Analisis Algoritma RP-GD dalam Kualitas Peringkasan Graf dari Basisdata Graf

Defrianda Rizky Pranata[1], Kemas Rahmat S. W. [2], Shaufiah[3]

[1.2.3] *Teknik Informatika, Fakultas Informatika, Telkom University*

Jalan Telekomunikasi No.1, Dayeuh Kolot, Bandung 40257 defrianda@gmail.com[1], bagindokemas@telkomuniversity.ac.id[2], ufi@telkomuniversity.ac.id[3]

Abstrak

Basisdata graf merupakan representasi dari pemodelan suatu koleksi data ke dalam bentuk *Node* dan *Edge*. Basisdata Graf juga merupakan sebuah bentuk atau model dari *database* yang menyediakan solusi efektif dan efisien terhadap penyimpanan data . Dikembangkan di era *bigdata* seperti sekarang ini merupakan terobosan baru di bidang *Computer Science* khususnya *Data Engineering.* Terdiri dari *Edges, Nodes*, dan *Properties* yang digunakan untuk merepresentasikan dan menyimpan data. Bersifat *index-free adjacency* yang berarti bahwa setiap elemen berisi pointer langsung ke elemen yang berdekatan dan tidak ada pencarian indeks diperlukan. *Database* grafik umum yang dapat menyimpan grafik pun berbeda dari *database* grafik khusus seperti *triplestores* dan *database* jaringan.

Ketika hanya menggunakan model *database* yang berbentuk *relational database* tentunya semakin lama semakin kesulitan karena datanya disini sangatlah banyak sekali. Disinilah penulis akan menggunakan model *database* yang masih tergolong baru, yaitu *Graph Database*. Model ini dapat merepresentasikan banyak data dalam suatu graf yang bisa dianalisis serta diambil kesimpulannya dari banyak simpul serta busur yang penulis peroleh dari dataset molekuler ikatan kimia. Dengan menggunakan model ini, tentunya dapat dilihat ringkasan molekuler yang dapat dilihat dari analisa dan peringkasan basisdata graf yang penulis ambil sebagai topik dari penulisan karya ini. Metode peringkasan yang penulis ambil adalah *RP-GD Algorithm* yang penulis gunakan mempunyai efisiensi dan kualitas yang dapat meringkas suatu basisdata graf. Diharapkan algoritma tersebut bisa meningkatkan kualitas dari sebuah *graph database* sehingga peringkasan dari model tersebut mempunyai hasil yang maksimal dalam merepresentasikan molekuler ikatan kimia dari dataset tersebut.

Dari hasil pengujian dan analisis, maka terbukti bahwa algoritma RP-GD dapat digunakan dalam peringkasan basisdata graf, serta menghasilkan kualitas yang baik dalam pemrosesan maupun hasilnya. Dilihat dari jumlah nodes dan edges hasil peringkasan lalu cakupan informasi dan rasio peringkasan menjadi parameter yang menunjukkan hasil tersebut. Variasi hasil peringkasan juga dapat dilakukan sesuai dengan minimum support yang diinginkan. Nilai cakupan informasi dari sebuah ringkasan basisdata graf berbanding lurus dengan nilai *minimum support* yang diberikan, sedangkan rasio peringkasan berbanding terbalik dengan nilai *minimum support* yang diberikan.

Kata Kunci : *Graph Database*, *RP-GD Algorithm*, *dataset SMILES*, *summarization graph*, *summariation quality*, *chemical compounds*, *chemical informatics*

Abstract

Graph database is a representation of modeling a collection of data into a form and Edge Node. Graph database is also a form or model of the database that provides effective and efficient solution for data storage. Developed in bigdata era like now is a new breakthrough in the field of Computer Science Engineering Data especially. Consisting of Edges, Nodes, and Properties are used to represent and store data. Are index-free adjacency, which means that each element contains a direct pointer to the adjacent elements and there is no search index is required. General graph database that can store any different graphs of specialized graphics such as triplestores database and network databases.

When only using the model database in the form of relational databases is certainly becoming increasingly difficult because the data here is very much at all. This is where the author will use the database model is still relatively new, ie Graph Database. This model can represent a lot of data in a graph that can be analyzed and conclusions drawn from the number of nodes and arcs that the authors obtained from molecular dataset chemical bonds. By using this model, of course, can be seen a summary of which can be seen from the molecular analysis and graph database summarization, authors take as the topic of the writing of this work. Summarization method that the authors take is RP-GD Algorithm that author use have an efficiency and quality can summarize a graph database. The algorithm is expected to improve the quality of a graph database so

summarization of these models have maximum results in molecular represents the chemical bonding of the dataset.

From the test results and analysis, it is evident that the RP-GD algorithm can be used in summarization graph database, as well as produce a good quality in processing and results. Judging from the number of nodes and edges summarization results and coverage summarization ratios information and a parameter that indicates the results. Variations of summarization results can also be carried out in accordance with the desired minimum support. Value range of information from a summary graph database is proportional to the value of a given minimum support, while the ratio of summarization inversely proportional to the value of a given minimum support.

Keywords: *Graph Database*, *RP-GD Algorithm*, *dataset SMILES*, *summarization graph*, *summariation quality*, *chemical compounds*, *chemical informatics*

1. Pendahuluan

Di era bigdata yang sedang populer seperti ini tentunya pengolahan data sangatlah penting dan menjadi hal mutlak dalam merepresentasikan penggunaan teknologi informasi baik di perusahaan maupun instansi di seluruh dunia.

Relational database merupakan sistem penyimpanan dan pengambilan data yang telah populer dan mendominasi selama lebih dari tiga dekade. Banyak aplikasi yang menggunakan relational *database* untuk penyimpanan data. *Relational database* dapat bekerja dengan baik apabila jumlah data sedikit dan memiliki data terstruktur[1]. Ketika terjadinya peningkatan jumlah data dan berbagai pemrosesan data maka relational *database* dengan skema yang kaku sangat tidak cocok untuk kasus data semi terstruktur bahkan tidak terstruktur[1]. Kasus data tidak terstruktur dan semi terstruktur memiliki fleksibilitas dalam hal pemrosesan data seperti halnya dengan kasus social network dengan interconnected data.

Salah satu solusi yang digunakan untuk mengatasi hal tersebut adalah menggunakan *Graph Database.* basisdata graf adalah salah satu metode implementasi dari NoSQL (*Not Only SQL*) yaitu sistem basisdata yang berguna menyimpan data dalam jumlah besar dan direpresentasikan ke dalam graf, berbentuk *Node* dan *Edge*[2]. Hal ini dilakukan karena *Node* dan *Edge* memberi peluang untuk ekstraksi informasi antar user. Kelebihan basisdata graf adalah dalam hal pencarian data bisa dilakukan secara tranversal dengan setiap relasi direpresentasikan dengan suatu *Edge* yang menghubungkan *Node-Node* yang berelasi, sehingga waktu pemrosesan dapat dilakukan dengan efektif[2].

Namun ketika hanya menggunakan model *database* yang berbentuk relational *database* tentunya semakin lama semakin kesulitan karena datanya disini sangatlah banyak sekali. Disinilah penulis akan menggunakan model *database* yang masih tergolong baru, yaitu *Graph Database*. Model ini dapat merepresentasikan banyak data dalam suatu graf yang bisa dianalisis serta diambil kesimpulannya dari banyak *Nodes* serta *Edges* yang penulis peroleh dari dataset NCI (National Cancer Institute) dan Cheminformatics.

Penulis memilih dataset NCI dan Cheminformatics disini adalah karena format *input* yang sudah terstandarisasi sehingga dalam menggali informasi dari ikatan molekul kimia tersebut dapat terlaksana dengan baik. Kemudian format yang dipakai oleh dataset ini adalah SMILES, yaitu bertipe *Simplified Molecular-Input Line-Entry System* (SMILES) yaitu suatu spesifikasi dalam bentuk notasi baris untuk menggambarkan struktur spesies kimia menggunakan string ASCII pendek. String dari SMILES dapat diimpor oleh aplikasi untuk dikonversi menjadi ringkasan molekul[6].

Dengan model basisdata graf tersebut diharapkan dapat meningkatkan kualitas untuk merepresentasikan informasi yang terkandung di dalamnya. Kemudian dilakukan peringkasan graf untuk lebih mencapai tujuan akhir yaitu sebuah ringkasan graf yang menjadi intisari dari dataset basisdata graf tersebut.

Peringkasan basisdata graf penulis ambil sebagai topik dari penulisan tugas akhir ini. Metode peringkasan yang penulis ambil adalah RP-GD Algorithm yang penulis ambil mempunyai efisiensi dan kualitas yang dapat dengan mudah meringkas suatu basisdata graf[1]. Kualitas hasil peringkasan graf juga diukur berdasarkan cakupan informasi serta rasio peringkasan sebagai parameter kualitasnya.

Proses peringkasan ini mengkombinasikan berbagai keuntungan seperti skalabilitas, konsumsi memory, skema pembangunan basisdata dan kapasitas untuk menguasai berbagai data menjadi data pilihan dari pengguna. Ringkasan yang dihasilkan menyediakan pandangan data yang dapat dibentuk sesuai dengan keinginan pengguna[3].

Selain itu, ada alasan pentingnya melakukan peringkasan basisdata, yaitu kembali kepada definisinya, perubahan penyusutan dari basisdata menjadi bentuk yang ringkas, melalui proses pengurangan isi dengan cara menyeleksi dan/atau menyamaratakan dari apa-apa yang penting di dalam basisdata tersebut[8]. Sehingga tujuan utama penulis melakukan peringkasan adalah untuk memberikan gagasan/ide pokok dari basisdata yang asli namun dalam bentuk yang ringkas.

Dari hasil peringkasan tersebut juga membuang data yang ‘tidak diperlukan’. Konsentrasi yang penulis berikan disini adalah dari sudut pandang demi kepuasan pengguna, sehingga dalam meringkas basisdata graf, dapat mengurangi simpul-simpul yang tidak ada hubungannya dengan topik yang pengguna inginkan. Alasan lain adalah, tentunya dapat mengurangi beban memori serta proses query yang dilakukan.

Seringkali *nodes* (simpul) mempunyai atribut yang berhubungan dengan diri mereka. Dalam banyak aplikasi, graf berukuran sangat besar,

dengan ribuan atau bahkan jutaan simpul dan tepi. Hasilnya, hamper mustahil untuk memahami informasi yang terkandung di dalamnya hanya dengan melihat sekilas saja. Maka, metode peringkasan graf sangat dibutuhkan agar membantu pengguna menggali dan memahami informasi pokok yang terkandung didalamnya[7].

2. Landasan Teori

2.1 Graf

Teori graph ini pertama kali dikenal sejak Leonhard Euler menulis paper "Seven Bridges of Königsberg" pada tahun 1736. Dalam matematika, graph ini merupakan bagian utama dari Matematika Diskrit. Meski banyak buku yang ditulis terkait dengan teori graph, buku dari Frank Harary tentang teori graph pada 1969 membuat banyak peneliti dari berbagai disiplin ilmu menyadari pentingnya teori graph terkait dengan berbagai macam disiplin ilmu. Teori graph ini juga dikembangkan di dunia software untuk pemodelan data, terutama untuk pemodelan dan analisis yang memerlukan keterkaitan dalam bentuk graph.

Secara konsep, sebuah graf dibentuk oleh vertex (*Node* atau simpul) dan *Edge*(lines atau sisi) yang menghubungkan vertex(*Node* atau simpul). Secara formal, sebuah graf adalah sepasang dari elemen G = (V,E) dimana V adalah sekumpulan simpul dan E adalah sekumpulan sisi yang dibentuk oleh sepasang simpul. Cara umum untuk menggambar graf adalah menggambar sebuah titik untuk setiap vertex dan menggabungkan kedua titik tersebut dengan sebuah garis. Tidak ada yang dianggap tidak relevan hanya bagaimana menggambar sebuah titik dan garis. Hal yang terpenting adalah informasi pasangan vertex mana yang membentuk sebuah *Edge* dan mana yang tidak[2].

2.2 Basisdata Graf

Basisdata graf adalah basisdata yang menggunakan graf dalam pemodelan data. Pada basisdata graf, data akan dimodelkan dalam bentuk graf (terdiri atas sekumpulan vertex (node) dan dihubungkan dengan dengan edge / arc) yang memungkinkan berbagai macam operasi dilakuan pada struktur data tersebut, misalnya traversal, pattern matching, penemuan "knowledge" baru (reasoning), dan lain-lain. Banyak aspek kehidupan manusia pada dasarnya bisa dimodelkan dengan graf. Contoh paling sederhana, fasilitas rekomendasi teman pada situs jejaring sosial Facebook adalah hasil dari penerapan teori graf: saya mengenal A, A mengenal B, B mengenal C, C mengenal D, dan seterusnya, kemudian dicari pola yang sama antar individu tersebut sehingga bisa dihasilkan rekomendasi bahwa saya mungkin juga mengenal D. Selain itu bisa dikatakan basisdata graf adalah sebuah database yang menggunakan struktur graf dengan Node, Edge, dan Property untuk merepresentasikan dan menyimpan data. Graph merupakan cabang ilmu dari matematika yang dikenal mempunyai keterkaitan aplikasi dengan banyak disiplin ilmu lainnya. Perkembangan akhir-akhir ini menunjukkan bahwa teori graph ini bisa diterapkan untuk database dan membentuk basisdata graf.

Sebuah sistem manajemen basisdata graf atau basisdata graf adalah sebuah sistem manajemen database online dengan metode Create, Read, Update dan Delete (CRUD) yang mengekspos sebuah model data graph. basisdata graf umumnya dibangun untuk penggunaan sistem transaksional atau OLTP. Oleh karena itu, basisdata graf biasanya dioptimalkan untuk performansi transaksional dan dibangun dengan integritas transaksional dan ketersediaan operasional..

2.3 Peringkasan Graf

Dataset graf dalam skala besar ada dimana-mana, termasuk dalam jejaring social, biologi. Teknik peringkasan graf sangat penting dalam lingkup tersebut dikarenakan dapat membantu membongkar wawasan berguna tentang pola tersembunyi dalam mendasari data [1]. Pentingnya peringkasan graf disini untuk membuat sebuah ringkasan graf berdasarkan atribut *node* dan *edge.* Dalam membuat peringkasan pola graf dikenalkan formula yang dapat dipakai :

Labeled Graph

Sebuah graf *G* yang dilabeli mempunyai 5 kumpulan elemen data yang dikumpulkan yaitu *G* (*V*, *E*, ℒ, *L*) dimana *V* adalah kumpulan simpul dan *E* adalah kumpulan dari tepi. Fungsi L melabelkan pemetaan $V \rightarrow$ ℒ dan $E \rightarrow$ ℒ.

Subgraph Isomorphism

Sebuah subgraf *G'* (*V'*, *E'*) disebut subgraf isomorfisme dari *G* (*V*, *E*) dimana simpul dan tepi adalah himpunan bagian dari V dan E masing-masing :

- $\forall u \in V(G), (l(u) = l'(f(u)))$
- $\forall (u,v) \in E(G), (f(u),f(v)) \in E(G')$ dan $l(u,v) = l'(f(u),f(v))$

Kemudian ketika *G* adalah subgraf isomorfis ke *G'*, maka *G* adalah subgraf dari *G'* dan *G'* merupakan supergraf dari *G*, dinotasikan dngan $G \subseteq G'$

Frequent Subgraph

Diberikan sebuah graf *G* dan sekumpulan graf *D* = {*g1*, *g2*, *g3*,..., *gn*}, support *G* adalah :

$$Support\ (G) = \frac{number\ of\ graphs\ in\ D\ that\ contain\ G}{|D|}$$

sebuah graf *G* dalam dataset *D* disebut Frequent jika mendukung tidak kurang dari threshold (batasan) yang telah ditentukan.

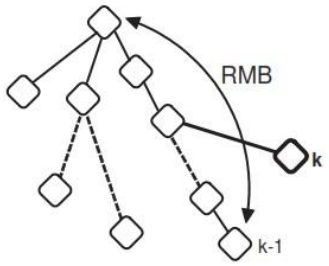
Frequent Subgraph Mining

Diberikan sebuah graf dataset, *GS* = {G_i | i = 0...n} dan sebuah minimum support σMaka σ (*g*, *GS*) dinotasikan sebagai frekuensi terjadinya *g*

dalam *GS*. *FSM* dipakai untuk menemukan setiap graf *g*, jika σ (*g, GS*) lebih besar atau sama dari *minimum support* σ.

Minimum Depth First Search Code (M-DFSC)

Kemudian *canonical representation* yang digunakan yaitu *Minimum Depth First Search Code* (M-DFSC), skema yang merepresentasikan dengan 5 kumpulan elemen data (*l,j, le li,lj*), dimana i dan j adalah ID simpul, *li* dan l_j adalah label dari simpul tersebut, dan l_e adalah label tepi yang menghubungkan simpul. DFS Code ini menghasilkan subgraf dengan cara Right Most Expansion dapat dilihat pada gambar dibawah ini :



Gambar 1 Right Most Extension

Jaccard Distance

Cara kerja dari *Jaccard Index* yaitu untuk menghitung kesamaan dan perbedaan antara dua buah himpunan. Dalam kasus ini, yang dibandingkan adalah himpunan *Node* tetangga pada dua buah *Node*. Diberikan contoh Apabila ada dua buah *Node a* dan *b*, yang masing-masing memiliki himpunan tentangga A dan B, maka nilai jaccard dari *Node a* dan *b* adalah :

$$J(a,b) = 1 - \frac{|A \cap B|}{|A \cup B|}$$

Gambar 2 Jaccard Distance

δ-cover

Pola graf *P* adalah *δ-covered* oleh pola graf lain *P* ' jika $P \subseteq P'$ dan $D(P, P') \leq \delta$.

δ-cover graph

Misalkan *S* kumpulan pola graf dan *δ* menjadi *distance threshold* (batas jarak) antara pola graf dalam *S*. Grafik *δ-cover S* didefinisikan sebagai grafik terarah *Gδ (S)*, dimana setiap *node* sesuai dengan pola graf di *S*. Jika pola graf *Pi* adalah *δ-cover* pola graf lain *Pj* (*Pi* ≠ *Pj*), maka terdapat *edge* terarah dari *Pi* ke *Pj*.

Jump Value

Misalkan *P* menjadi pola graf dalam satu kumpulan pola graf *S*. Jika *JV (P)*> *δ*, maka *P* disebut *δ-jump* di *S*.

Dalam graf *cover* $G_{\delta,}$ *node* dengan derajat masuk sebesar 0 sesuai dengan pola *δ-jump*.

Graph Pattern Summarization

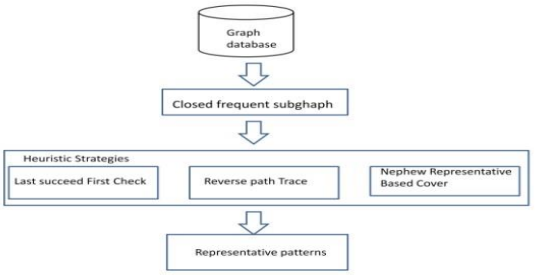
Diberikan *D* basisdata graf, *minimum support M* dan *distance threshold δ*, pola peringkasan graf adalah untuk menemukan satu set representasi pola graf RS, seperti bahwa untuk setiap *frequent graph P (w.r.t. M)*, terdapat representasi graf *Pr* ∈ *RS (w.r.t. M)* yang *δ-cover P*, dan nilai | *RS* | diminimalkan.

2.4 Algoritma RP-GD

Algoritma RP-GD ini dikenalkan oleh Jianzhong Li, Yong Liu, dan Hong Gao untuk membuat ringkasan dari pola graf dalam teknik peringkasan graf tersebut. Di dalam kehidupan nyata dimana jumlah dari *closed frequent patterns* yang digunakan berukuran sangat besar, sehingga algoritma RP-FP yang merupakan metode sebelum pengembangan RP-GD tidak dapat memberikan skala yang baik.

Lalu dikembangkan algoritma yang lebih efisien untuk membangkitkan set representasi pola graf yang secara langsung dari basisdata graf. Algoritma RP-GD, yang mana bisa mengkalkulasi set representasi dari pola graf secara simultan selama proses penggalian data. Ada tiga strategi heuristic dari RP-GD ini adalah :

- Last Succeed First,
- Reverse-Path-Trace-Strategy
- Nephew-Representative-Based-Cover Strategy



Gambar 3 Strategi Heuristik RP-GD

Cara Kerja Algoritma RP-GD

1. Diberikan P sebagai *closed frequent subgraph*
2. Secara sekuensial meninjau keluaran dari pola graf
3. P dikunjungi CloseGraph di waktu pertama, keluaran P hanya ketika P telah dikunjungi di waktu kedua atau setelah mengunjungi semua anak dari P maka keluaran ini disebut *covered-order*
4. Ketika pola graf P yang muncul sebagai keluaran, maka pertama kita cek apakah *jump pattern* atau tidak.
5. Jika iya, buat representasi pola graf baru.
6. Jika tidak, RP-GD berusaha untuk menemukan representasi R yang mana R dapat mencakup P.
7. Jika keluaran pola graf P bukan *jump pattern* dan tidak dapat

menemukan representasi graf yang dapat mencakup P, maka dipanggil P sebagai *greedy graph pattern.*

8. Ketika menghadapi *greedy graph pattern*, dibangun sebuah representasi pola graf yang mencakup P berdasarkan tiga pemilihan heuristic strategi.
9. Integrasikan mekanisme peringkasan kedalam CloseGraph, lalu gali representasi pola graf secara langsung dari bassisdata graf.

***Pseudo-code* Algoritma RP-GD**

Untuk memfasilitasi keefisienan peringkasan dari pola graf, konsep yang perlu diperhatikan antara lain : cover graph, jump value, jump pattern.

Algoritma RP-GD

Input : A graph database *D*;
A minimum support *M*;
A distance threshold δ
Output : A set of representative graph patterns *RS*

1. Remove infrequent vertices and edger from *D*;
2. S^1 = code of all frequent 1-edge from *D* (w.r.t.*M*);
3. $GS = \varnothing$;
4. **For** each node *s* in S^1 **do**
5. Call GetRepresentative (*s, NULL ,D, M,*).

Function : GetRepresentative

Input : A minimum DFS code *s,*
Its parent ,
A graph Database *D,*
A minimum support *M*;
A distance threshold δ
Output : A set of representative graph patterns *RS*

1. **If** $s \neq \min(s)$ **then**
2. Return;
3. **If** $\exists cp$, *cp* is equivalent-occuring child of *p* and *s* > *cp* in term of DFS lexicographic order **then**
4. Return;
5. $C = \varnothing$;
6. Scan *D* once, find every edge *e* such that *s* can be extended to frequent subgraph $s \diamond_x e$; insert $s \diamond_x e$ into *C*;
7. Compute the jump value JV(s) of *s;*
8. Push the last edge of *s* into *GS*;
9. **If** JV (*s*) = 0 **then**
10. *GS*[*top*].covered = true;
11. **If** JV (*s*) > 0 **then**
12. **For** each entry *P* (frequent subgraph *P*) in *GS* from the root **do**
13. **If** *GS*[*P*].*covered* = false, P is *covered* by *s*, and *s* is better than *GS* [*P*].R_{cand} in term of heuristic selection strategy **then**
14. *GS*[*P*]. $R_{cand} = s$;
15. Remove $s \diamond_x e$ from *C* which cannot be right-most extended from *s*;
16. Sort *C* in the lexicographic DFS order;
17. **For** all frequent right-most children $s \diamond_x e$ of *s* **do**
18. Call GetRepresentative ($s \diamond_x e$, *s, D, M,* δ);
19. **If** JV (*s*) > δ **then**
20. Build a new representative $R_{new} = s$ and put R_{new} into *RS*;
21. **For** each entry *P* in *GS* from the root **do**
22. If *GS*[P].*covered* = false, *P* can be *covered* by R_{new} then
23. *GS*[*P*].*covered* = true;
24. **If** *GS*[*top*].covered = false **then**
25. Search a representative graph pattern R which can cover s in RS;
26. **If** there exists no such representative pattern R **then**
27. Build a new representative pattern R_{new} from *GS*[*top*].R_{cand} and put R_{new} into *RS*;
28. **For** each entry *P* in *GS* from the root **do**
29. **If** *GS*[*P*].*covered*=false,*P* can *covered* by R_{new} **then**
30. *GS*[*P*].*covered* = true;

Else

31. *GS*[*top*].*covered* = true;
32. Pop *GS*[*top*] from *GS*;

2.5 Implementasi RP-GD

Impementasi dilakukan untuk mengetahui hasil ringkasan basisdata graf. Setelah algoritma RP-GD diimplementasikan kepada ketiga graf dataset molekuler ikatan kimia, dihasilkan tiga buah ringkasan. Lalu akan dilakukan perbandingan antara *Node* dengan *Super Node* dan *Edge* dan *Super Edge*. Perbandingan dilakukan dan dianalisis hasilnya.

Penggunaan RP-GD dalam peringkasan basisdata graf dapat menghasilkan ringkasan jumlah *Nodes* dan *Edges* dalam suatu graf dan dataset secara signifikan. Hal ini dikarenakan setiap graf dicari hanya yang paling sering muncul dan terdekat serta menghilangkan yang tidak sering dan dekat dengan pola ringksan.

$$Persentase\ Peringkasan\ Node = \frac{Jumlah\ node\ awal - Jumlah\ node\ ringkasan}{Jumlah\ node\ awal}$$

$$Persentase\ Peringkasan\ Edge = \frac{Jumlah\ edge\ awal - Jumlah\ edge\ ringkasan}{Jumlah\ edge\ awal}$$

2.6 Rasio Peringkasan

Dalam pembentukan ringkasan basisdata graf, tingkat peringkasan juga dapat dihitung dengan membandingkan jumlah graf dari basisdata graf sebelum dan setelah dilakukan peringkasan. Hal ini akan dihitung menggunakan rumus:

$$Persentase\ Peringkasan\ Graf = \frac{Jumlah\ graf\ pada\ dataset\ awal - Jumlah\ graf\ pada\ dataset\ hasil\ ringkasan}{Jumlah\ pada\ dataset\ awal}$$

2.7 Lama Waktu Kinerja Peringkasan Graf

Kinerja algoritma peringkasan dataset ini diuji dengan menghitung rata-rata lama waktu proses dari lima kali pengujian terhadap masing-masing dataset. Hasil pengukuran waktu ini kemudian digunakan untuk membandingkan performa graph.

2.8 Pengecekan Cakupan Informasi

Pengecekan cakupan informasi akan didapatkan hasilnya dengan melakukan perbandingan cakupan *support* pada setiap graf di dataset ringkasan terhadap *database* awal. Dihitung berdasarkan rata-rata dari tiap graf hasil ringkasan yang *support* dengan dataset awal, kemudian dibuat perhitungannya. Hal ini akan digunakan untuk membuktikan bahwa hasil kompresi basisdata tetap mencakup informasi dari dataset awal, dengan *minimum support* yang bervariasi. Pengecekan dilakukan pada tiap minimum support dan pada tiap dataset dengan rumus sebagai berikut:

$$|Cakupan\ informasi| = \frac{total\ jumlah\ dari\ absolut\ support}{jumlah\ graf\ pada\ ringkasan\ graf}$$

3. Perancangan Sistem

Penganalisaan basisdata graf ini dilakukan untuk mengetahui kualitas dari salah satu algoritma peringkasan graf, yaitu RP-GD dalam hal melihat dari parameter yang telah penulis tentukan. Rasio peringkasan, lama waktu pemrosesan dan cakupan informasi menjadi nilai untuk menganalisa kualitas keluaran berupa ringkasan dari basisdata graf tersebut. Gambaran yang dapat dijelaskan disini adalah dengan penganalisaan terhadap algoritma RP-GD untuk mengetehaui seberapa tinggi kualitas yang dihasilkan sebagai sebuah ringkasan dari dataset yang penulis ambil di NCI dan Cheminformatics**.** Tertuang dalam beberapa poin :

1. Dataset diambil dari *http://cactus.nci.nih.gov/download/nci/* oleh *Computer-Aided Drug Design (CADD) at National Cancer Institute (NCI)* dan dataset dari http://cheminformatics.org/datasets/
2. Dilakukan proses *graph mining*
3. Menemukan frequent graph pattern yang dapat dijadikan calon ringkasan graf
4. Meringkaskan pola graf tersebut dengan menggunakan algoritma RP-GD
5. Kemudian dihasilkan sebuah ringkasan graf
6. Analisa ringkasan graf tersebut dengan dua parameter pengukuran kualitas peringkasan graf
7. Penilaian ini diuji dengan dataset yang nyata
8. Proses mengambil kesimpulan serta rekomendasi dilaksanakan

Gambar 4 Gambaran Umum Sistem

Dari gambaran umum sistem di atas maka pemrosesan yang dilakukan adalah :

1. Sistem menggunakan input dari dataset NCI dan Cheminformatics yang kemudian disimpan dalam bentuk graph.
2. Sistem melakukan transformasi ke basisdata peringkasan graf menggunakan Algoritma RP-GD.
3. Melakukan pengujian terhadap hasil transformasi dari basisdata graf ke basisdata peringkasan graf:
4. Melakukan analisis terhadap hasil pengujian

Setelah mendapatkan output basisdata graf awal, maka akan dilakukan proses analisis terhadap hasil dari basisdata graf awal dan basisdata ringkasan graf yaitu mengukur performansi.

a. Setelah mendapatkan output basisdata peringkasan graf, maka akan dilakukan proses analisis terhadap hasil dari basisdata peringkasan graf yaitu mengukur performansi dan mengukur hasil ringkasan.

b. Untuk uji performansi basisdata graf merupakan pengujian berdasarkan hasil implementasi Algoritma RP-GD, rasio peringkasan nodes dan edges, performansi waktu lama pemrosesan, dan pengecekan cakupan informasi.

c. Setelah dilakukan pengujian, maka akan didapatkan hasil uji. Hasil uji ini nantinya dapat memberikan gambaran performansi dari peringkasan basisdata graf.

4. Pengujian Sistem

Pada dataset yang digunakan, Super Edge terbentuk secara pencarian Greedy. Setelah mendapatkan data Super Node maka akan dilakukan pencarian pasangan dalam Super Node tersebut. Terdapat pasangan yang merepresentasikan ringkasan dari node-node tersebut. Berikut merupakan contoh hasil pencarian peringkasan Super Edge :

Dataset awal

```
a,0,[C]C1=[C]C(=O)[C]=[C]C1=O
b,0,S(SC2=NC1=[C][C]=[C][C]=C1[S]2)C4=NC3=[C][C]=[C][C]=C3[S]4
c,0,[N+](=O)([O-])C1=[C]C(=[C]C(=C1[O])C1)[N+](=O)[O-]
```

Gambar 5 Sebagian Dataset Awal

```
1,Cl-C,2,1,3,11.538462,0,0.0
2,S(-C)-C,3,2,3,11.538462,0,0.0
3,N(-O)(-C)=O,4,3,3,11.538462,0,0.0
```

Gambar 6 Hasil Peringkasan Dataset

```
id:list
1:c,g,m
2:b,d,s
3:c,d,h
```

Gambar 7 Kamus Identifikasi

Ekstensi File yang akan dibentuk menjadi basisdata graf adalah file .smi. Dalam file input seperti gambar diatas akan diinputkan data dimana kolom pertama merepresentasikan id molekul sedangkan kolom kedua merepresentasikan bobot fokus dari molekul tersebut terhadap database dan kolom ketiga merepresentasikan molekul tersebut. Edge yang dibentuk direpresentasikan sebagai baris dalam file tersebut. Setelah data diinputkan, kemudian akan dilakukan proses pengecekan setiap kondisi yang diperlukan lalu proses data mining terhadap dataset tersebut. File output berjumlah dua file, yaitu file hasil peringkasan dan file kamus.

Pada file hasil peringkasan terbagi menjadi 8 kolom. Kolom pertama adalah id integer subgraf sebagai bentuk Super Node dan kolom kedua adalah isi Node dari Super Node. Kolom kedua menunjukkan peringkasan subgraf diawal tadi dengan menuliskan support nodes pada kolom ketiga dan support edges pada kolom keempat. Support nodes adalah jumlah nodes yang dapat diringkas sehingga menjadi deskripsi molekul pada kolom kedua, sedangkan support edges adalah jumlah edges yang dapat terwakili dengan nodes pada kolom kedua. Kemudian file tersebut berisi absolute support pada kolom kelima dan juga relative support pada kolom keenam. Absolute support sendiri merupakan jumlah molekul yang dapat terwakili oleh isi nodes yang sudah diringkas, yaitu berupa subgraf pada kolom kedua yang merepresentasikan molekul-molekul pada dataset diawal (input), sedangkan untuk relative support yaitu persentase antara absolute support dengan jumlah total dataset molekul diawal.

Kolom ketujuh dan kedelapan berisikan absolute support dan relative support untuk pelengkap database, namun karena kasus ini penulis tidak membagi database menjadi fokus dan pelengkap, maka data input diawal penulis isi 0 sehingga kolom ketujuh dan kedelapan pada file output berisi 0 juga. Hasil output ditulis kedalam format file berekstensi sub.

Kemudian file lain yaitu file identifikasi pembanding antara hasil output dengan input ditulis dalam file berekstensi ids. Isi dari file tersebut adalah untuk setiap subgraf hasil peringkasan ditampilkan daftar dari molekul mana saja yang direpresentasikan oleh ringkasan tersebut sesuai dengan id identifikasi pada kedua file dataset maupun file hasil peringkasan. Kolom pertama berupa id dari subgraf yang telah diringkas berupa bilangan bulat dari 1 sampai dengan jumlah subgraf yang ditemukan. Kolom kedua berisikan array molekul mana saja sesuai id pada database input yang telah direpresentasikan oleh subgraf pada kolom pertama.

4.1 Tujuan Pengujian

Tujuan pengujian yang dilakukan pada Tugas Akhir ini adalah

1. Untuk melakukan analisis terhadap hasil implementasi Algoritma RP-GD dalam studi kasus molekuler ikatan kimia dilihat dari jumlah nodes dan edges pada setiap basisdata graf dengan minimum support yang bervariasi.
2. Untuk melakukan analisis terhadap hasil rasio yaitu membandingkan jumlah ikatan molekul kimia file dataset sebelum dan sesudah dilakukan peringkasan.
3. Untuk melakukan analisis terhadap lama waktu proses peringkasan basisdata graf .
4. Untuk melakukan analisis terhadap hasil pengecekan cakupan informasi dengan membandingkan antara hasil sumarisasi serta kamusnya dengan dataset awal.

4.2 Dataset

Data graf yang digunakan dalam penelitian ini berasal dari website CADD (Computer-Aided Drug Design) Group di NCI (National Cancer Institute) sebanyak satu dataset, lalu juga dari website organisasi non komersial *cheminformatics* sebanyak dua dataset. Terdapat tiga graph yang masing-masing berisi molekuler dengan format SMILES, yang mempunyai karakteristik berbeda-beda. Penulis tampilkan spesifikasi dataset *input* kedalam tabel berikut ini :

Tabel 1 Spesifiasi File Input

File Input	Jumlah Molekul	Ukuran (kilobytes)
NCISMA99.smi	95995	6420
Karthikeyan Melting Point Dataset.smi	4449	217
Huuskonen Data Set.smi	1312	37,8

Dapat dilihat bahwa ketiga graf memiliki kombinasi jumlah molekul dan atom yang bervariasi. Sistem ini diharapkan dapat diimplementasikan terhadap graf dengan jumlah *Node* dan *Edge* yang cukup besar.

4.3 Analisis Implementasi Algoritma RP-GD dalam Studi Kasus Molekuler Ikatan Kimia

Setelah RP-GD diimplementasikan kepada ketiga graph input, dihasilkan tiga buah file yang

berisikan graf yang telah disumarisasikan sesuai dataset inputan awal. Graf-graf pada dataset NCISMA99.smi menghasilkan NCISMA99.sub sebagai file peringkasan grafnya. Kemudian dataset Karthikeyan Melting Point Dataset.smi menghasilkan Karthikeyan Melting Point Dataset.sub sebagai file peringkasan grafnya, serta dataset Huuskonen Data Set.smi menghasilkan Huuskonen Data Set.sub sebagai file peringkasan grafnya. Dihasilkan juga kamus berisi semua identifikasi hasil dari peringkasannya pada file berekstensi ids (.ids) sesuai masing-masing dataset.

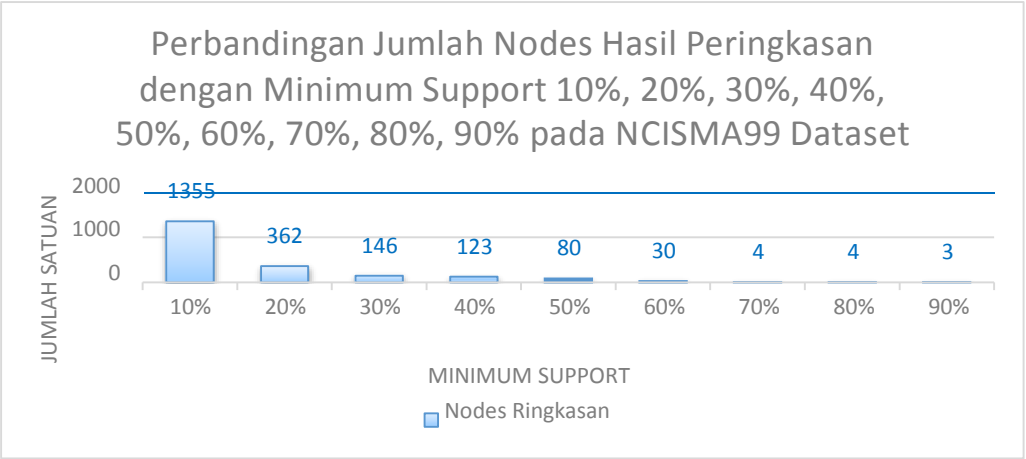
Dalam penerapan RP-GD terlihat bahwa jumlah graf dari tiap dataset sangat berkurang karena membentuk ringkasan graf. Hal ini dikarenakan peringkasan memang telah berhasil dilakukan kepada graf-graf inputan terebut. Dari pengujian dengan nilai *minimum support* yang bervariasi mendapatkan hasil yang bervariasi juga.

Dari percobaan terhadap ketiga dataset, algoritma ini dapat dapat mengurangi jumlah molekul graf pada dataset NCISMA99 dengan syarat *minimum support* 10% diringkas sebesar 99,74%. Jika dengan syarat *minimum support* sebesar 20% dihasilkan ringkasan 99,91%. Jika dengan syarat *minimum support* sebesar 30% dihasilkan ringkasan 99,96%. Jika dengan syarat *minimum support* sebesar 50% dihasilkan ringkasan 99,97%. Berikut merupakan hasil implementasi algoritma RP-GD dan jumlah hasil ringkasan graf dengan masing-masing dataset dalam bentuk Tabel.

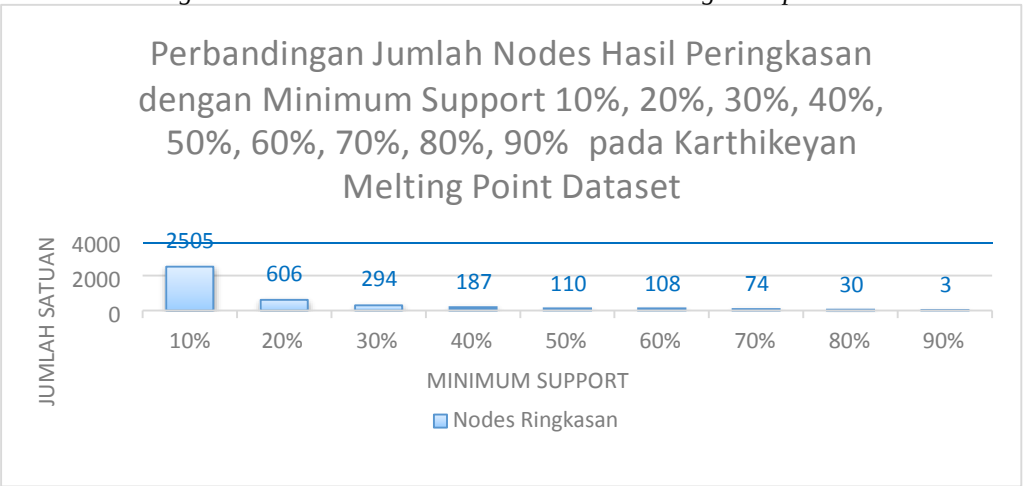
Tabel 2 Hasil Implementasi dan Persentase Peringkasan Nodes dan Edges

File Dataset Graph	Minimum Support (%)	Dataset Awal		Ringkasan Graf		Persentase Peringkasan (%)	
		Node	Edge	Node	Edge	Node	Edge
NCISMA99	10	1837735	1930000	1355	1122	99,926	99,942
	20			362	288	99,980	99,985
	30			146	111	99,992	99,994
	40			123	94	99,993	99,995
	50			80	59	99,996	99,997
	60			30	19	99,998	99,999
	70			4	1	100	100
	80			4	1	100	100
	90			3	1	100	100
Karthikeyan Melting Point Dataset	10	100579	108164	2505	2124	97,509	98,036
	20			606	492	99,397	99,545
	30			294	236	99,708	99,782
	40			187	148	99,814	99,863
	50			110	85	99,891	99,921
	60			108	84	99,922	99,922
	70			74	56	99,948	99,948
	80			30	22	99,980	99,980
	90			3	1	99,999	99,999
Huuskonen Data Set	10	17167	17674	497	397	97,105	97,754
	20			139	104	99,190	99,412
	30			111	85	99,353	99,519
	40			32	21	99,814	99,881
	50			27	19	99,892	99,843
	60			4	1	99,994	99,977
	70			3	1	99,994	99,983
	80			3	1	99,994	99,983
	90			1	0	100	99,994

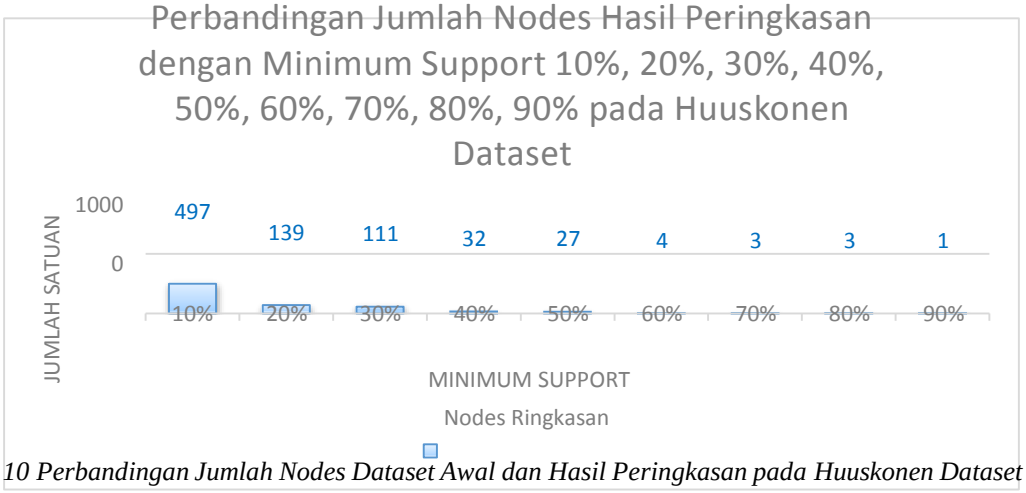
Dalam perbandingan jumlah *Nodes* dan *Nodes* yang sudah diringkas dibuat grafik perbandingan. Grafik perbandingan membandingkan antara masing-masing jumlah *Node*s dan *Edges* antara dataset awal dengan hasil peringkasan pada ketiga dataset diatas. Berikut ini visualisasi diagramnya.



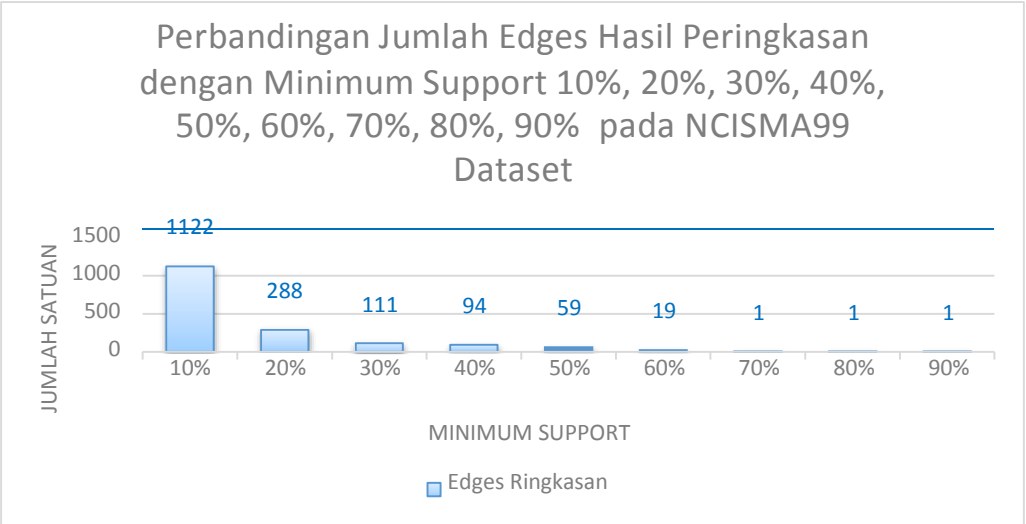
Gambar 8 Perbandingan Jumlah Nodes Dataset Awal dan Hasil Peringkasan pada NCISMA99 Dataset



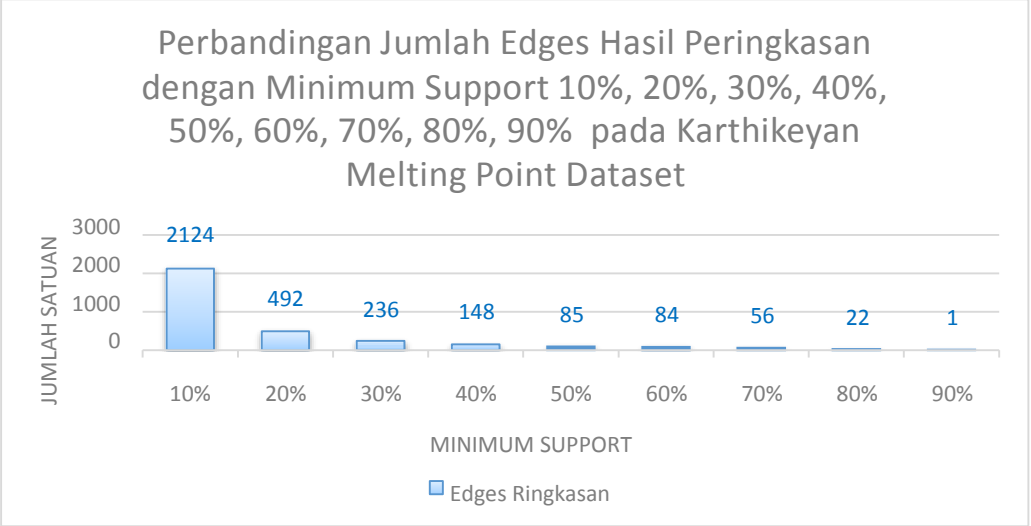
Gambar 9 Perbandingan Jumlah Nodes Dataset Awal dan Hasil Peringkasan pada Karthikeyan Melting Point Dataset



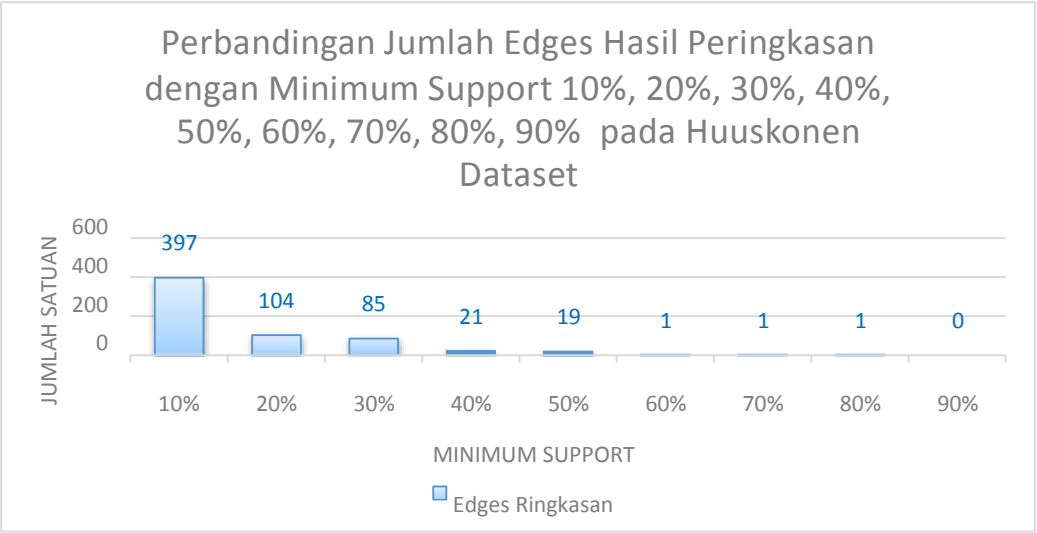
Gambar 10 Perbandingan Jumlah Nodes Dataset Awal dan Hasil Peringkasan pada Huuskonen Dataset



Gambar 11 Perbandingan Jumlah Edges Dataset Awal dan Hasil Peringkasan pada NCISMA99 Dataset



Gambar 12 Perbandingan Jumlah Edges Dataset Awal dan Hasil Peringkasan pada Karthikeyan Melting Point Dataset



Gambar 13 Perbandingan Jumlah Edges Dataset Awal dan Hasil Peringkasan pada Huuskonen Dataset

Dapat dilihat bahwa hasil peringkasan berdasarkan jumlah *nodes* maupun *edges* sangat tinggi perbandingannya. Sehingga tujuan untuk meringkas *nodes* dan *edges* menjadi super node dan super edge berfungsi. Masing-masing hasil peringkasan dataset tersebut dituliskan ke dalam file berekstensi sub (.sub) dan

juga ids (.ids). File sub berisi hasil peringkasan basisdata graf , sedangkan file ids berisi identifikasi kamus untuk sinkronisasi antara file dataset awal dan file hasil peringkasan.

4.4 Analisis Rasio Peringkasan

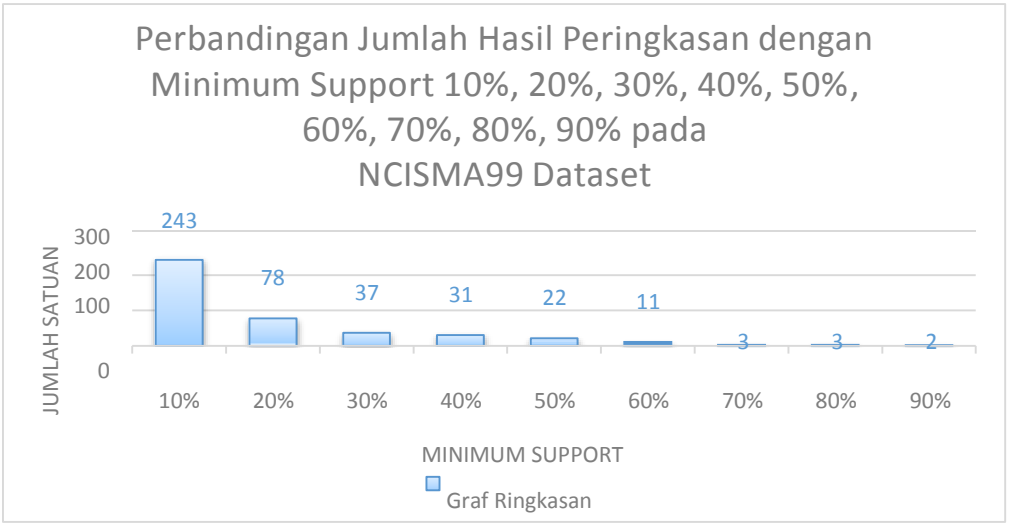
Jumlah molekul dari ketiga dataset ini berbeda-beda. Jumlah molekul terbesar terdapat pada dataset NCISMA99 sebanyak 95.995 molekul kimia, yang dengan kata lain sebanyak 95.995 graf. Peringkasan terhadap dataset tersebut menghasilkan output yang berbeda-beda sesuai dengan parameter *minimum support* (minsup) yang diberikan. Dengan minsup sebesar 10% dihasilkan ringkasan graf menjadi sebanyak 243 graf, sehingga rasio yang dihasilkan sebesar 99,747%. Kemudian dengan minsup 20% dihasilkan ringkasan graf menjadi sebanyak 78 graf dan rasio yang dihasilkan sebesar 99,919%. Kemudian dengan minsup 30% dihasilkan ringkasan graf menjadi sebanyak 37 graf dan rasio yang dihasilkan sebesar 99,961%. Kemudian dengan minsup 40% dihasilkan ringkasan graf menjadi sebanyak 31 graf dan rasio yang dihasilkan sebesar 99,968%. Kemudian dengan minsup 50% dihasilkan ringkasan graf menjadi sebanyak 22 graf dan rasio yang dihasilkan sebesar 99,977%.

Sedangkan untuk dataset Karthikeyan Melting Point yang memiliki jumlah molekul atau graf awal sebanyak 4449 graf tercatat hasil berikut ini. Dengan minsup sebesar 10% dihasilkan ringkasan graf menjadi sebanyak 406 graf dan

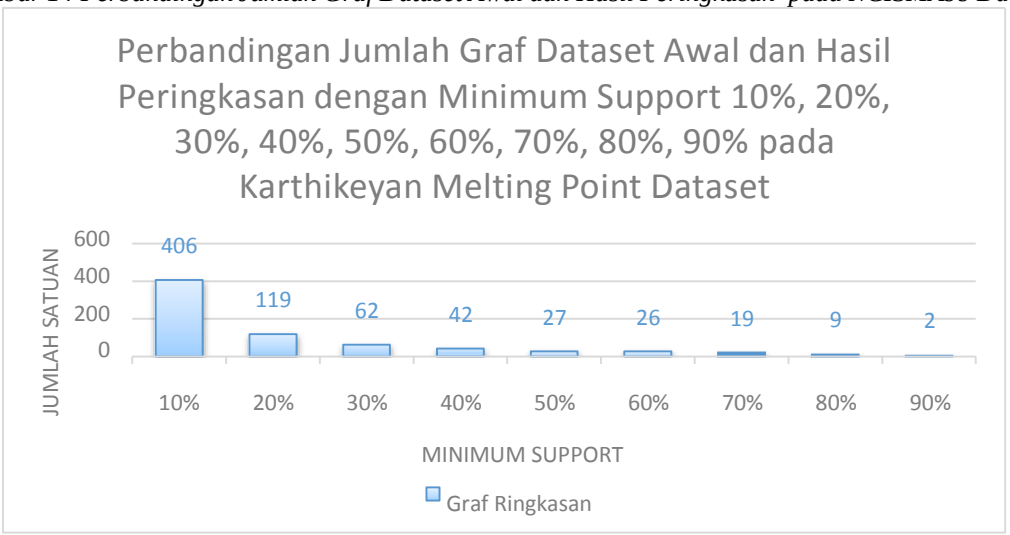
Tabel 3 Hasil Pengujian Rasio Peringkasan Graf

File Dataset Graph	Minimum Support (%)	Jumlah Graf Awal	Jumlah Graf Teringkas	Rasio Peringkasan Graf (%)
NCISMA99	10	95.995	243	99,747
	20		78	99,919
	30		37	99,961
	40		31	99,968
	50		22	99,977
	60		11	99,989
	70		3	99,997
	80		3	99,997
	90		2	99,998
Karthikeyan Melting Point Dataset	10	4.449	406	90,874
	20		119	97,325
	30		62	98,606
	40		42	99,056
	50		27	99,393
	60		26	99,416
	70		19	99,573
	80		9	99,798
	90		2	99,955
Huuskonen Dataset	10	1.312	106	91,921
	20		37	97,180
	30		28	97,866
	40		12	99,085
	50		9	99,314
	60		3	99,771
	70		2	99,848
	80		2	99,848
	90		1	99,924

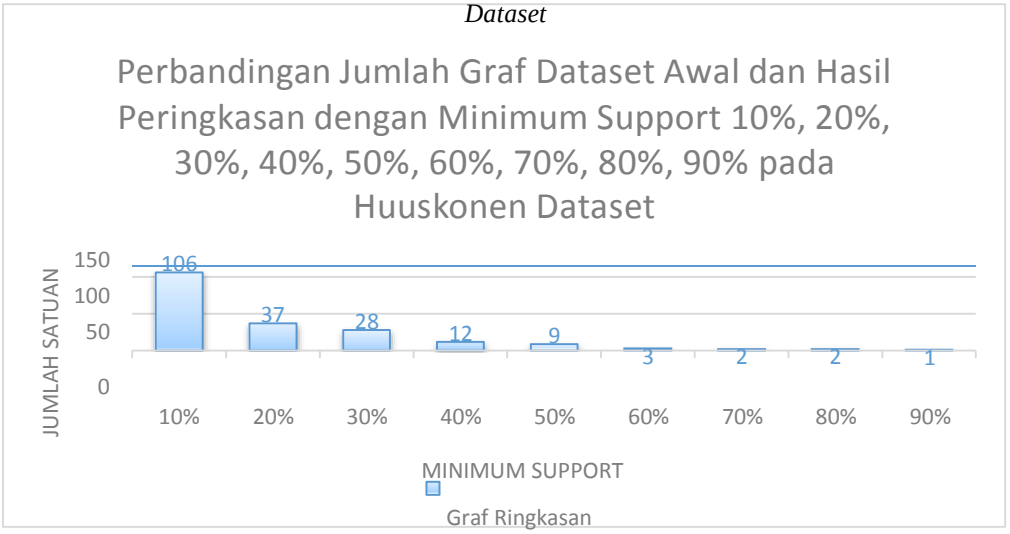
Bentuk visualisasi dari perbandingan Rasio Peringkasan Graf akan dibuat secara grafik. Grafik menunjukkan perbandingan hasil peringkasan dari setiap dataset dan *minimum support* yang diberikan. Berikut ini merupakan bentuk diagramnya.



Gambar 14 Perbandingan Jumlah Graf Dataset Awal dan Hasil Peringkasan pada NCISMA99 Dataset



Gambar 15 Perbandingan Jumlah Graf Dataset Awal dan Hasil Peringkasan pada Karthikeyan Melting Point Dataset



Gambar 16 Perbandingan Jumlah Graf Dataset Awal dan Hasil Peringkasan pada Huuskonen Dataset

4.5 Analisis Lama Waktu Proses Peringkasan Basisdata Graf

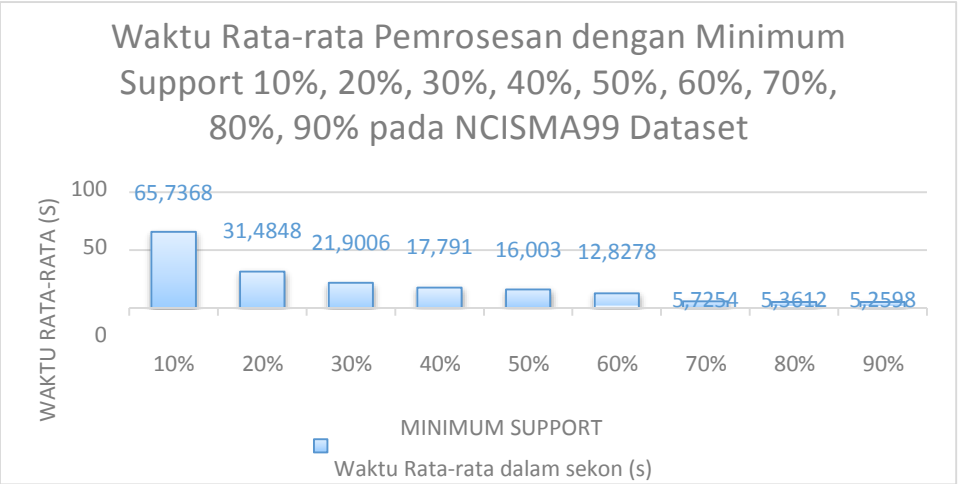
Pengujian dilakukan terhadap basisdata awal dengan menjalankan program lalu menghitung waktu pemrosesan sampai selesai proses pencarian ringkasan graf tersebut. Proses dijalankan pada tiap dataset dengan minimum support yang bervariasi. Dengan pencatatan waktu pemrosesan sebanyak 5

kali proses peringkasan graf, lalu diacatat lama waktu rata-ratanya.

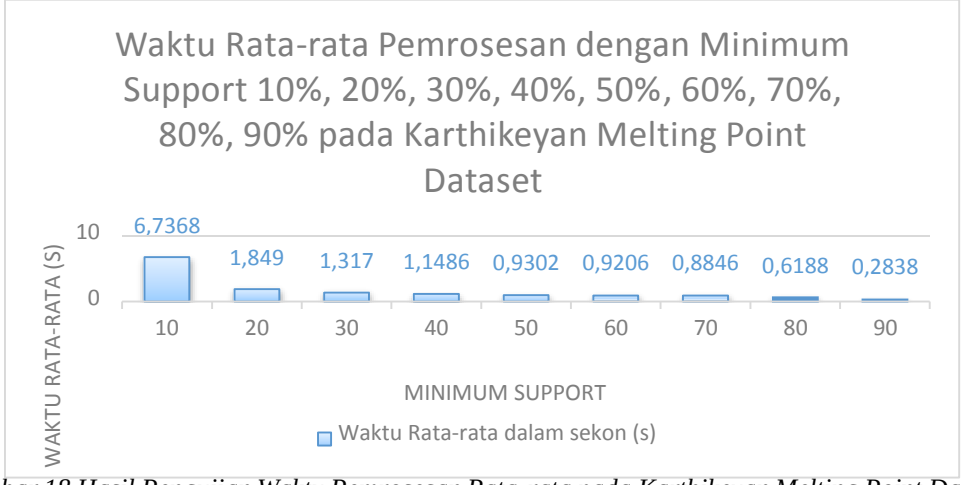
Hasil dari tiap pengujian mempunyai rata-rata waktu yang berbeda-beda dan dalam dataset dengan *minimum support* sebesar 10% mempunyai waktu yang paling lama untuk pemrosesannya. Sedangkan pada dataset dengan *minimum support* sebesar 50% waktu pemrosesannya paling sedikit. Hasilnya dapat dilihat pada tabel dan diagram dibawah ini.

Tabel 4 Hasil Pengujian Waktu Pemrosesan Rata-rata

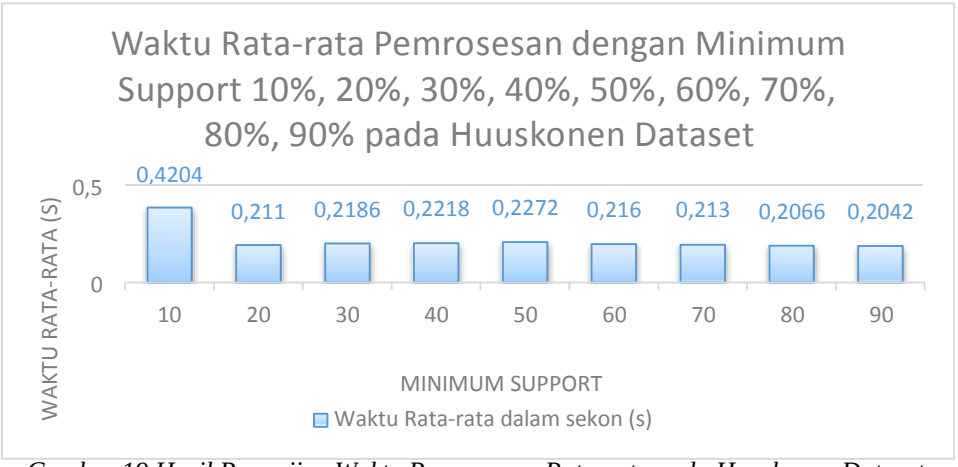
File Dataset Graph	Minimum Support (%)	Waktu Rata-rata (s)
NCISMA99	10	65,7368
	20	31,4848
	30	21,9006
	40	17,791
	50	16,003
	60	12,8278
	70	5,7254
	80	5,3612
	90	5,2598
Karthikeyan Melting Point Dataset	10	6,7368
	20	1,849
	30	1,317
	40	1,1486
	50	0,9302
	60	0,9206
	70	0,8846
	80	0,6188
	90	0,2838
Huuskonen Dataset	10	0,4204
	20	0,211
	30	0,2186
	40	0,2218
	50	0,2272
	60	0,216
	70	0,213
	80	0,2066
	90	0,2042



Gambar 17 Hasil Pengujian Waktu Pemrosesan Rata-rata pada NCISMA99 Dataset



Gambar 18 Hasil Pengujian Waktu Pemrosesan Rata-rata pada Karthikeyan Melting Point Dataset



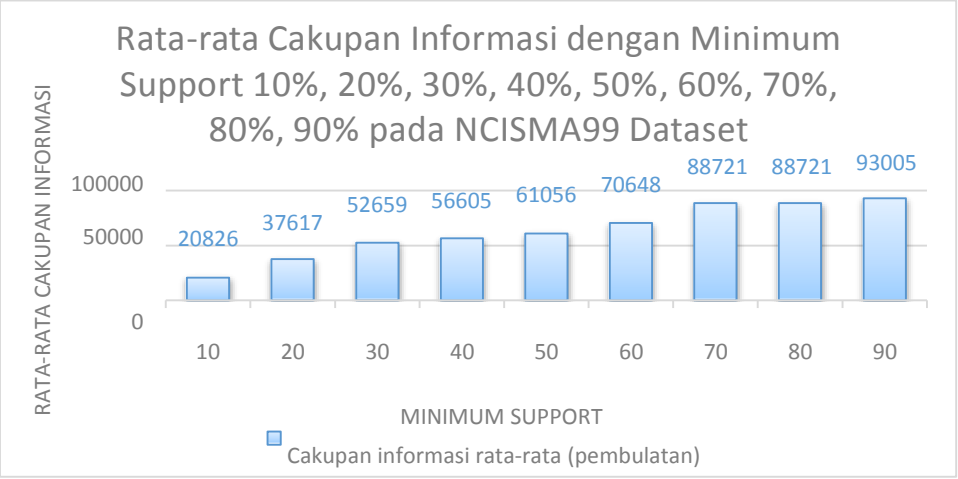
Gambar 19 Hasil Pengujian Waktu Pemrosesan Rata-rata pada Huuskonen Dataset

4.6 Analisis Cakupan Informasi Peringkasan

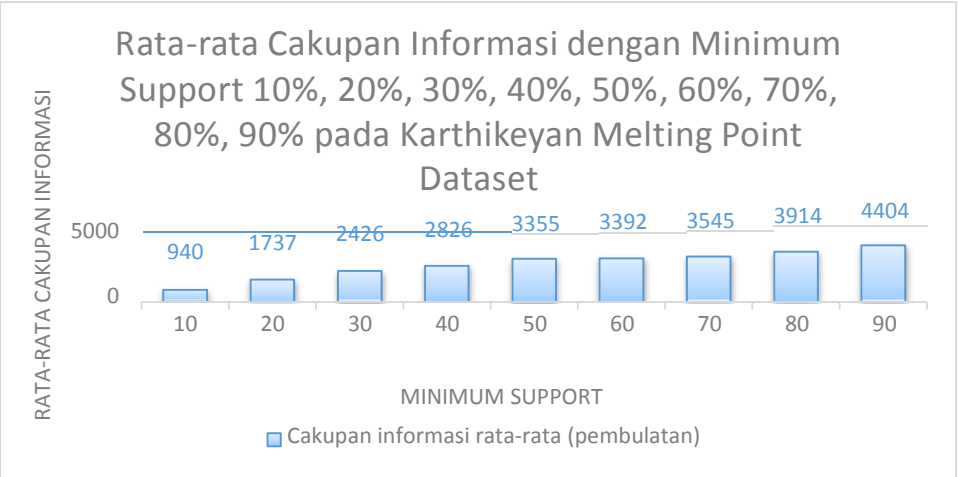
Analisis untuk dapat mengetahui seberapa besar cakupan informasi yang diberikan oleh ringkasan dapat dilihat dengan cara menghitung rata-rata dari jumlah dari setiap graf hasil peringkasan yang mendukung informasi pada graf dataset awal di setiap minimum support. Hal ini dilakukan sebagai salah satu analisis tingkat cakupan informasi dari graf-graf pada hasil peringkasan dibandingkan dengan graf-graf dataset awal dilihat dari jumlah graf yang dicakup oleh sebuah graf tersebut, dari graf ringkasan pertama sampai dengan graf ringkasan terakhir lalu dibuat rata-ratanya. Hasil pencatatan dapat dilihat dari tabel dan diagram dibwah ini.

Tabel 5 Hasil Peringkasan Graf berupa Cakupan Informasi Rata-rata

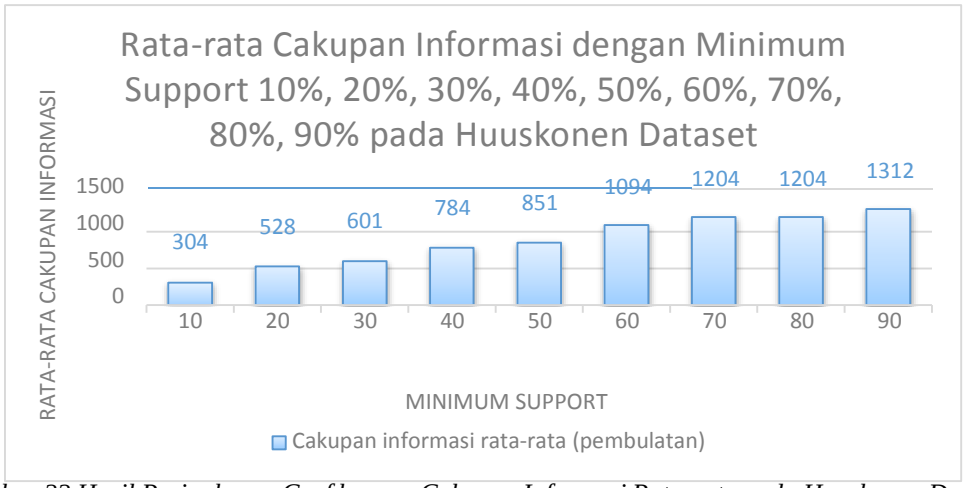
File Dataset Graph	Minimum Support (%)	Cakupan informasi rata-rata (pembulatan)
NCISMA99	10	20826
	20	37617
	30	52659
	40	56605
	50	61056
	60	70648
	70	88721
	80	88721
	90	93005
Karthikeyan Melting Point Dataset	10	940
	20	1737
	30	2426
	40	2826
	50	3355
	60	3392
	70	3545
	80	3914
	90	4404
Huuskonen Dataset	10	304
	20	528
	30	601
	40	784
	50	851
	60	1094
	70	1204
	80	1204
	90	1312



Gambar 20 Hasil Peringkasan Graf berupa Cakupan Informasi Rata-rata pada NCISMA99 Dataset



Gambar 21 Hasil Peringkasan Graf berupa Cakupan Informasi Rata-rata pada Karthikeyan Melting Point Dataset



Gambar 22 Hasil Peringkasan Graf berupa Cakupan Informasi Rata-rata pada Huuskonen Dataset

4.7 Analisis Pengaruh Korelasi Antar Parameter Pengujian

Analisis korelasi antar parameter disini yaitu dengan memberi nilai *minimum support* yang rendah maka dihasilkan sebagai berikut :

- Jumlah nodes dataset yang tinggi.
- Jumlah edges dataset yang tinggi.
- Jumlah subgraf hasil ringkasan yang tinggi.
- Rata-rata waktu pemrosesan aplikasi yang tinggi.
- Nilai rata-rata cakupan informasi jumlah yang rendah.

Untuk minimum support yang tinggi dihasilkan analisa sebagai berikut :

- Jumlah nodes dataset yang rendah.
- Jumlah edges dataset yang rendah.
- Jumlah subgraf hasil ringkasan yang rendah.
- Rata-rata waktu pemrosesan aplikasi yang rendah.
- Nilai rata-rata cakupan informasi jumlah yang tinggi.

Sehingga dari hasil diatas dapat dianalisis bahwa nilai *minimum support* berbanding terbalik dengan jumlah nodes ringkasan dataset, jumlah edges ringkasan dataset, jumlah subgraf ringkasan dataset, dan rata-rata waktu pemrosesan aplikasi. Serta nilai *minimum support* tersebut berbanding lurus dengan nilai rata-rata cakupan informasi hasil peringkasan.

5. Penutup

5.1 Kesimpulan

Berdasarkan analisis terhadap pengujian yang dilakukan dalam Tugas Akhir ini, dapat disimpulkan bahwa:

1. Tiga buah dataset yang berisikan banyak molekul/graf yang direpresentasikan oleh basisdata graf dan melakukan peringkasan basisdata graf dari sebuah metode Algoritma RP-GD dapat menghasilkan kualitas ringkasan graf yang tinggi.
2. Menurut hasil analisis dapat disimpulkan :
   - Analisis pengujian menunjukkan bahwa dataset basisdata graf dengan nilai minimum support sebesar 90% dapat diringkas maksimal sebesar 99,998% dan menghasilkan nilai rata-rata cakupan support sebesar 93005.
   - Dengan algoritma RP-GD, maka kualitas peringkasan basisdata graf menjadi lebih efektif dilihat dari jumlah graf, nodes, dan edges yang teringkas.
   - Nilai *minimum support* berbanding lurus dengan nilai rata-rata cakupan informasi.
   - Nilai *minimum support* berbanding terbalik dengan jumlah nodes ringkasan dataset, jumlah edges ringkasan dataset, jumlah subgraf ringkasan dataset, dan rata-rata waktu pemrosesan aplikasi
3. Rekomendasi dari sudut pandang *Computer Science* untuk dataset NCI dan Cheminformatics diringkas lebih padat agar lebih efektif dan efisien untuk pemrosesan dataset molekul-molekul yang ditampilkan sesuai dengan *minimum support.*

5.2 Saran

Saran yang diperlukan dari Tugas Akhir ini untuk pengembangan sistem lebih lanjut adalah sebagai berikut :

1. Penggunaan dataset pada penelitian berikutnya menggunakan dataset graf yang berarah.
2. Hasil yang diperoleh mungkin dapat ditampilkan dalam bentuk visualisasi graf yang lebih nyata dan menarik.

6. Daftar Pustaka

[1] Vicknair et al., 2010. A Comparison of a Graph Database and a Relational Database, ACMSE.

[2] I. Robinson, J. Webber, and E. Eifrem, “Graph Databases”, O’Reilly Media Inc., ISBN : 978-1-449-35626-2, USA, 2003

[3] R. Saint-Paul, G. Raschia, N. Mouaddib, “General Purpose Database Summarization”, LINA – Polytech’Nantes, ATLAS-GRIM Group, France, 2005.

[4] D. Dong and F. Liu, “Graph Database Indexing”, University of Illinois Urbana Champaign, USA

[5] Sujata J. Suryawanshi, and Prof. Mrs. S. M. Kamalapur, “Algorithms for Frequent Subgraph Mining”, Computer Engineering Department, KKWECOE, Nashik, 2013

[6] [Online]. Available: http://en.wikipedia.org/wiki/Simplified_molecular-input_line-entry_system. [Diakses 1 Juni 2015].

[7] Yuanyuan Tian, Richard A, Hankins, Jignesh M. Palel, “Efficient Aggregation for Graph Summarization”, University of Michigan, Nokia Research Center, University of Michigan. [8] TRIKI, Amel, Yann POLLET, and Mohamed BEN AHMED. "User Focused Database Summarization Approach." Current Trends in Information Technology (CTIT), 2009 International Conference on the. IEEE, 2009.

[9] C. Borgelt, T. Meinl Full, “Perfect Extension Pruning for Frequent Graph Mining” IEEE Press, Piscataway, NJ, USA 2006