

Teknik Pencarian dalam Jaringan *Peer to Peer* dengan Memanfaatkan *Breadth First Search*

Andarias Silvanus 13512022
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
andarias@std.stei.itb.ac.id

Abstrak—Beragam jenis pendistribusian data telah dikembangkan seiring perkembangan jaman. Mulai dari pendistribusian data antar server dan klien bahkan diciptakan juga pendistribusian data antar komputer. Teknik ini dilihat sebagai suatu kesempatan untuk memperbaiki teknik pendistribusian data. Pembaharuan teknik juga dilakukan dengan tujuan meningkatkan efisiensi dan performansi serta ditambahkan cakupan terhadap banyak pelaku. Oleh karena itu muncullah sistem pembagian data dengan memakai *torrent*. Teknik pencarian antar komputer memakai banyak metode, salah satunya mengaplikasikan algoritma *Breadth First Search*.

Kata kunci—*BFS, peer to peer, jaringan, teknik pencarian, breadth first search*

I. PENDAHULUAN

Informasi merupakan suatu kebutuhan dasar yang beredar di sekitar kita. Hampir segala sesuatu yang berada di sekitar kita memiliki nilai dan informasi. Kebutuhan akan informasi telah muncul dari jaman dahulu. Seiring berkembangnya jaman, kuantitas dan kualitas akan kebutuhan informasi semakin berkembang. Kita memerlukan banyak informasi yang berguna untuk dapat bertahan hidup. Kuantitas juga diperlukan, semakin banyak informasi yang kita dapat, semakin berharga juga nilai yang kita dapat.

Pada abad ke 21 juga dikenal sebagai era informasi. Informasi dapat beredar dengan luas dan dapat ditemui dengan mudah di sekitar kita. Perkembangan di dunia internet sangat mempengaruhi perkembangan penyebaran informasi. Dengan berkembangnya infrastruktur dan perangkat lunak yang mumpuni, berselancar di dunia maya hampir dapat dilakukan oleh banyak orang. Apalagi dengan berkembangnya berbagai perkakas (*gadget*) yang disuntikkan berbagai persiapan yang mampu menyokongnya untuk berinteraksi dengan dunia maya dengan mudah. Sebut saja telepon pintar (*smartphone*), tablet dan laptop. Perangkat-perangkat tersebut dapat dibawa dengan mudah oleh banyak orang ke berbagai tempat karena sifatnya yang *mobile* dan *portable*.

Berbagai informasi maupun data telah banyak beredar dan tersimpan di dunia maya. Berbagai macam foto,

video, dokumen, perangkat lunak dan informasi yang langsung menyentuh masyarakat luas secara umum. Data-data ini dapat berskala besar maupun kecil. Dan data-data ini juga disediakan untuk memenuhi kebutuhan banyak orang. Karena begitu banyaknya orang yang memerlukan data-data ini, maka sistem pendistribusian data perlu dipikirkan dan dirumuskan secara spesifik agar keefisienan dan performansi yang diberikan dapat menampung semua keinginan pengguna secara tepat guna.

Berdasarkan fungsinya, jaringan pendistribusian data dapat terbagi dua. Yaitu sistem jaringan *client-server* dan sistem jaringan *peer to peer*. Sistem jaringan *client-server* banyak diaplikasikan dalam penyimpanan data (*file hosting*) oleh berbagai pihak. Contohnya mediafire dan megaupload. Prinsip dari sistem ini adalah data disimpan pada suatu server dan pengguna yang hendak mengakses data tersebut akan berperan sebagai klien dan diharuskan meminta kepada servernya terlebih dahulu.

Namun setelah sistem jaringan *peer to peer* ditemukan, dikembangkan pendistribusian data dengan memakai *torrent*. Prinsip kerja dari sistem ini akan dibahas lebih lanjut di poin selanjutnya.

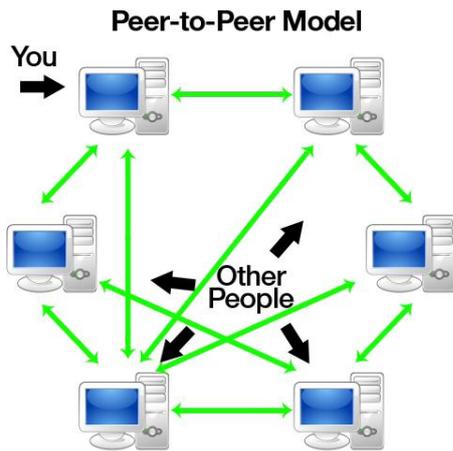
II. DASAR TEORI

Jaringan *Peer to Peer*

Jaringan *peer to peer* merupakan salah satu model jaringan komputer yang terdiri dari dua atau beberapa komputer, dimana setiap komputer yang terdapat di dalam lingkungan jaringan tersebut bisa saling berbagi. Jaringan ini juga dapat diimplementasikan pada ruang lingkup kecil dan lokal. Sebagai contoh, untuk membuat jaringan *peer to peer* dengan dua komputer, kita dapat menggunakan kabel UTP yang dipasang pada kartu jaringan masing-masing komputer untuk mengkomunikasikan antar komputer.

Dalam jaringan *peer to peer*, semua komputer memiliki hak akses yang sama. Setiap komputer yang terhubung dapat saling berbagi sumber daya tanpa harus dikendalikan oleh satu komputer saja seperti sistem jaringan *client-server*. Setiap permintaan layanan (*service*)

dan setiap sumber daya (*resource*) disimpan dan dibagikan pada hubungan antar komputernya.



Ilustrasi Sistem Jaringan Peer to Peer (sumber: Codeproject.com)

Akibat dari sistem ini adalah setiap komputernya memiliki kemampuan dan tugas yang sepadan. Hal ini sangat berbeda jika dibandingkan dengan sistem jaringan *client-server*. Server pada jaringan tersebut memiliki kemampuan yang melebihi komputer lainnya yang bergantung kepadanya dan tugas yang lebih berat karena server harus mengurus berbagai permintaan layanan yang diajukan oleh klien-kliennya.

Keuntungan dari sistem ini adalah sebagai berikut:

- Implementasinya murah dan mudah.
- Tidak memerlukan perangkat lunak administrasi jaringan khusus.
- Tidak membutuhkan administrator jaringan
- Jika ada gangguan pada salah satu komputer yang berada di dalam jaringan, jaringan tidak akan terganggu. Hal ini disebabkan karena hak akses dan kedudukan yang sama antar komputernya.

Namun di samping keuntungan, ada juga kelemahan yang belum mampu ditaklukkan oleh sistem ini. Kelemahan tersebut antara lain:

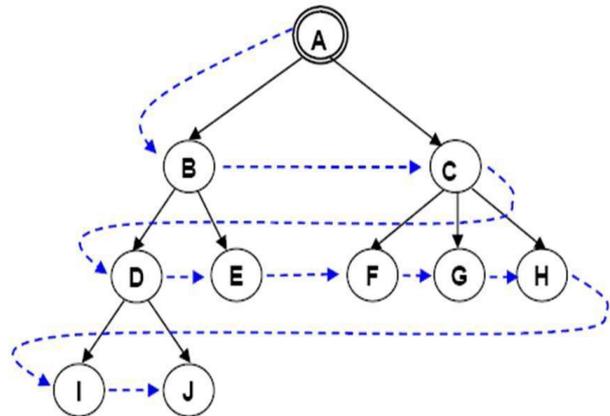
- Tidak cocok digunakan untuk jaringan dalam skala besar, karena administrasi menjadi tidak terkontrol.
- Tiap user harus dilatih untuk menjalankan tugas administratif agar dapat mengamankan komputernya masing-masing.
- Tingkat keamanannya rendah.
- Semakin banyak data yang dibagikan, efeknya akan mempengaruhi kinerja komputer dan sistem.

Breadth First Search

Breadth First Search atau BFS adalah algoritma pencarian yang diterapkan dalam sebuah graf yang inti dari operasi yang dilakukannya hanya dua jenis. Yaitu mendatangi dan mengecek pada suatu simpul tertentu yang terdapat di dalam graf dan memiliki hak akses untuk mendatangi simpul tetangga dari simpul yang sekarang

sedang dikunjungi.

Algoritma BFS dimulai dari simpul akar, dan simpul akar tersebut akan menciptakan simpul anak atau memiliki simpul anak. Kemudian BFS akan mendatangi satu per satu simpul anaknya dan apabila semua simpul anak yang berada pada tingkat yang sama telah semua dikunjungi, BFS akan mendatangi simpul anak pertama dari simpul pertama di tingkat tersebut. Setiap simpul yang telah dikunjungi ditandai pada sebuah tipe data agar simpul yang telah dikunjungi tidak akan diulangi pengecekannya. Dan hal tersebut diulang sampai ditemukannya simpul solusi atau simpul yang dicari.



Ilustrasi Pencarian BFS dalam Sebuah Graf (sumber: <http://www.ramizilmazer.com/>)

Dan ini *pseudocode* untuk algoritma BFS berdasarkan slide kuliah Strategi Algoritma yang dibuat oleh Pak Dr. Ir. Rinaldi Munir, MT. dan dimodifikasi oleh Bu Masayu Leylia Khodra, ST.,MT.

```

procedure BFS (input v:integer)
Deklarasi
w: integer
q : antrian
procedure BuatAntrian (input/output q: antrian)
procedure MasukAntrian (input/output q: antrian,
input v: integer)
procedure HapusAntrian (input/output q: antrian,
output v: integer)
function AntrianKosong (input q: antrian)
→boolean
Algoritma
BuatAntrian(q)
write(v)
dikunjungi[v] ← true
MasukAntrian (q,v)
while not AntrianKosong(q) do
  HapusAntrian(q,v)
  for tiap simpul w yang bertetangga dengan
  simpul v do
    if not dikunjungi[w] then
      write(w)
      MasukAntrian(q,w)
      dikunjungi[w] ← true
  
```

```

endif
endfor
endwhile

```

Algoritma BFS yang murni juga memiliki struktur data yang mendukungnya untuk dapat melakukan pencarian dalam graf. Berdasarkan *slide* kuliah yang dibuat oleh Pak Dr. Ir. Rinaldi Munir, MT. struktur data algoritma BFS adalah sebagai berikut.

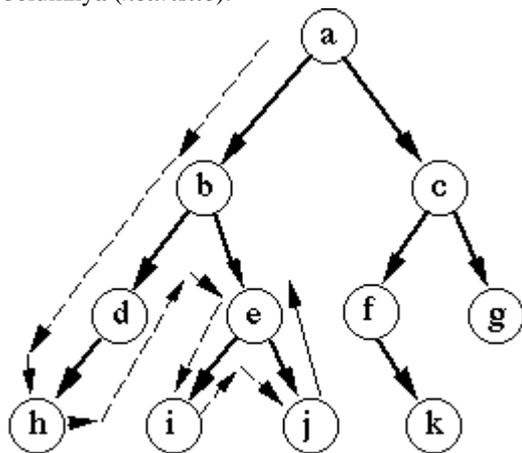
Matriks ketetanggaan $A = [a_{ij}]$ yang berukuran $n \times n$, $a_{ij} = 1$, jika simpul i dan simpul j bertetangga, $a_{ij} = 0$, jika simpul i dan simpul j tidak bertetangga.

Antrian q untuk menyimpan simpul yang telah dikunjungi.

Tabel boolean yang bernama dikunjungi
dikunjungi : array[1..n] of boolean
dikunjungi[i] = true jika simpul i sudah dikunjungi
dikunjungi[i] = false jika simpul i belum dikunjungi

Depth First Search

Depth First Search atau DFS adalah algoritma pencarian yang diterapkan pada lingkungan graf juga seperti dalam algoritma BFS. Metode pencarian dimulai dari simpul akar dan kemudian pencarian dilakukan terhadap setiap simpul anak yang dimiliki. Pencarian dilakukan apabila simpul solusi ditemukan atau pencarian telah melewati batas tertentu yang telah diterapkan sebelumnya (*heuristic*).



Depth-first search

Ilustrasi Pencarian DFS dalam Sebuah Graf (sumber: <http://www.cse.unsw.edu.au/~billw>)

Algoritma DFS dasar juga memiliki struktur data yang mirip dengan algoritma BFS. Untuk struktur data dasar dapat dilihat sebagai berikut:

Matriks ketetanggaan $A = [a_{ij}]$ yang berukuran $n \times n$, $a_{ij} = 1$, jika simpul i dan simpul j bertetangga, $a_{ij} = 0$, jika simpul i dan simpul j tidak bertetangga.

Antrian q untuk menyimpan simpul yang telah dikunjungi.

Tabel boolean yang bernama dikunjungi
dikunjungi : array[1..n] of boolean
dikunjungi[i] = true jika simpul i sudah dikunjungi
dikunjungi[i] = false jika simpul i belum dikunjungi

Pseudocode untuk algoritma DFS adalah sebagai berikut:

Procedure DFS (input v : integer)

Deklarasi

w : integer

Algoritma

Write (v)

Dikunjungi[v] \leftarrow true

for $w \leftarrow 1$ to n do

if $A[v,w] = 1$ then

if not dikunjungi[v] then

DFS(w)

endif

endif

endfor

Algoritma DFS umumnya dilakukan secara rekursif. Namun bisa juga dilakukan dengan pengulangan (*looping*).

Torrent

File torrent (biasa dikenal lewat ekstensinya yang berupa .torrent) merupakan sebuah file yang memiliki data mengenai file dan direktori yang terdistribusi dalam lingkungan torrent (*swarm*) dan memiliki daftar lokasi jaringan dari para komputer yang menjadi stasionnya. Setiap komputer berperan dalam pembagian (*sharing*) data terhadap komputer yang meminta layanan.

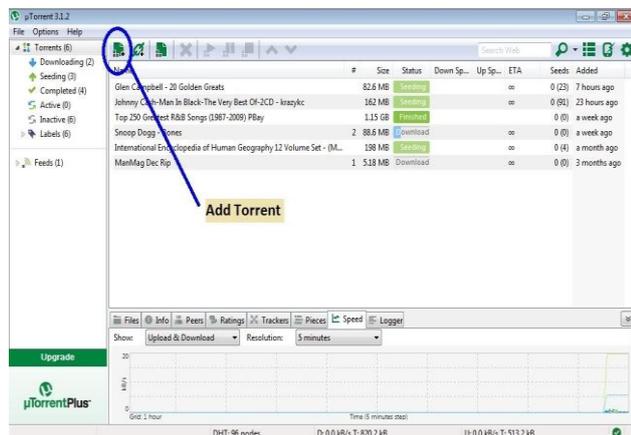
File torrent dibuat dalam format file binari. Elemen-elemen yang terkandung di dalamnya adalah sebagai berikut:

- *Announce*, URL dari *tracker* (server yang membantu pengadaan komunikasi antara pengguna berdasarkan protokol torrent)
- *Info*, memetakan pada kamus apakah suatu atau lebih file sedang dibagi (*share*) atau tidak. Di dalam info terbagi lagi atas:
 - o *Name*, direktori atau tempat penyimpanan file
 - o *Piece length*, jumlah *byte* per bagian file
 - o *Pieces*, berupa daftar *hash*
 - o *Length*, ukuran file dalam *bytes*

- *Files*, daftar kamus yang masing-masingnya berkorespondensi pada file. Setiap kamus memiliki kunci (*key*) sebagai berikut:
 - *Path*, kumpulan *string* yang berkorespondensi dengan upadirektori dari *name*
 - *Length*, ukuran file dalam *bytes*

Prinsip torrent merupakan pengembangan terhadap jaringan *peer to peer*. Setiap komputer yang tergabung dalam *swarm*, merupakan partisipan yang berhak meminta layanan atau mengakses data dan wajib membagikan data yang telah diambil. Semua komputer berkedudukan sama, seperti layaknya jaringan *peer to peer*.

Setiap data yang beredar telah diatur dalam protokol torrent. Protokol ini memperbolehkan setiap pengguna untuk langsung terhubung secara langsung dengan pengguna lainnya. Pada suatu *swarm* umumnya dapat terdapat lebih dari 2 komputer. Dan bila ada satu pengguna yang hendak mengakses suatu data, maka pengguna tersebut akan tersambung dengan semua *seeder*. *Seeder* adalah orang memiliki file yang utuh dari sebuah file torrent. Jika terdapat lebih dari satu *seeder*, pengguna tersebut akan mengambil setiap potongan-potongan bit dari setiap file yang dimiliki dari setiap *seeder*. Hal ini dimungkinkan oleh adanya peran protokol torrent. Protokol torrent mengatur pembagian file/bit tersebut bersifat unik sehingga tidak ada pengunduhan duplikasi data. Ketika data didistribusikan menggunakan protokol torrent, setiap pengguna yang mengunduh file tersebut juga ikut mendistribusikan sebagian kecil data kepada pengguna baru yang mendownload file/data tersebut.



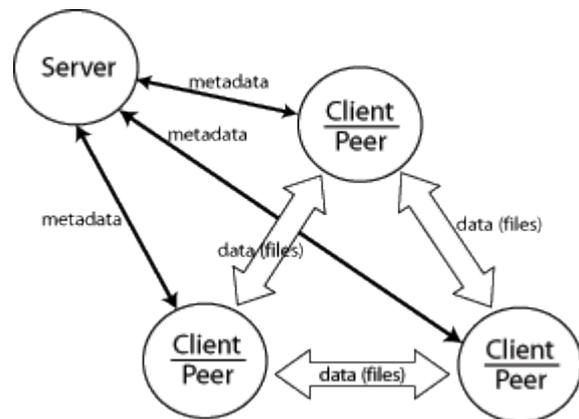
Contoh Pengunduhan File Melalui Torrent Client uTorrent (sumber gambar: <http://s201.photobucket.com/user/559Josh/media/>)

Jadi, torrent bekerja dengan mengelompokkan file yang diminta menjadi potongan-potongan kecil dari file tersebut. Saat ada permintaan pengaksesan terhadap suatu file, *seeder* akan mengirimkan potongan kecil file tersebut pada *swarm*. Protokol torrent yang mengawasi dan mengontrol alur pembagian data. Untuk tetap terhubung dan dapat mendistribusikan data, pengguna memerlukan suatu perangkat lunak, yaitu torrent client.

III. TEKNIK PENCARIAN DALAM TORRENT

Dalam pencarian file yang diminta oleh pengguna, salah satu metode pencariannya memanfaatkan algoritma BFS, baik secara langsung maupun dengan modifikasi. Karena BFS adalah algoritma yang menangani pencarian di dalam graf, maka lingkungan pun dimodelkan menjadi seperti graf.

Setiap pengguna, atau komputer yang berada pada *swarm* dilambangkan sebagai simpul. Prinsip dasar dari pencarian dengan metode BFS adalah simpul yang meminta layanan mengirimkan berbagai permintaan (*request*) atau pesan *query* terhadap simpul-simpul tetangganya. Simpul-simpul tetangga ini akan mengirimkan pesan *query* ini terhadap simpul-simpul tetangga lainnya. Pencarian dilakukan untuk mencari simpul mana saja yang memiliki file serupa seperti yang diminta oleh simpul peminta. Metode pencarian persis seperti pada algoritma BFS.



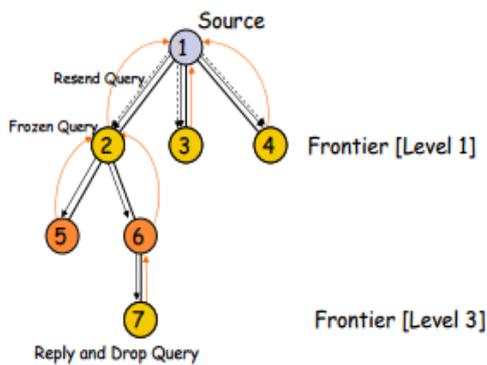
Ilustrasi Hubungan Antara Simpul dan Data Torrent (sumber: <http://ww2.cs.fsu.edu/~jungkim/figures/>)

Menurut Xiuqi Li dan Jie Wu dari Universitas Florida Atlantic, sistem dapat memilih simpul yang dikategorikan baik ataupun tidak. Definisi simpul yang baik didapat dari data statistik terhadap simpul-simpul tetangganya. Misalnya jumlah keluaran yang dikembalikan terhadap pesan *query* dari simpul tersebut. Atau pun *latency* jaringan simpul tersebut.

Oleh karena itu, pemilihan simpul yang baik dapat dipilih dengan menggunakan batasan (*heuristic*) sebagai berikut:

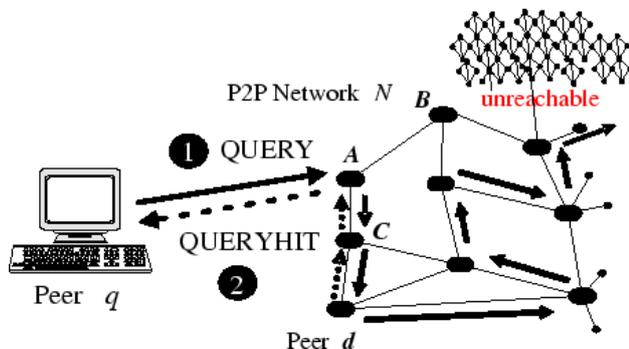
- Jumlah kembalian dari pesan *query* dari proses sebelumnya
- Waktu pengiriman terkecil terhadap pesan *query*
- Jumlah pesan terbanyak

P={1,3}



Contoh pengiriman pesan query terhadap simpul (sumber gambar: <http://www.cs.gmu.edu/~setia/cs699/P2P-Search.pdf>)

Karena adanya ketergantungan terhadap data statistik, maka metode pencarian bisa dimodifikasi dan dikembangkan. Menurut V. Kalogeraki, D. Gunopulos dan D. Zeinalipour-yazti dalam karyanya yang berjudul "A Local Search Mechanism For Peer-to-Peer Networks", metode pencarian dengan BFS dapat dikembangkan menjadi metode pencarian pintar (*intelligent search*). Pesan *query* menjadi *query* kunci dalam lingkungan sistem. Pencarian terhadap dokumen-dokumen dilakukan dengan penelusuran terhadap file-file yang dimiliki simpul tetangga dengan berpatokan pada daftar *query* kunci yang hendak dicari.



Ilustrasi Pencarian Simpul Dengan Query yang Dicari (sumber: <http://alumni.cs.ucr.edu/~csyiazti/papers/>)

Ketika simpul meminta suatu layanan, pesan *query*-nya tidak langsung disebarkan terhadap semua simpul tetangga. Sistem akan melihat data statistik terhadap proses yang telah dilakukan sebelumnya dan memilih simpul yang telah mengembalikan nilai *query* yang mirip dengan *query* yang diminta oleh simpul awal.

Selanjutnya pencarian dilakukan terhadap simpul-simpul yang terpilih. Ketika pencarian mendarang terhadap simpul terpilih, pencarian akan dilakukan terhadap data lokal tersimpannya file. Jika data yang dimaksud terdapat pada simpul tersebut, pencarian berhenti dilakukan dan langsung mengembalikan nilai terhadap simpul akar.

Jika data yang dimaksud belum terdapat pada simpul tersebut, maka pencarian selanjutnya dilakukan sampai ditemukannya data yang dimaksud atau ketika waktu hidup (*time to live, TTL*) telah habis.

IV. ANALISIS

Algoritma BFS yang diaplikasikan pada metode pencarian terhadap sistem torrent cocok jika dilihat dari struktur lingkungannya. Karena setiap simpul-simpulnya memiliki keterkaitan terhadap simpul-simpul lainnya yang berada di dalam *swarm*, algoritma BFS dapat langsung melacak terhadap simpul-simpul tetangga secara merata. Hal ini lebih menguntungkan dibandingkan pengaplikasian terhadap algoritma DFS karena peluang ditemukannya simpul solusi lebih besar.

Algoritma DFS juga dapat dipakai dan diimplementasikan pada pencarian di sistem torrent ini. Namun performa yang dihasilkan kurang efektif, mengingat sifatnya yang menelusuri simpul sampai ke kedalaman tertentu. Penelusuran jenis ini akan lebih 'melelahkan' dan mengambil ongkos yang lebih besar daripada pengaplikasian algoritma BFS.

V. KESIMPULAN

Metode pencarian terhadap sistem torrent maupun terhadap sistem jaringan *peer to peer* lebih cocok memakai algoritma BFS daripada algoritma DFS. Pengaplikasian algoritma BFS beserta algoritma pengembangannya yang memperhatikan data-data statistik lebih efektif dalam pencarian data/file dari setiap pengguna yang memiliki file yang dicari (*seeder*).

REFERENSI

- [1] V. Kalogeraki, D. Gunopulos, and D. Zeinalipour-yazti, "A local search mechanism for peer-to-peer networks", Proc. of the 11th ACM Conference on Information and Knowledge
- [2] B. Yang, and H. Garcia -Molina, "Improving search in peer-to-peer networks", Proc. of the 22nd B. Smith, "An approach to graphs of linear forms (Unpublished work style)," unpublished.
- [3] Banaei-Kashani, Farnoush and Cyrus Shahabi. Criticality-based Analysis and Design of Unstructured Peer-to-Peer Networks as "Complex Systems." Proc. IEEE/ACM CCGRID'03, pp. 351-358, 2003.
- [4] UBC Computer Science Undergraduate, 16 Mei 2014, 20:30. Tersedia dari <http://www.ugrad.cs.ubc.ca/~cs421/hw/5/a.pdf>
- [5] Munir, Rinaldi. "Diktat Kuliah Strategi Algoritma"
- [6] G.H.L. Fletcher, H.A.Sheth, and K.Borner, "Unstructured Peer-to-Peer Networks: Topological Properties and Search Performance".
- [7] L.Xiuqi and W.Jie, "Searching Technique in Peer-to-Peer Networks"
- [8] Net Beginner, How Torrent Downloading Works, 17 Mei 2014, 21:22. Tersedia dari netforbeginners.about.com
- [9] University Princeton, Breadth First Search, 17 Mei 2014, 19:49. Tersedia dari www.princeton.edu
- [10] Computerworld, Peer-to-Peer Network, 18 Mei 2014, 17:36. Tersedia dari www.computerworld.com
- [11] Webopedia, What is peer-to-peer architecture?, 18 Mei 2014, 18:34. Tersedia dari www.webopedia.com
- [12] Gufon Rajo Kaciak, Artikel Pengertian Jaringan Peer To Peer, 17 Mei 2014, 11:55. Tersedia dari dosen.gufon.com

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, 18 Mei 2014

A handwritten signature in black ink, appearing to be 'Andarias Silvanus', written in a cursive style.

Andarias Silvanus, 13512022