

Penerapan Homomorfisma Graf untuk Mengatasi Restriksi Jumlah Maksimum Diameter Jaringan pada Protokol Pohon Rentangan

Rizky Faramita 13515055
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13515055@std.stei.itb.ac.id

Abstract—Implementing the homomorphism property of a graph in a spanning tree protocol that contains data packets in order to build a network. The mechanism includes an algorithm that is able to generate a random new and lighter minimum spanning tree with a homomorphism property compared to its origin. This mechanism is very useful when the topology becomes rough and therefore needs special treatment knowing that the protocol has a specific maximum time to transfer the packet. Due to its restriction for the maximum network diameter, the mechanism can effectively boost the performance as it can find out a shorter path to reduce the diameter without disturbing the packet transfer process.

Keywords—algoritma, diameter jaringan, homomorfisma, protokol pohon rentangan.

I. PENDAHULUAN

Aplikasi teori graf dapat ditemui di kehidupan sehari-hari, contohnya di dalam jaringan komputer. Jaringan komputer yang terdiri atas protokol berbasis pohon rentangan atau yang biasa dikenal dengan nama *Spanning Tree Protocol* (selanjutnya akan disingkat menjadi STP) ini mengalami perkembangan yang cukup pesat. STP yang semula membutuhkan waktu relatif lama untuk menghubungkan satu komputer ke komputer lain, kini disempurnakan menjadi jauh lebih cepat demi menjawab tuntutan manusia dari masa ke masa.

Berbagai macam optimasi dilakukan untuk meningkatkan efisiensi dan efektivitas STP, mulai dari modifikasi setelan waktu pengiriman dan penerimaan informasi antar komputer hingga ke algoritma protokol pohon rentangan. Meskipun demikian, masih terdapat kekurangan pada STP, salah satunya adalah masalah restriksi diameter jaringan yang dapat membuat sistem protokol gagal menerima informasi tepat pada waktunya.

Pada dasarnya, STP merupakan pohon merentang berbobot yang dapat dibuat dengan Algoritma Prim ataupun Algoritma Kruskal dengan beberapa modifikasi untuk penyesuaian terhadap atribut jaringan komputer. Dengan begitu, STP memiliki karakteristik yang sama

seperti pohon merentang pada umumnya. Salah satu karakteristik yang dapat membantu memecahkan persoalan di atas adalah sifat homomorfisma pada graf. Tujuan dari makalah ini adalah melakukan analisis kritis terhadap implementasi homomorfisma graf pada STP untuk mengatasi masalah pengiriman informasi akibat restriksi diameter jaringan.

II. DASAR TEORI

A. Graf

Sebuah graf $G = (V, E)$ merupakan suatu representasi geometri yang terdiri dari himpunan yang tidak kosong V yang disebut sebagai simpul dan E , garis penghubung simpul yang disebut sisi [1]. Di bawah ini merupakan beberapa terminologi yang relevan mengenai graf.

1. Derajat

Derajat merupakan jumlah sisi yang dimiliki oleh setiap simpul pada graf. Dalam Teori Euler dikatakan bahwa, apabila e merepresentasikan jumlah sisi dan r adalah jumlah derajat, maka nilai r yang dimiliki oleh suatu graf harus bernilai genap masing-masing simpul harus memiliki derajat yang lebih kecil dari $e-1$.

2. Tetangga

Dua buah simpul yang dihubungkan oleh satu atau lebih sisi dikatakan tetangga.

3. Diameter

Diameter merupakan jarak terpanjang antara dua buah simpul pada suatu graf. Hubungan antara diameter dengan jumlah simpul (n), sisi dengan derajat tertinggi (r), dan sisi dengan derajat kedua tertinggi (λ) dinyatakan dengan persamaan berikut [2] :

$$d \leq 2 \left\lceil \left(\frac{2r - \lambda}{4(r - \lambda)} \right) \ln(n - 1) \right\rceil \quad (1)$$

4. Upagraf

Upagraf merupakan bagian atau subgraph dari sebuah graf $G = (V, E)$ yang didefinisikan sebagai

graf $H = (W, F)$ di mana $W \subseteq V, F \subseteq E$, dan H tidak sama dengan G [1].

5. Lintasan
Lintasan merupakan himpunan sisi yang dapat menghubungkan suatu simpul ke suatu simpul lainnya.
6. Sirkuit
Sirkuit merupakan himpunan sisi yang diperoleh ketika suatu simpul melalui simpul lainnya hingga kembali ke simpul awal.
7. *Girth number*
Girth number adalah panjang minimum sebuah sirkuit pada suatu graf G .
8. *Clique number*
Clique number dari suatu graf G yang dinotasikan dengan $\omega(G)$ merupakan jumlah sisi terbanyak dari sebuah set pada graf. Hubungan antara jumlah sisi n dengan derajat dari simpul i d_i dapat dirumuskan sebagai berikut [3]-[12]:

$$\omega(G) \geq \sum_{i=1}^n \frac{1}{n - d_i}, \quad (2)$$

9. Bilangan kromatis
Bilangan kromatis dari suatu graf G yang dinotasikan dengan $\chi(G)$ atau $\gamma(G)$ merupakan nilai terkecil yang mungkin didapatkan dari pewarnaan graf G . Hubungan antara bilangan dan g dengan g adalah *girth number* dari G dapat dirumuskan sebagai berikut [3]-[12]:

$$\gamma(g) = \left\lfloor \frac{1}{2} \left(7 + \sqrt{48g + 1} \right) \right\rfloor, \quad (3)$$

Ada berbagai macam graf, berikut ini merupakan beberapa model graf yang sering kita jumpai:

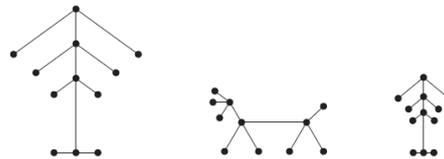
1. Graf Teratur
Graf teratur adalah graf yang setiap simpulnya memiliki derajat yang sama.
2. Graf Melingkar
Graf melingkar adalah graf yang setiap simpulnya memiliki derajat dua sehingga kemudian dapat membentuk lingkaran.
3. Graf Bipartit
Graf bipartite adalah graf yang simpul-simpulnya dapat terbagi menjadi dua himpunan di mana setiap simpul hanya dapat bertetangga dengan satu atau lebih simpul di himpunan lain.
4. Graf Lengkap
Graf lengkap adalah graf yang setiap simpulnya bertetangga dengan semua simpul lainnya. Dengan kata lain, jika n adalah jumlah simpul, maka setiap simpul memiliki $(n-1)$ derajat.
5. Graf Terhubung
Graf terhubung adalah graf yang setiap simpulnya memiliki lintasan ke simpul lainnya. Sebaliknya, jika terdapatnya satu atau lebih simpul yang tidak

memiliki lintasan ke satu atau lebih simpul lainnya maka graf tersebut dikatakan graf tak-terhubung.

6. Graf Berbobot
Graf berbobot merupakan graf yang setiap sisinya merepresentasikan nilai tertentu sehingga penting untuk menyertakan setiap sisi dengan bobot tertentu.

B. Pohon

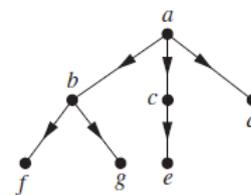
Pohon adalah graf terhubung yang tidak mengandung sirkuit dan semua simpulnya terhubung ke simpul utama [1].



Figur 1 Contoh-contoh pohon

Figur di atas merupakan beberapa graf yang tergolong ke dalam pohon. Berikut ini adalah beberapa terminologi yang relevan dengan pohon:

1. Akar
Akar dari simpul v adalah simpul unik u yang terdapat sisi untuk menghubungkan v dengan u . Tinjau figur 2 untuk memahami terminology dari akar. Simpul a adalah akar dari simpul $b, c,$ dan d . Simpul b adalah akar dari simpul f dan g , tetapi a bukan merupakan akar dari f dan g .
2. Anak
Apabila simpul u merupakan akar dari v , maka simpul v dapat dikatakan anak dari u . Pada figur 2, e merupakan anak dari c , tetapi bukan merupakan anak dari a .



Figur 2 Ilustrasi Pohon Berakar

C. Pohon Merentang Minimum

Pohon merentang G adalah upagraf dari G yang merupakan pohon yang mengandung seluruh simpul G [1]. Pohon merentang minimum dapat ditemukan pada suatu graf terhubung dan berbobot yang apabila bobot dari setiap sisi dijumlahkan maka akan menghasilkan jumlah terkecil. Untuk dapat membentuk pohon merentang minimum terdapat beberapa algoritma yang dapat digunakan, berikut ini merupakan beberapa algoritma yang sering dijumpai untuk membentuk pohon merentang minimum:

1. Algoritma Prim

Algoritma Prim dimulai dengan menentukan terlebih dahulu sisi v yang memiliki bobot terkecil. Kemudian, dengan n adalah jumlah simpul, dilakukan sebanyak $n-2$ kali proses pemilihan simpul dengan bobot terkecil yang bertetangga dengan simpul yang dihubungkan oleh v , tetapi tidak menimbulkan adanya sirkuit. Algoritma Prim adalah algoritma yang cukup efektif apabila digunakan untuk mencari pohon merentang minimum pada graf yang memiliki banyak simpul.

2. Algoritma Kruskal

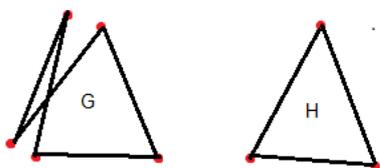
Algoritma Kruskal dimulai dengan mengurutkan seluruh sisi mulai dari yang terkecil hingga terbesar. Lalu dilakukan pemilihan sisi dari yang terkecil hingga terbesar yang tidak menimbulkan adanya sirkuit sebanyak $n-1$ kali, dengan n adalah jumlah simpul pada graf. Algoritma Kruskal adalah algoritma yang cukup efektif apabila digunakan untuk mencari pohon merentang minimum pada graf yang memiliki banyak simpul.

C. Homomorfisma

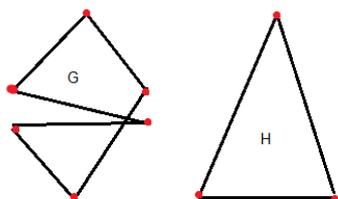
Homomorfisma dari graf G dan graf H adalah pemetaan V_G ke V_H dengan meninjau sisi-sisinya dan memanfaatkan bilangan kromatis dan *clique number* [4]-[5]. Secara matematis hubungan homomorfisma graf G dan H dapat didefinisikan melalui sebuah fungsi $f : V(G) \rightarrow V(H)$ di mana $xy \in E(G) \rightarrow f(x)f(y) \in E(H)$ dengan memperhatikan hal-hal sebagai berikut [6]-[7]:

1. $\omega(G) = r \iff r$ adalah bilangan bulat terbesar untuk $K_r \rightarrow G$, (4)
2. $\omega(G) \leq \omega(H)$ dan $\chi(G) \leq \chi(H)$. (5)

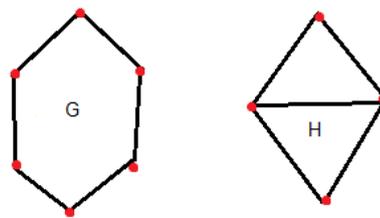
Sebagai ilustrasi, berikut ini merupakan contoh-contoh dua graf yang homomorfis:



Figur 3 Pasangan pertama graf homomorfis



Figur 4 Pasangan kedua graf homomorfis



Figur 5 Pasangan ketiga graf homomorfis

D. STP

STP merupakan sebuah protokol jaringan *layer-2* dengan topologi yang terdiri dari *bridge* di dalam sebuah *Local Area Network* (selanjutnya akan disingkat menjadi LAN) yang memanfaatkan sebuah algoritma pohon merentang yang dimodifikasi sedemikian rupa agar memori per *bridge* yang dibutuhkan untuk mentransfer data menjadi kecil [8]-[9]. LAN sendiri merupakan jaringan yang terbatas dari segi geografis, *traffic*, dan banyaknya *stations*, contohnya adalah jaringan komputer di dalam sekolah atau kampus.

Ide yang ditawarkan oleh STP ada dengan menyediakan sebuah topologi yang dapat menghubungkan seluruh LAN namun terbebas dari sirkuit mengingat hadirnya sirkuit di dalam sebuah jaringan dapat mengakibatkan penurunan performa secara drastis akibat paket yang terus menerus berputar di dalam *loop*. Berikut ini adalah beberapa terminologi yang relevan terhadap STP[8]-[9]:

1. Grup pohon merentang

Grup pohon merentang merupakan domain dari jaringan *layer-2* yang dibuat dengan memasang sebuah grup dari jaringan *layer-2* menjadi bagian dari domain jaringan *layer-2* lainnya. Di dalam domain *layer-2* banyak terdapat grup pohon merentang yang bekerja berdasarkan Algoritma STP-nya masing-masing demi menghindari adanya sirkuit.

2. Station

Station merupakan sebuah simpul di dalam LAN yang tidak mengantarkan paket data dan terhubung dengan LAN untuk berkomunikasi dengan *station* lainnya. *Station* tidak memberikan kontribusi terhadap algoritma STP.

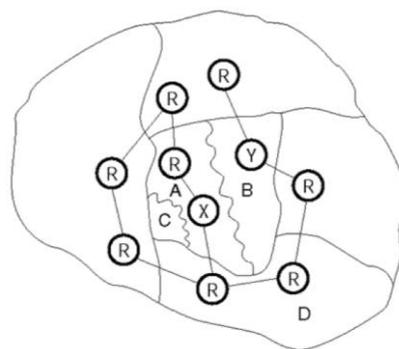
3. Bridge

Bridge merupakan simpul yang terhubung dengan dua atau lebih LAN dan berfungsi untuk mengantarkan paket data kepada LAN yang bersesuaian.

4. Prioritas *bridge*

Prioritas *bridge* merupakan besaran tertentu yang digunakan untuk menentukan *switch* mana yang akan menjadi *root*.

5. *Root*
Root merupakan *bridge* terpilih yang menjadi akar dari pohon merentang yang terbentuk. Penentuan *root* ini dilakukan berdasarkan prioritas *bridge* (dengan alatmat MAC apabila *bridge* memiliki prioritas yang sama) dan hanya akan ada satu *root* saja yang terpilih.
6. *Designated bridge*
Designated bridge merupakan *bridge* pada LAN yang terdekat dari *root*. Untuk setiap LAN hanya akan ada satu *designated bridge* saja yang terpilih secara dinamis.
7. *Bobot port*
Bobot *port* merupakan nilai yang disimpan pada setiap *switch port*. Bobot *port* ini menentukan jalur mana yang memiliki bobot terendah sehingga mempengaruhi algoritma STP secara keseluruhan. Sebuah port dengan bobot terendah akan dipilih menjadi *forwarding port* dan port sisanya berada pada *block state*.
8. *Prioritas port*
Prioritas *port* merupakan suatu nilai yang akan disimpan pada setiap *switch port*. Apabila *port* memiliki prioritas yang sama maka akan dilihat *port number* untuk menentukan *designated port* bagi sebuah segmen.
9. *Bridging Protocol Data Units* (selanjutnya akan disingkat menjadi BPDU)
BPDU merupakan paket berukuran 64-byte yang menyediakan informasi tentang *switch* tertentu dan dikirimkan oleh semua *switch* yang tergabung di dalam STP tersebut. Informasi yang terkandung di dalam BPDU meliputi prioritas *bridge*, bobot *port*, dan prioritas *port*. Informasi tersebut kemudian digunakan untuk membangun sebuah *root switch* dan *designated path*.
10. *Link*
Link merupakan penghubung antara LAN yang bertetangga dengan *bridge*. Dengan kata lain, *link* bertindak seperti sebuah sisi pada graf.
11. *Hello Message*
Hello message merupakan sebuah pesan yang dikirimkan oleh *designated bridge* kepada LAN secara berkala berdasarkan *hello time* tertentu. Isi dari *hello message* ini berupa ID *bridge*, ID *root bridge*, panjang dari jalur terpendek ke *root*, waktu semenjak informasi dari *root* dikirimkan melalui jalur tersebut, *link identifier*, dan *max age*.
12. *Max age*
Max age merupakan sebuah parameter yang lebih besar atau sama dengan *MaxPropTime* agar *root* yang sedang beroperasi tidak *timed out*.



Figur 6 Sub-jaringan computer [10]

E. Cara Kerja STP

Cara kerja STP akan dijabarkan melalui poin-poin di bawah ini [8]-[11]:

1. *Switch* akan secara otomatis membaca identitas setiap *port*, menentukan parameter untuk pohon merentang dan *switch* (bisa memilih untuk mengikuti *default parameter* atau memberi parameter baru).
2. Setelah pembacaan identitas, maka didapatkan pohon merentang minimum beserta jalur alternatifnya yaitu jalur yang berada pada *blocked state*.
3. Ketika sebuah jalur *down*, maka sistem akan secara otomatis mengaktifkan jalur yang berada pada *blocked state* hingga jalur utama pulih.

F. Masalah Diameter Jaringan pada STP

Topologi STP dapat kehilangan stabilitasnya. Beberapa faktor yang memengaruhi stabilitas topologi STP, diantaranya adalah parameter *max age*, *forward delay*, dan diameter jaringan *bridge* [12]. Nilai *default* dari STP *timers* memperlihatkan nilai maksimum untuk diameter *bridge*. Dengan meletakkan nilai tujuh pada *default STP timers* artinya diameter maksimum yang dapat dimiliki oleh jaringan *bridge* adalah tujuh.

Setiap kali BPDU berjalan dari *root* ke ujung *bridge*, *age field* akan bertambah. Ketika *age field* telah melebihi *max age* maka *bridge* akan menghiraukan BPDU tersebut. Ketika sebuah *root* terletak cukup jauh dari *bridge* yang mana hal ini sangat mungkin terjadi di dalam jaringan yang cukup rumit, maka permasalahan semacam ini memiliki peluang yang besar untuk terjadi.

Sayangnya, mengubah STP *timers* dapat berdampak buruk terhadap stabilitas STP secara substansial dan tentunya berpengaruh pula terhadap diameter dari jaringan. Maka dari itu, dibutuhkan suatu mekanisme yang dapat menangani kasus ketika topologi jaringan membutuhkan diameter yang cukup besar.

III. PEMBAHASAN

A. Fisibilitas Implementasi Homomorfisma Graf pada STP secara Garis Besar

Ketika sebuah jaringan *bridge* membutuhkan diameter

yang cukup besar, namun terhambat akibat *max age* yang tidak mendukung, maka dibutuhkan sebuah mekanisme yang dapat memotong jalur yang panjang tersebut agar diameter yang dibutuhkan oleh *bridge* berkurang. Restriksi terhadap jumlah maksimum diameter ini harus ditangani dengan mekanisme khusus karena opsi lain seperti memodifikasi *max age* dapat mengganggu proses pengiriman informasi di tempat lain.

Menimbang kebutuhan terhadap penyusutan jarak untuk mengatasi restriksi jumlah maksimum diameter ini cukup tinggi, maka analisis kritis terhadap fisibilitas penerapan homomorfisma graf pada STP. Syarat cukup untuk dua buah graf dikatakan homomorfis adalah tidak bertolakbelakang dengan persamaan (4) dan (5) pada halaman tiga. Secara umum, tanpa mempertimbangkan faktor lain di mana STP membawa paket-paket data tertentu, secara matematis sebuah STP seharusnya dapat memiliki jalur alternatif yang homomorfis.

Beberapa faktor dapat menggagalkan sebuah kemungkinan bagi STP untuk dapat mengimplementasikan homomorfisma graf, diantaranya sebagai berikut:

1. *Link* bukan merupakan sisi di graf pada umumnya, melainkan membawa informasi tertentu yang dapat memengaruhi mekanisme STP pada *bridge* lainnya,
2. Proses pembuatan graf yang homomorfis dapat melibatkan sebuah penambahan atau pengurangan simpul, mengingat simpul pada STP bertindak tidak hanya sebagai sebuah titik, *bridge* tentunya berperan dalam proses penyimpanan informasi tertentu yang di bawa melalui BPDU.

B. Fisibilitas Implementasi Homomorfisma Graf pada STP Meninjau Eksistensi Bridge sebagai Simpul pada Graf

Analisis kritis yang ingin dibangun adalah apakah dampak yang akan terjadi apabila terdapat perubahan *designated path* pada salah satu jaringan *bridge* untuk *bridge* itu sendiri dan apakah dampak yang akan terjadi pada *bridge* lain. Dengan mempertimbangkan kembali cara kerja STP pada halaman empat, dapat diperkirakan kemungkinan yang akan terjadi jika dilakukan suatu perubahan terhadap STP dari sisi arsitektur.

STP didesain dengan suatu algoritma tertentu yang memungkinkan bagi seluruh *bridge* untuk dapat berkomunikasi salah satunya dengan adanya *root* yang bertindak seperti *remote* dengan mengendalikan informasi yang diberikan oleh setiap *bridge* berupa BPDU. Setiap perubahan yang dibuat oleh *bridge* akan dilaporkan melalui BPDU agar *root* menerima perubahan tersebut lalu meng-*acknowledge*-nya. Prosedur *acknowledgement* ini memungkinkan *bridge* yang melakukan perubahan mengetahui bahwa perubahan yang diakibatkan olehnya telah diterima oleh pengendali *bridge*. *Root* di saat yang bersamaan juga memberikan informasi kepada seluruh

bridge bahwa telah terjadi perubahan agar seluruh *bridge* segera mengubah status mereka menjadi status yang terbaru.

Mekanisme seperti ini sangat menguntungkan karena antar *bridge* tidak terlalu banyak interaksi yang berarti, sehingga apabila *link* dihilangkan atau ditambahkan maka tidak akan mengganggu kerja *bridge* secara umum. Dengan kata lain, jika ditinjau dari eksistensi *bridge* sebagai ujung simpul pada pohon merentang minimum, maka homomorfisma tetap dapat diimplementasikan pada STP.

C. Fisibilitas Implementasi Homomorfisma Graf pada STP Meninjau Eksistensi Link sebagai Sisi pada Graf

Analisis kritis yang ingin dibangun adalah apakah dampak yang akan terjadi apabila terdapat perubahan *designated path* pada salah satu jaringan *bridge* untuk *link* yang semula menghubungkan *bridge* dan apakah dampak yang akan terjadi pada *link* lain yang tidak melakukan interaksi terhadap *link* yang termodifikasi. Dengan mempertimbangkan kembali cara kerja STP pada halaman empat, dapat diperkirakan kemungkinan yang akan terjadi jika dilakukan suatu perubahan terhadap STP dari sisi arsitektur.

STP didesain dengan algoritma tertentu yang memungkinkan bagi setiap *bridge* untuk berkomunikasi melalui perantara *root*. Namun, sayangnya setiap sisi tidak memiliki mekanisme yang sama, sehingga perlu analisis lebih lanjut untuk melihat pengaruh perubahan jaringan *bridge* terhadap *link*.

Link merupakan perantara dua *bridge* yang fungsinya tidak lain hanyalah sebatas sebagai penanda bahwa kedua *bridge* merupakan tetangga karena tidak terjadi interaksi khusus di luar posisi *link* sebagai sebuah sisi pada graf. Ketika sebuah *link* terhapus, maka tidak akan terjadi *loss data* karena tidak ada data yang tersimpan pada *link* mengingat semua data penting diletakkan pada BPDU dan letaknya bukan berada di *link*, melainkan di *bridge*. Dengan kata lain, karena tidak terdapat pengaruh yang signifikan terhadap proses transfer data, maka property homomorfisma pada graf tetap dapat diimplementasikan pada jaringan *bridge* di dalam STP.

D. Algoritma Pohon Merentang Minimum dengan Memperhatikan Dua Jalur yang Homomorfis

Agar komputer dapat memproses masalah yang manusia ingin coba selesaikan, manusia harus dapat mengubah solusi tersebut ke dalam bahasa yang dimengerti oleh komputer. Atas dasar itu pula, jika penulis ingin membuat suatu mekanisme pencarian pohon merentang minimum dengan memerhatikan dua jalur yang homomorfis di sebuah jaringan komputer, maka perlu dibuat algoritma program yang dapat dilihat pada bagian lampiran.

IV. KESIMPULAN

Permasalahan restriksi diameter jaringan pada topologi pohon rentangan dapat diselesaikan dengan melakukan modifikasi terhadap arsitektur topologi jaringan. Modifikasi tersebut dilakukan dengan mengimplementasikan sifat homomorfisma pada graf agar jalur yang diambil adalah jalur yang lebih singkat dan lebih baik secara bobot. Modifikasi tersebut memerlukan analisis kritis terhadap eksistensi *bridge* sebagai simpul pada graf dan *link* sebagai sisi dari graf, serta dua buah graf dapat dikatakan homomorfis, yakni :

1. $G \rightarrow H$ apabila $\omega(G) = r \leftrightarrow r$ adalah bilangan bulat terbesar untuk $K_r \rightarrow G$,
2. $G \rightarrow H$ apabila $\omega(G) \leq \omega(H)$ dan $\chi(G) \leq \chi(H)$.

V. UCAPAN TERIMA KASIH

Penulis berterima kasih kepada Ibu Harlili, M.Sc yang telah memberikan pengajaran serta bimbingan untuk mata kuliah IF2120 Matematika Diskrit. Ucapan terima kasih juga penulis berikan kepada Bapak Ir. Rinaldi Munir karena telah menyediakan informasi yang sangat relevan terhadap keberjalan kuliah selama satu semester ini. Penulis mengucapkan terima kasih pula kepada rekan-rekan mahasiswa informatika angkatan 2015 dan terdahulu yang telah memberikan inspirasi terhadap proses pembuatan makalah ini. Besar harapan penulis, materi yang disampaikan di kelas bukan hanya sekedar menjadi pelajaran tetapi juga pengajaran agar dapat diimplementasikan di kehidupan sehari-hari.

VI. LAMPIRAN

procedure pathHomomorfis (input P : designated path, output T : pohon)

{ Membentuk pohon yang homomorfis dengan designated path pada jaringan *bridge*

Masukan: designated path P yang merupakan pohon merentang minimum berbobot

Keluaran: pohon T yang homomorfis dengan P dengan diameter yang lebih pendek, mengembalikan P jika tidak ada. }

Kamus Global

k1, k2 c1, c2, r : **integer**

found, done : **boolean**

temp : pohon

Algoritma

c1 \leftarrow $\omega(P)$

r \leftarrow (bilangan bulat terbesar untuk $K_r \rightarrow P$)

if not (c1 = r){

T \leftarrow P

}

else{

k1 \leftarrow $\chi(P)$

found \leftarrow **false**

done \leftarrow **false**

while not (found) **and not** (done) **do**{

temp = *generate* pohon acak yang bisa jadi homomorfis dari P

c2 \leftarrow $\omega(\text{temp})$

k2 \leftarrow $\chi(\text{temp})$

if (c1 \leq c2) **and** (k1 \leq k2){

found \leftarrow **true**

}

if (semua pohon acak telah dicoba){

done \leftarrow **true**

}

}

if (found) {

T \leftarrow temp

}

else{

T \leftarrow P

}

}

DAFTAR PUSTAKA

- [1] Rosen, Kenneth H., *Discrete Mathematics and Its Applications*. 7th ed. New York: McGraw-Hill, 2012.
- [2] Beezer, Rob, *An Introduction to Algebraic Graph Theory*. Puget Sound, Washington DC: Mathematics Department Seminar Pacific University, 2009.
- [3] Beagley, Jonathan and Walter Morris, *Clique Number and Chromatic Number of Graphs defines by Convex Geometries*. Arizona: George Mason University, 2012.
- [4] Cameron, Peter J., *Graph Homomorphisms*. Combinatorics Study Group Notes, September 2006.
- [5] Dyer, Martin and Catherine Greenhill, *The Complexity of Counting Graph Homomorphisms*. Poznan, Poland: John Wiley & Sons, Inc., 2000.
- [6] Shurbert, Davis, *An Introduction to Graph Homomorphisms*. Puget Sound, Washington DC: Department of Mathematics and Computer Science University of Puget Sound, 2013.
- [7] Brewster, Rick, *Graph Homomorphism Tutorial*. Thompson River, Canada: Field's Institute Covering Arrays Workshop of Thompson River University, 2006.
- [8] Perlman, Radia, *An Algorithm for Distributed Computation of a Spanning Tree in an Extended LAN*. San Diego, CA: Digital Equipment Corporation, 1985.
- [9] Blade Network Technologies. *Spanning Tree Protocol Problems and Related Design Considerations*. Santa Clara, California: Blade Network Technologies Inc., 2006.
- [10] Perlman, Radia, *Interconnections*. 2nd ed. Canada: Addison Wesley Longman, Inc., 2000.
- [11] IEEE. *802.1d Spanning Tree Protocol (STP)*. New Jersey, US: IEEE Standard Association, 1990.
- [12] Cisco. *Spanning Tree Protocol Problems and Related Design Considerations*. San Jose, California: Cisco Inc., 2006.
- [13] Cisco Packet Tracer. San Jose, California: Cisco System Inc., 2016.
- [14] <http://mathworld.wolfram.com/>. Diakses pada tanggal 9 Desember 2016.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 9 Desember 2016



Rizky Faramita
13515055