setelah dilakukan *grouping* dari kedua *group* tersebut, kemudian dihitung nilai t_u . Dari nilai t_u , *group* O_1 , dan *goup* O_3 dimasukkan sebagai satu elemen di heap v_8 . Setiap selesai menggabungkan *group*, heap simlink diperbaharui dengan memperhatikan susunan pasangan value adjacent *group*-nya.

c.! Tahap penentuan Cutoff

Tahap 3: Penentuan Cutoff

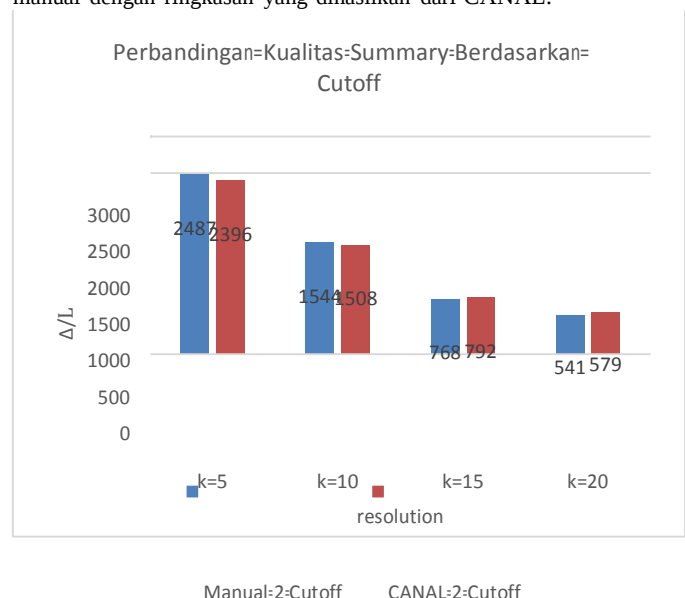
- 1.! **FOR** kategori C-1 **DO:**
- 2.! **POP** elemen dalam heap μ_p dengan nilai μ_p paling besar

- 3.! Dari gi dan gj, ambil range kedua *group* tersebut sebagai *cutoff* lalu simpan di array O
- 4.! **END FOR**
- 5.! **OUTPUT:** Array O yang berisi elemen *cutoff* untuk atribut a

Dari heap v_8 yang dihasilkan di tahap sebelumnya, dilakukan pengambilan elemen dengan nilai t_u tertinggi sebanyak jumlah kategori yang diinputkan di tahap inisialisasi beserta kedua *group* yang terkait. Nilai sebuah *cutoff* diambil nilai dari range kedua *group* dan dijadikan sebagai batas atas dan bawah dari sebuah *cutoff*.

IV.! HASIL PENGUJIAN

Metode CANAL digunakan untuk menggali informasi yang dapat dijadikan sebagai bahan pengelompokan *node* dalam *graph*[1]. Pengujian dilakukan dengan membandingkan kualitas ringkasan *cutoff* manual dengan ringkasan yang dihasilkan dari CANAL.



Kualitas *summary* yang dihasilkan diukur dengan nilai $\frac{\Delta}{M}$. Percobaan dilakukan sebanyak empat kali dengan nilai k yang berbeda. Dari hasil percobaan diatas, untuk nilai $\frac{\Delta}{M}$ pada k minimum mempunyai nilai $\frac{\Delta}{M}$

yang tinggi. Dengan bertambahnya nilai k maka nilai $\frac{\Delta}{M}$ semakin turun. Grafik tersebut juga menunjukkan bahwa kualitas hasil *summary* dengan *cutoff* yang didapat secara otomatis dari CANAL cenderung mendekati nilai kualitas *summary* dengan *cutoff* yang dibuat secara manual.

Selanjutnya akan dilakukan perbandingan antar hasil *cutoff* dengan jumlah resolution dan categories yang berbeda. Berikut ini merupakan hasil yang didapatkan.



Dua titik puncak dari nilai *interesting measure* tersebut merepresentasikan dua jenis *graph summary* yang dihasilkan[1]. Dalam koordinat titik puncak, *graph summary* dengan nilai *k* dan *C* minimal merepresentasikan sebuah *Overall summary*. Sedangkan pada puncak maksimal kedua, merepresentasikan sebuah *Informative summary*.

Sebuah *overall summary* terdiri dari jumlah *group k* yang sedikit, sehingga memiliki nilai *Conciseness* yang rendah dan terdiri dari *strong group relationship* yang dominan serta memiliki nilai *Coverage* dan *Diversity* yang tinggi. Jika dilihat dari segi kemudahan pemahaman dari visualisasi *summary graph*, maka *overall summary* ini merupakan *summary* yang paling mudah untuk dipahami.

Sedangkan untuk sebuah *informative summary*, walaupun memiliki nilai *Conciseness* yang tinggi artinya terdiri dari *group* yang relatif banyak namun dapat memberikan pengetahuan kepada user akan berbagai hubungan antar *group* yang dapat mengarahkan pada informasi baru dari data tersebut. Sebuah *informative summary* mempunyai ukuran yang relatif besar dibandingkan dengan *overall summary*, namun menunjukkan lebih banyak pola hubungan dari *graph* asli.

REFERENSI

- [1] Zhang, Ning, Yuanyuan Tian, and Jignesh M. Patel. "Discovery-driven graph summarization." *Data Engineering (ICDE), 2010 IEEE 26th International Conference on*. IEEE, 2010.
- [2] Tian, Yuanyuan, Richard A. Hankins, and Jignesh M. Patel. "Efficient aggregation for graph summarization." *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. ACM, 2008.
- [3] Buerli, Mike, and C. P. S. L. Obispo. "The current state of graph databases." *Department of Computer Science, Cal Poly San Luis Obispo, mbuerli@calpoly.edu* (2012): 1-7.
- [4] Riondato, Matteo, David Garcia-Soriano, and Francesco Bonchi. "Graph Summarization with Quality Guarantees." *Data Mining (ICDM), 2014 IEEE International Conference on*. IEEE, 2014.
- [5] Navlakha, Saket, Rajeev Rastogi, and Nisheeth Shrivastava. "Graph summarization with bounded error." *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. ACM, 2008.
- [6] Navlakha, Saket, Michael C. Schatz, and Carl Kingsford. "Revealing biological modules via graph summarization." *Journal of Computational Biology* 16.2 (2009): 253-264.
- [7] Angles, Renzo, and Claudio Gutierrez. "Survey of graph database models." *ACM Computing Surveys (CSUR)* 40.1 (2008): 1.
- [8] Gross, Jonathan L., and Jay Yellen, eds. *Handbook of graph theory*. CRC press, 2004.

- [9] Viégas, Fernanda B., and Judith Donath. "Social network visualization: Can we go beyond the graph." *Workshop on social networks, CSCW*. Vol. 4. 2004.
- [10] Tian, Yuanyuan. *Querying graph databases*. ProQuest, 2008.
- [11] Huan, Jun, Wei Wang, and Jan Prins. "Efficient mining of frequent subgraphs in the presence of isomorphism." *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*. IEEE, 2003.
- [12] "substrate interface," *IEEE Transl. J. Magn. Japan*, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
- [13] M. Young, *The Technical Writer's Handbook*. Mill Valley, CA: University Science, 1989.

