Penerapan Teori Graf pada Tor Path Selection

M. Reyhanullah Budiaman - 13519045[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13519045@std.stei.itb.ac.id

***Abstract*—*Onion routing* adalah metode penelusuran web dengan status anonim yang diterapkan pada TOR *browser*. Karena prosesnya yang melakukan hop dari satu network node ke node lainnya, maka algoritma pemilihan simpul/node dapat memilih rute dengan *latency* sekecil mungkin. Pada makalah ini, akan dibahas implementasi graf pada Tor *Path Selection* dan penjelasan beberapa algoritma yang telah ada, beserta perbandingan satu sistem dengan yang lainnya.**

***Keywords*—*Onion routing*, TOR, Tor *Path Selection*, graf.**

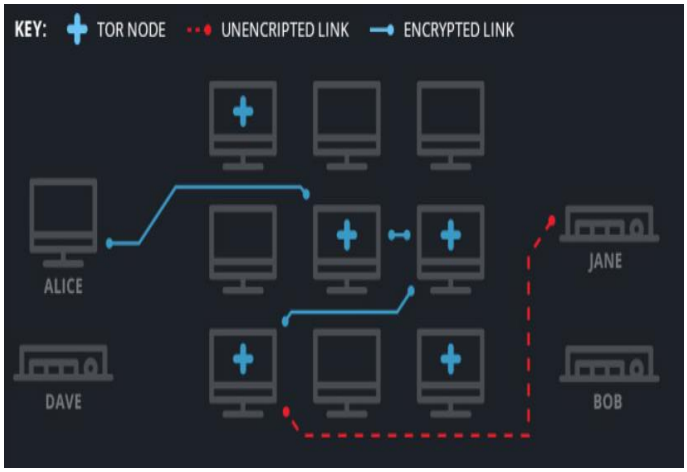
I. PENDAHULUAN

Web browser merupakan salah satu aplikasi yang paling umum digunakan untuk mengakses web dari platform apapun saat ini. Privasi tentunya menjadi permasalahan yang sangat serius di era digital ini. Tentunya melihat beberapa insiden yang melibatkan kebocoran data, pengguna menjadi lebih sadar akan ancaman yang ditimbulkan oleh *hacker* yang memiliki akses ke data pribadi serta aplikasi rentan yang dapat membahayakan data mereka. Sudah ada solusi untuk mengatasi masalah ini yaitu dengan adanya *Virtual Private Network* yang mengamankan paket data yang dikirim ke server dengan cara dienkripsi, namun hanya dengan VPN ini saja apakah data kita aman dari ancaman kebocoran data? Tentu tidak, Walaupun paket data terenkripsi, ada jejak digital server yang diakses dan siapa yang mengakses. Pengguna dengan niat jahat tentu dapat mengeksplotasi hal ini.

Tor atau browser onion router merupakan salah satu aplikasi yang tidak hanya memastikan keamanan privasi tetapi juga memungkinkan komunikasi anonim di internet dengan adanya jaringan *relay* yang pengguna dapat merutekan TCP Traffic-nya.

Namun, beberapa langka untuk meningkatkan anonimitas pengguna secara fundamental meningkatkan latensi komunikasi. Sebagai contoh, klien Tor yang default membuat *tunnel* antara dirinya dengan tujuan melalui tiga *relay* yang dipilih secara acak, dengan beberapa tolak ukur untuk stabilitas *relay* dan *bandwith* tautan akses. Pemilihan *relay* secara acak ini dapat menyebabkan rute yang diambil sangat jauh sehingga menyebabkan *latency* tinggi.

Karena permasalahan *latency* yang telah dijelaskan diatas, tentunya algoritma *Path Selection* pada Tor haruslah mempertimbangkan beberapa hal dan memastikan agar *latency* sekecil mungkin. Permasalahan ini tentunya dapat diselesaikan dengan teori graf karena topologi *network* menyerupai sebuah graf.



Gambar 1 Contoh Tor Network
Sumber: https://www.anonabox.com/what-is-tor.html

Gambar diatas merupakan visualisasi dari *Tor circuit*. Tor circuit ini dapat dikatakan sebagai graf dengan Tor Node sebagai simpul dan Link (*relay*) sebagai sisi..

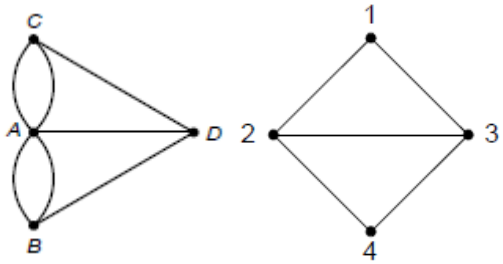
II. LANDASAN TEORI

A. Graf

Graf adalah struktur yang terdiri dari himpunan sejumlah objek yang disebut simpul dan himpunan sisi yang menyatakan hubungan antara simpul-simpul tersebut. Graf dinyatakan sebagai berikut,

$$G = (V,E)$$

dengan G merupakan graf, V merupakan himpunan simpul (*vertex*) dan E merupakan himpunan sisi (*edge*).



Gambar 2.1 Contoh Graf
Sumber: https://informatika.stei.itb.ac.id/~rinaldi-munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf

Ada beberapa terminologi graf yang perlu diketahui antara lain:

1. Bertetangga (*Adjacent*)
   Dua buah simpul dikatakan bertetangga jika kedua simpul tersebut terhubung secara langsung oleh suatu sisi. Sebagai contoh, pada gambar 2.1, simpul 1 bertetangga dengan simpul 2 dan 3.

2. Bersisian (*Incidency*)
   Suatu sisi e dikatakan bersisian dengan simpul v_1 dan simpul v_2 jika menghubungkan kedua simpul tersebut.

3. Simpul Terpencil (*Isolated Vertex*)
   Jika suatu simpul tidak mempunyai sisi yang bersisian dengannya maka simpul tersebut dinamakan simpul terpencil.

4. Derajat (*Degree*)
   Derajat suatu simpul merupakan jumlah sisi yang bersisian dengan simpul tersebut. Sebagai contoh, dari gambar 2.1, simpul 4 berderajat 2 dan simpul 3 berderajat 3.

5. Lintasan (*Path*)
   Lintasan dari suatu simpul awal v_0 ke simpul tujuan v_T di dalam suatu graf G merupakan barisan berselang-seling simpul-simpul dan sisi-sisi yang berbentuk v_0, e_1, v_1, e_2, …, v_{n-1}, e_n, v_n sedemikian sehingga $e_1 = (v_0, v_1)$, $e_2 = (v_1, v_2)$, … , $e_n = (v_{n-1}, v_n)$.

6. *Cut-Set*
   Cut-set dari graf terhubung G adalah himpunan sisi yang bila dibuang dari G menyebabkan G tidak terhubung.

7. Sirkuit (*Circuit*) atau Siklus (*Cycle*)
   Suatu lintasan yang berawal dan berakhir pada simpul yang sama dinamakan Sirkuit (*Circuit*) atau Siklus (*Cycle*).

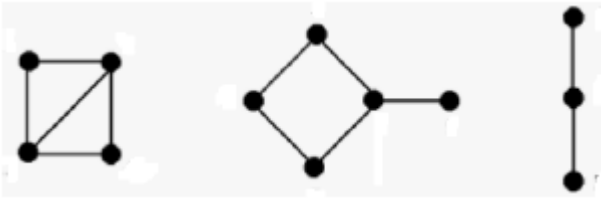
8. Keterhubungan
   Dua buah simpul v_1 dan v_2 dikatakan terhubung jika terdapat lintasan dari simpul v_1 ke simpul v_2.

9. Upagraf (*Subgraph*)
   Misalkan $G = (V, E)$ merupakan suatu graf, maka $G_1 = (V_1, E_1)$ dinamakan upagraf dari G jika $V_1 \subseteq V$ dan $E_1 \subseteq E$.

10. Upagraf Merentang (*Spanning Subgraph*)
    Misalkan $G_1 = (V_1, E_1)$ merupakan upagraf dari graf $G = (V, E)$. Jika $V_1 = V$ maka G_1 dinamakan upagraf merentang.

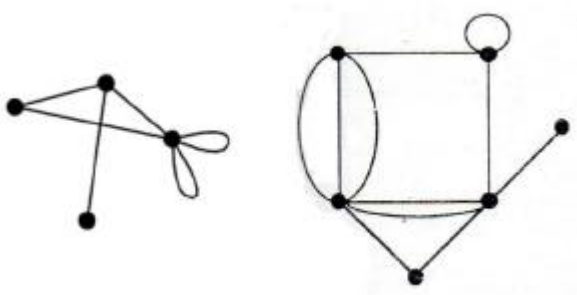
Ada beberapa jenis graf, berdasarkan ada tidaknya gelang atau sisi ganda pada suatu graf, graf digolongkan menjadi graf sederhana sederhana (*simple graph*) dan graf tak-sederhana (*unsimple-graph*).

a. Graf Sederhana
Graf yang tidak mengandung gelang maupun sisi ganda dinamakan graf sederhana.



Gambar 2.2 Contoh graf sederhana
Sumber: https://informatika.stei.itb.ac.id/~rinaldi-munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf

b. Graf Tak-sederhana
Graf yang mengandung sisi ganda atau gelang dinamakan graf tak sederhana. Graf tak-sederhana dibedakan lagi menjadi graf ganda, yaitu graf yang mengandung sisi ganda, dan graf semu, graf yang mengandung sisi gelang. Pada contoh graf dibawah, graf di kiri merupakan graf semu karena memiliki 2 sisi gelang. Graf di kanan memiliki 2 sisi ganda dan 1 sisi gelang.

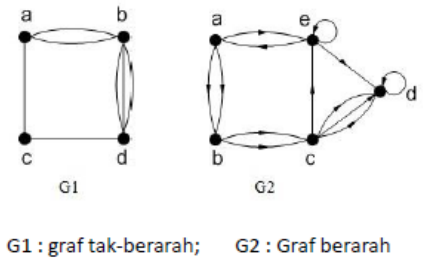


Gambar 2.3 Contoh graf tak-sederhana
Sumber: https://informatika.stei.itb.ac.id/~rinaldi-munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf

Berdasarkan orientasi arah pada sisi, secara umum, graf dibedakan atas 2 jenis:

c. Graf Tak-berarah (*undirected graph*)
Graf yang sisinya tidak mempunyai orientasi arah disebut graf tak-berarah.

d. Graf Berarah (*directed graph* atau *digraph*)
Graf yang setiap sisinya diberikan orientasi arah disebut sebagai graf berarah.

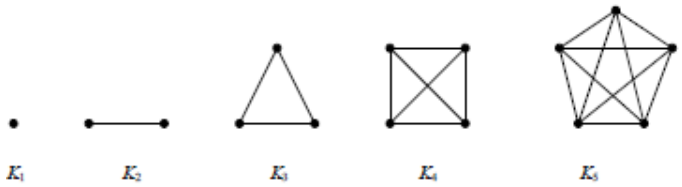


Gambar 2.3 Contoh Graf tak-berarah dan berarah
Sumber: https://informatika.stei.itb.ac.id/~rinaldi-munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf

Beberapa jenis graf lainnya antara lain:

e. Graf Lengkap (*complete graph*)

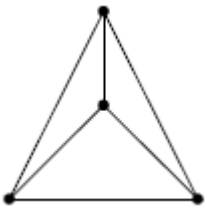
Graf lengkap merupakan graf sederhana yang setiap simpulnya terhubung ke semua simpul lainnya. Dengan kata lain, setiap simpulnya bertetangga.Graf lengkap dengan n buah simpul jumlah sisinya adalah n(n-1)/2 sisi.



Gambar 2.4 Contoh graf lengkap
Sumber: https://informatika.stei.itb.ac.id/~rinaldi-munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf

f. Graf Teratur (*regular graph*)

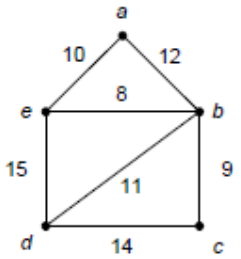
Graf teratur merupakan graf yang setiap simpulnya mempunyai derajat yang sama. Apabila derajat setiap simpul pada graf teratur adalah r, maka graf tersebut dinamakan graf teratur berderajat r. Jumlah sisi pada graf teratur dengan n simpul adalah nr/2 sisi.



Gambar 2.5 Contoh graf teratur
Sumber: https://informatika.stei.itb.ac.id/~rinaldi-munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf

g. Graf Berbobot (*weighted graph*)

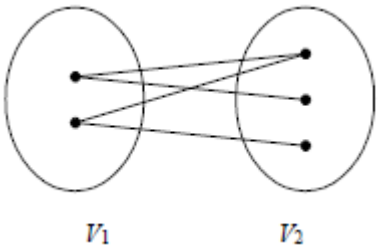
Graf berbobot adalah graf yang setiap sisinya diberi sebuah nilai atau bobot.



Gambar 2.6 Contoh graf berbobot
Sumber: https://informatika.stei.itb.ac.id/~rinaldi-munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf

h. Graf Bipartit (*bipartite graph*)

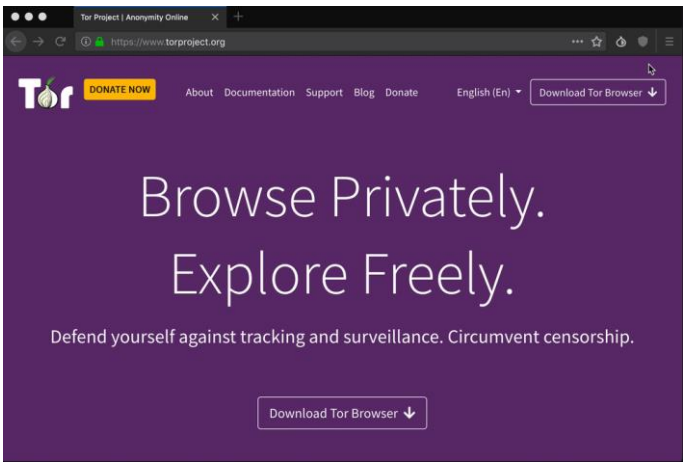
Sebuah graf G dikatakan graf bipartite jika himpunan simpul pada graf tersebut dapat dipisah menjadi dua himpunan tak kosong yang *disjoint*, misalkan V_1 dan V_2, sedemikian sehingga setiap sisi pada G menghubungkan sebuah simpul pada V_1 dan V_2.



Gambar 2.7 Contoh graf bipartit
Sumber: https://informatika.stei.itb.ac.id/~rinaldi-munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf

B. TOR

Tor adalah jaringan anonimitas latensi rendah yang berbasis pada *onion routing* tetapi dengan beberapa modifikasidan perbaikan dari desain aslinya dari segi keamanan, dan efisiensi. Jaringan Tor mencakup sekumpulan set authoritative directory servers terpercaya yang bertanggung jawab untuk mengumpulkan dan mendistribusi *signed infomation* tentang *router* yang ada di jaringan.



Gambar 2.18 Homepage Tor Browser
Sumber: https://blog.torproject.org/new-release-tor-browser-85

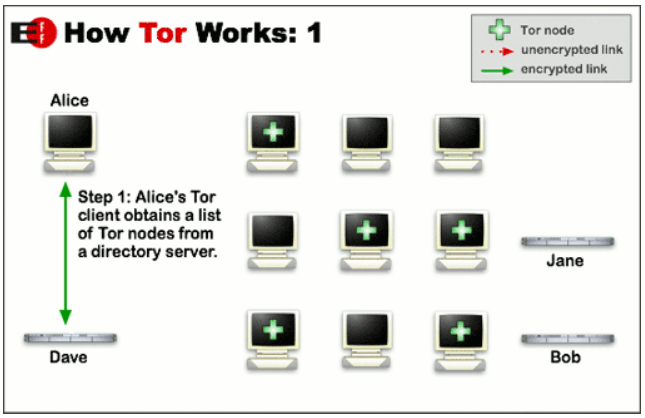
Klien Tor mengambil *directory information* dari *directory*

mirror untuk mengetahui infomasi tentang server lain yang ada di jaringan Tor, seperti alamat IP, *public keys*, dan lain-lain.

Untuk membuat koneksi anonim melalui jaringan Tor, klien pertama-tama memilih barisan terurut dari 3 server (umumnya). Klien kemudian mengambil *session keys* dari masing-masing server dimulai dari server pertama (*entry node*) pada barisan. Klien kemudian dapat terhubung ke server kedua (*middle node*) melalui *tunnel* terenkripsi yang dibangun oleh server pertama, kemudian hal yang sama dilakukan pada server ketiga (*last node*). Server terakhir atau *last node* dinamakan *exit node* karena server ini bertanggung jawab untuk membuat koneksi dari jaringan Tor ke tujuan yang diinginkan klien. Hasilnya adalah *tunnel* terenkripsi melalui jaringan Tor yang disebut sebagai *circuit* (mulai dari sini, istilah *circuit* yang dipakai mengacu pada *circuit* di jaringan Tor, bukan *circuit* pada graf).

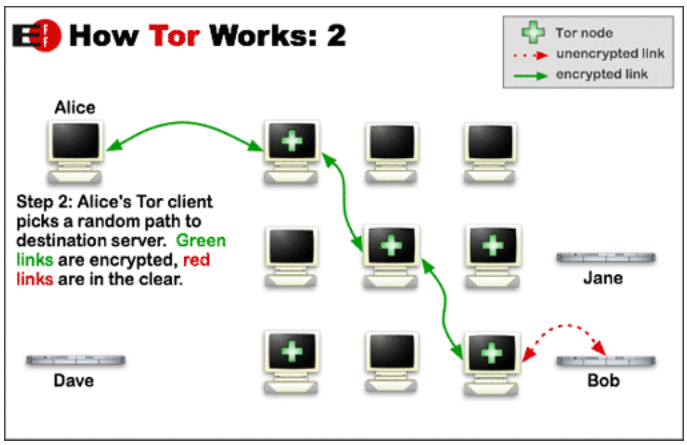
Metode yang digunakan perangkat lunak Tor untuk memilih node pada *circuit* klien telah mengalami banyak perubahan karena sejak desain awalnya dipublikasikan. Sebagai contoh, pada awalnya klien memilih semua node pada *circuit* secara merata dan acak. Setelahnya, bentuk primitif dari *load-balancing* ditambahkan yaitu relay dipilih secara proporsional dengan perkiraan *self-reported bandwith* berdasarkan seberapa banyak lalu lintas yang di relay setiap server selama interval pengukuran.

Cara kerja Tor adalah sebagai berikut,



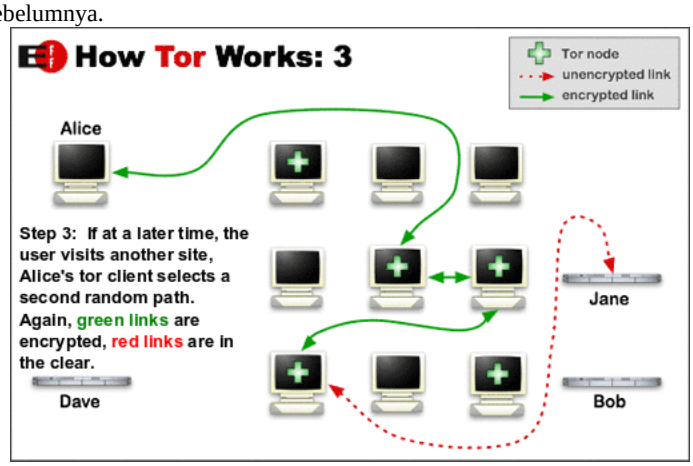
Gambar 2.9 Cara Kerja Tor 1
Sumber: https://2019.www.torproject.org/about/overview

Pertama, Klien, dalam contoh ini Alice, akan mengambil list dari Tor node pada *directory server*, Kemudian Alice ingin mengakses Bob (tujuan), maka Alice akan memilih tiga buah node secara acak dari list Tor Node, kemudian mengambil *session key* dari tiap node. informasi yang ingin disampaikan ke Bob akan dienkripsi dengan tiga session key ini secara berlapis. Lapisan enkripsi pertama hanya bisa didekripsi oleh node pertama (*entry node*), lapisan enkripsi kedua didekripsi oleh node kedua, dan lapisan enkripsi ketiga didekripsi oleh node ketiga. Kemudian pesan yang telah didekripsi akan disampaikan ke Bob.



Gambar 2.10 Cara Kerja Tor 2
Sumber: https://2019.www.torproject.org/about/overview

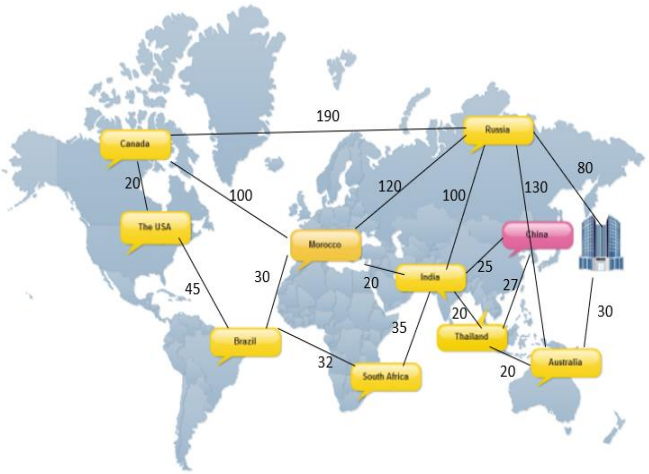
Saat Alice ingin berkomunikasi dengan laman lain (Jane), maka Alice akan memilih tiga Tor node dari list yang telah diambil dan melakukan proses yang sama dengan diatas. Rute yang dipilih pada tahap ini haruslah berbeda dari rute yang telah dipilih sebelum-sebelumnya.



Gambar 2.11 Cara Kerja Tor 3
Sumber: https://2019.www.torproject.org/about/overview

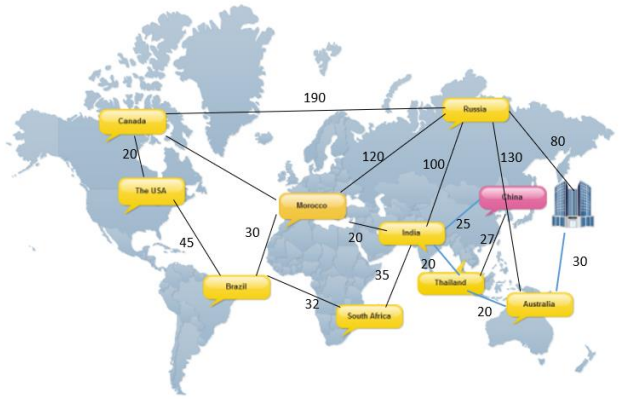
III. PENGAPLIKASIAN TEORI GRAF PADA TOR PATH SELECTION

LASTor merupakan algoritma Tor Path Selection yang meimilih *circuit* berdasarkan rute terpendek secara geografis. Klien akan mengambil sekumpulan set server dari *directory server*, kemudian sekumpulan set server ini beserta klien dan tujuan klien dapat direpresentasikan dengan graf berbobot. Upagraf yang mencakup semua set server merupakan graf lengkap. Kemudian algoritma akan mencari rute dari simpul klien ke simpul tujuan dengan syarat harus melewati tepat 3 simpul kecuali simpul klien dan simpul tujuan. Sebagai contoh, seorang klien di china ingin mengakses suatu server yang direpresentasikan oleh ikon bangunan pada gambar dibawah.



Gambar 3.1 Contoh node jaringan Tor
Sumber: penulis

Perlu diperhatikan bahwa nilai pada gambar hanyalah sebagai contoh dan tidak sesuai dengan jarak antar node seperti yang sebenarnya. Klien tersebut kemudian mendapatkan sekumpulan server pada jaringan Tor yang direpresentasikan dengan warna kuning. Kemudian LASTor akan mengkomputasi jarak dari klien ke tujuan melalui 3 node berdasarkan bobot pada graf. Pilihan 3 node tadi dikatakan sebagai rute. Dari seluruh rute yang telah dikomputasi, LASTor kemudian menggunakan algoritma Weighted Shortest Path (WSP) kemudian memilih satu rute dengan probabilitas dari rute yang dipilih proporsional dengan bobot yang berhubungan dengannya; bobot yang berhubungan yang dimaksud adalah perbedaan antara jarak *end-to-end* maksimum dari semua rute dan jarak dari rute tertentu. Hasil rute yang dipilih dapat dilihat pada gambar dibawah yang berwarna biru.



Gambar 3.2 Hasil pemilihan rute dari penerapan LASTor
Sumber: penulis

Pada implementasi sebenarnya, sekumpulan server pada jaringan Tor sangat banyak dan setiap node dapat berhubungan dengan node lainnya.

Penerapan WSP secara langsung ini memiliki 2 permasalahan. Pertama, preferensi WSP untuk rute dengan jarak geografis lebih rendah akan menghasilkan preferensi yang lebih besar terhadap rute dengan relay yang mendekati lintasan langsung ke tujuan. Dengan kata lain, rute yang dipilih oleh klien akan sangat mudah ditebak dan menurunkan anonimitas klien. Kedua, waktu komputasi yang besar. Tor memiliki ribuan node dan kurang dari setengah node tersebut yang bisa menjadi *exit node*. Komputasi naif dari algoritma untuk sekitar 2500 node dan 1000 *exit node* akan memakan waktu sekitar 6.5 detik pada proses 2.5 GHz.

Untuk mengatasi masalah tersebut, Algoritma LASTor mengelompokkan node yang dekat secara geografis. Node tersebut dikelompokkan dengan cara membagi *globe* (peta dunia) menjadi beberapa kotak kecil yang ukurannya dapat diatur. WSP akan mengkomputasi *Clustered Tor Network* tersebut dengan tiap simpul graf merupakan kelompok node dan tiap kandidat rute adalah melewati tiga kelompok/kluster. WSP kemudian mengkomputasi jarak dari tiap rute (lintasan) tingkat kluster dan memilih rute terpendek. Node pada tiap kluster dipilih secara acak. Modifikasi tersebut telah terbuktikan menurunkan waktu komputasinya.

Tentunya algoritma Tor Path Selection yang ada pada era ini sudah tidak memakai algoritma ini. Pada tahun 2017, Browser Tor telah menerapkan Counter-RAPTOR yang memiliki preferensi *guard* dengan BGP *hijack-resilient path*. Pada tahun 2019, Tor menerapkan TrilateraTor yang memilih rute dari berbagai negara.

IV. KESIMPULAN

Teori graf dapat diterapkan pada kehidupan sehari-hari. Teori ini tentunya sangat berguna dalam mempelajari jaringan komputer. Salah satunya adalah penerapannya dalam mendesain algoritma *Tor Path Selection*. *Tor Path Selection* ini membuat penelusuran secara anonim dapat dilakukan secara efeisien. Saat klien akan mengunjungi website yang lain, Tor akan mencari rute yang berbeda dari rute yang telah dipilih sebelumnya. Hal ini dilakukan agar lokasi klien sukar untuk dilacak. Karena itu algoritma *Tor Path Selection* akan lebih baik jika waktu komputasi yang dibutuhkan seminimal mungkin. Hal ini tentunya dapat diselesaikan dengan mendalami teori graf.

Penelitian untuk Tor Path Selection ini tentunya akan terus dilanjutkan untuk mendapatkan algoritma yang lebih efisien dan aman. Di era dimana jaringan komputer sudah menjadi bagian dari hidup kita, tentunya kita perlu tidak bisa mengacuhkan keamanan dari privasi kita

V. UCAPAN TERIMA KASIH

Penulis bersyukur kepada Tuhan Yang Maha Esa karena atas berkat dan rahmat-Nya, penulis mampu menyelesaiakan makalah berjudul “Penerapan Teori Graf pada Tor Path Selection”. Penulis ucapkan terima kasih kepada Pak Rinaldi Munir selaku pengajar mata kuliah Matematika Diskrit di K01 yang telah membantu penulis dalam memahami dan mempelajari materi yang diambil sebagai bahan referensi dalam makalah ini.

REFERENSI

[1] Munir, Rinaldi. 2020. Bahan Kuliah Matematika Diskrit IF2120 Graf (Bag 1).

[2] https://informatika.stei.itb.ac.id/~rinaldi-munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf (Diakses pada tanggal 10 Desember 2020)

[3] https://blog.torproject.org/new-release-tor-browser-85 (Diakses pada tanggal 10 Desember 2020)

[4] https://2019.www.torproject.org/about/overview (Diakses pada tanggal 10 Desember 2020)

[5] https://www.anonabox.com/what-is-tor.html (Diakses pada tanggal 10 Desember 2020)

[6] https://ieeexplore.ieee.org/abstract/document/6234431 (Diakses pada tanggal 10 Desember 2020)\

[7] https://content.sciendo.com/view/journals/popets/2019/4/article-p272.xml?language=en (Diakses pada tanggal 10 Desember 2020)

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, 11 Desember 2020

M. Reyhanullah Budiaman
13519045