

# Graf dan Algoritma pada Google PageRank

Ray Andrew Obaja Sinurat - 13515073  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
13515073@std.stei.itb.ac.id

**Abstract**—Google Pagerank adalah algoritma paling utama dari *search engine* Google yang paling utama. Algoritma paling sederhana dari Google Pagerank dengan membuat sistem penomoran (*ranking*) berdasarkan banyaknya halaman yang menuju halaman utama.

**Keywords**—Google, Pagerank, Graf, Webgraph, Algorithm.

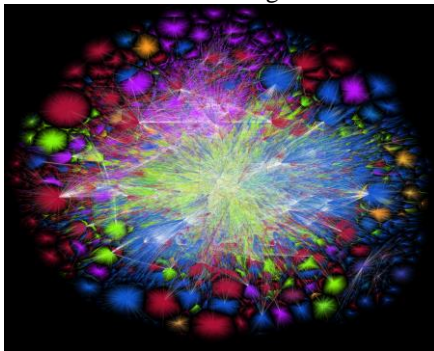
## I. PENDAHULUAN

PageRank adalah algoritma yang digunakan oleh Google *Search Engine* untuk memberikan system penilaian berbasis *rank* terhadap *webpage-webpage* yang ada di *internet*. Nama Page dari PageRank diambil dari pendiri Google yaitu Larry Page.

PageRank bekerja dengan cara menghitung banyak dan kualitas dari suatu *link* ke suatu *page* untuk menentukan secara kasar seberapa pentingnya situs *web* tersebut. <sup>[1]</sup>

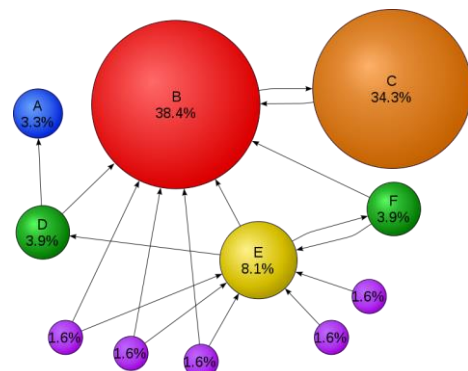
Google masih memiliki algoritma lain untuk menelusuri dan membuat sistem *ranking* dari suatu website, namun PageRank merupakan algoritma pertama dan yang paling terkenal.

Dunia *Web* merupakan dunia yang sangat luas dan sangat bebas untuk dieksplorasi, bahkan sampai ke dalam *deep-web*. *Web* juga sangat identik dengan sebuah teori dalam matematika diskrit yaitu graf. Contohnya saja sistem pemetaan yang dibuat oleh [www.opte.org](http://www.opte.org) untuk memetakan dunia *internet* secara global.



Gambar 1.1 WebGraph using LGL graph engine  
(sumber : <http://www.opte.org/the-internet/>)

PageRank juga menggunakan teori graf dalam merancang algoritma yang simpel. Graf digunakan untuk memetakan *webpage-webpage* yang terkait satu sama lain dengan menggunakan *link*.



Gambar 1.1 Contoh PageRank dari *webpages*  
(sumber : <https://en.wikipedia.org/wiki/PageRank>)

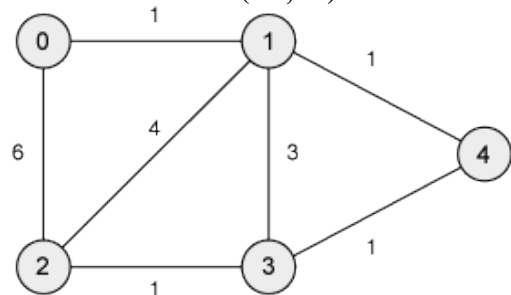
## II. DASAR TEORI

### A. Definisi Graf

Graf merupakan struktur diskrit yang digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut.

Secara matematis, graf dapat didefinisikan sebagai suatu pasangan himpunan dari himpunan tak kosong dari simpul ( $V$ /Vertex) dan sisi ( $E$ /Edges) yang menghubungkan simpul satu dengan yang lainnya. Suatu graf dapat ditulis dengan notasi sebagai berikut :

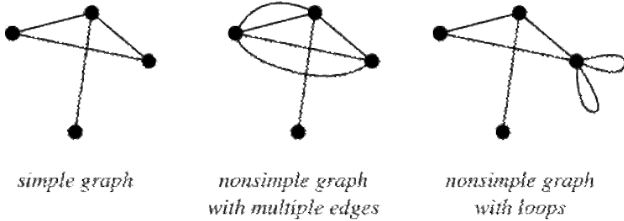
$$G = (V, E)$$



Gambar 2.1. *Weighted Graph* atau Graf Berbobot  
(sumber : <https://www.codementor.io/rishabhdaal/tutorials/graph-algorithms-interview-questions-du1085u8l>)

## B. Jenis-jenis graf

Berdasarkan ada tidaknya gelang atau sisi ganda pada suatu graf, maka graf dapat digolongkan menjadi dua jenis. Pertama adalah graf sederhana, yaitu graf yang tidak mengandung sisi ganda maupun gelang. Kedua adalah graf tak-sederhana, yaitu graf yang memiliki sisi ganda ataupun gelang.



Gambar 2.2 Graf sederhana dan Graf tak-sederhana

(sumber :

<https://progmoahamedali.wordpress.com/2013/05/01/introduction-to-graphs/>)

Berdasarkan orientasi arah pada sisi, graf dapat dibagi menjadi dua. Pertama adalah graf tak-berarah, yaitu graf yang sisinya tidak memiliki arah. Kedua adalah graf berarah, yaitu graf yang sisinya memiliki orientasi arah sehingga sisi dari simpul A menuju simpul B berbeda dengan sisi dari simpul B menuju simpul A.

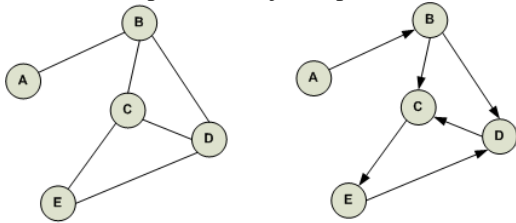


Fig 1. Undirected Graph

Fig 2. Directed Graph

Gambar 2.2 Graf tak berarah dan graf berarah

(sumber : <http://www.codediesel.com/algorithms/building-a-graph-data-structure-in-php/>)

Terdapat juga graf semu, yaitu graf tak-berarah yang memiliki sisi ganda ataupun sisi gelang. Graf kosong adalah graf yang himpunan sisinya adalah kosong (tidak memiliki sisi).

Berdasarkan jumlah simpul atau *node*, graf dapat dibagi menjadi dua. Pertama adalah graf berhingga, yaitu graf yang jumlah simpulnya,  $n$ , berhingga. Kedua adalah graf tak-berhingga, yaitu graf yang jumlah simpulnya tak-hingga. *Webgraph* dapat dikatakan graf tak-hingga mengingat dapat munculnya halaman atau *website-website* baru seiring dengan pertambahan waktu.

## C. Representasi Graf

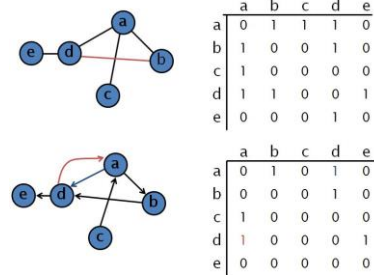
Graf dapat direpresentasikan dalam tiga cara untuk memudahkan komputer dalam membaca dan mengaplikasikan graf.

### 1. Adjacency Matrix (Matriks Ketetanggaan)

Matriks dengan ukuran  $N \times N$ .

Bila matrix tersebut dinamakan  $A = [a_{ij}]$  maka :

$a_{ij} = 1$  jika simpul  $i$  dan  $j$  bertetangga  
 $= 0$  jika simpul  $i$  dan  $j$  tidak bertetangga



Gambar 2.2 Adjacency Matrix

(sumber : <http://sourcecodemania.com/graph-implementation-in-cpp/>)

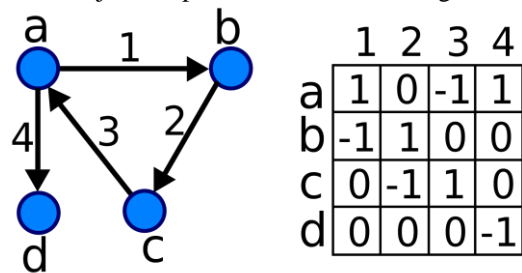
### 2. Incidency Matrix (Matriks Bersisian)

Matriks dengan ukuran  $N \times N$ .

Bila matrix tersebut dinamakan  $A = [a_{ij}]$  maka :

$a_{ij} = 1$  jika simpul  $i$  bersisian dengan sisi  $j$

$= 0$  jika simpul  $i$  tidak bersisian dengan sisi  $j$



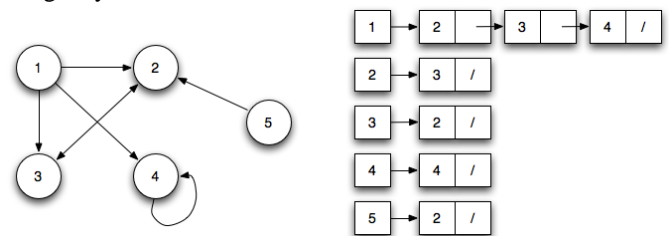
Gambar 2.3 Incidency Matrix

(sumber :

[https://commons.wikimedia.org/wiki/File:Incidence\\_matrix\\_-\\_directed\\_graph.svg](https://commons.wikimedia.org/wiki/File:Incidence_matrix_-_directed_graph.svg))

### 3. Adjacency List (Senarai Ketetanggaan)

Setiap simpul, jika memiliki tetangga, maka akan ada list yang berisi simpul-simpul yang bertetangga dengannya.



Gambar 2.4 Adjacency List

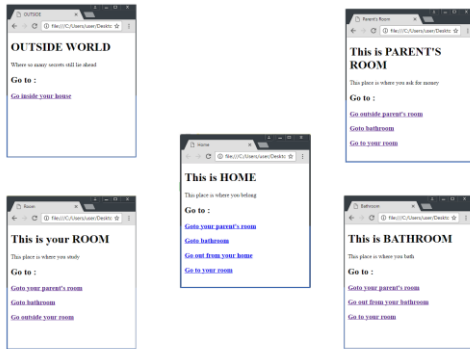
(sumber : <http://ycpcs.github.io/cs360-spring2015/lectures/lecture15.html>)

### III. GRAF DAN ALGORITMA GOOGLE PAGERANK

#### A. Graf

Seperti telah dikemukakan, *webpage* dapat direpresentasikan sebagai graf berarah. Namun, untuk *world wide web* graf yang direpresentasikan adalah graf berarah yang tak-berhingga.

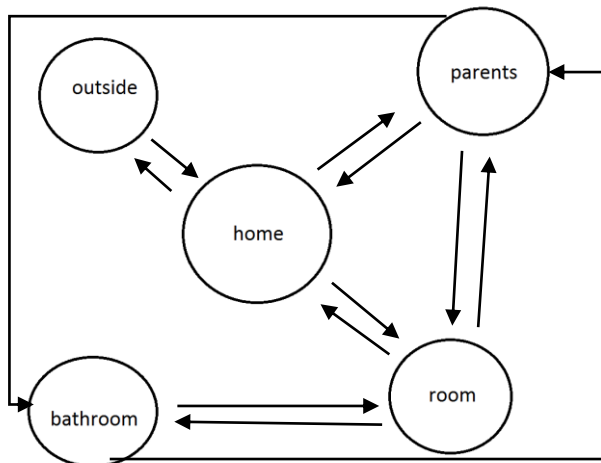
Graf berhingga sangat cocok dengan *webpage* dikarenakan oleh adanya *link* atau tautan yang sifatnya bisa saja tidak reversible. Sehingga digunakan graf berarah sebagai representasi terbaik dari *webpage*.



Gambar 3.1 Contoh *webpages* (source : dokumen pribadi)

	Home	Outside	Parents	Bathroom	Room
Home	0	1	1	1	1
Outside	1	0	0	0	0
Parents	1	0	1	1	1
Bathroom	1	0	1	1	1
Room	1	0	1	1	1

Gambar 3.2 Graf *webpage* pada *adjacency matrix* (source : dokumen pribadi)



Gambar 3.3 Graf berarah pada *webpage* (source : dokumen pribadi)

#### B. Algoritma

PageRank memiliki Algoritma seperti berikut :

$$PR(A) = (1-d) + d (PR(T1)/C(T1) + \dots + PR(Tn)/C(Tn))$$

Keterangan :

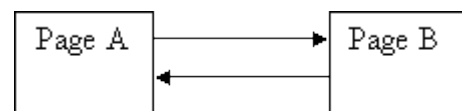
- PR(A) adalah PageRank dari *page* A menunjukkan seberapa pentingnya *page* A terhadap *page-page* lainnya
- C(Tn) adalah banyaknya link yang menuju ke *page* A dari *page* T1..Tn
- PR(Tn)/C(Tn) adalah porsi PageRank *page* A yang didapatkan dari PageRank *page* Tn yang memuat sebanyak "C(Tn)" *link* atau tautan ke *page* A
- d adalah nilai yang menahan laju penjumlahan *page* (*damped down*) agar tidak ada *page* yang memberikan terlalu banyak porsi terhadap PageRank A, d biasanya bernilai 0.85 (disebut juga *damping factor*)
- (1-d) adalah keunikan dari PageRank dimana (1-d) berarti penjumlahan dari setiap *webpages* akan tetap bernilai satu

Tujuan dari algoritma ini agar mendapatkan yang disebut *normalized probability distribution* dimana rata-rata dari semua *page* akan bernilai 1. Jika belum 1, maka PR akan diiterasi sampai rata-rata semua *page* bernilai 1.

Secara harfiah, PageRank merupakan sebuah sistem *voting* dimana setiap *page* yang menyumbang *link* atau tautan akan memberikan porsi *voting* masing-masing terhadap PageRank *link* yang dituju.

Hal ini lah menyebabkan nilai PageRank dari setiap *page* dapat berbeda-beda tergantung ada beberapa *link* atau tautan yang menuju dirinya.

#### C. Kalkulasi PageRank



Gambar 3.4 Graf *webpage* masing-masing 1 *outgoing-link* (source :

[www.cs.princeton.edu/~chazelle/courses/BIB/pagerank.htm](http://www.cs.princeton.edu/~chazelle/courses/BIB/pagerank.htm))

Dari penjelasan algoritma di upabab B, maka kita mengetahui :

$$C(A) = 1$$

$$C(B) = 1$$

$$d = 0.85$$

Kalkulasi :

Asumsikan saja PR awal dari A dan B masing-masing 1 karena kita tidak mengetahui nilai awalnya.

Sehingga :

$$PR(A) = 0.15 + 0.85 * 1 = 1$$

$$PR(B) = 0.15 + 0.85 * 1 = 1$$

Apakah perhitungan diatas sebuah kebetulan?

Ternyata kita mendapat nilai rata-rata yang sangat tepat seperti telah dijelaskan pada upabab B, yaitu 1. Namun apakah kita bisa langsung membuat asumsi terbaik nilai PR awal dari A dan B?

Karena asumsi yang belum jelas, alangkah baiknya jika kita mencoba tips iterasi berikut ini. Perhitungan PageRank dimulai dari C(Tn) dan PR(Tn) (hal ini juga didasari karena kita tidak memiliki nilai awal dari PR(Tn)) bernilai 0 sehingga didapat PR(A) bernilai 0.15 karena :

$$PR(A) = (1-0.85) + d * 0 = 0.15$$

0.15 merupakan *next guess* terbaik yang bisa didapatkan saat ini untuk PR(B), dikarenakan nilai PR(A) sudah didapat.

$$PR(B) = (1-0.85) + 0.85 * 0.15 = 0.2275$$

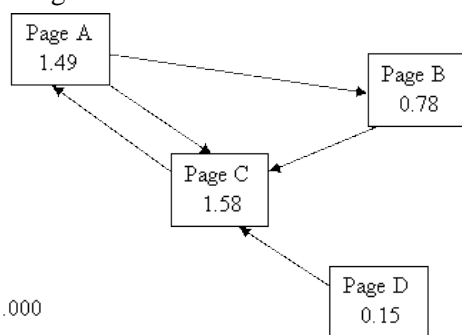
Berlanjut,

$$PR(A) = 0.15 + 0.85 * 0.2275 = 0.385875$$

$$PR(B) = 0.15 + 0.85 * 0.385875 = 0.4799375$$

Dan akan terus berlanjut menuju 1! Ternyata iteratif ini akan manjur dilakukan, meskipun kita berasumsi bahwa PR awal dari *page A* dan *B* lebih besar dari 1. Hal ini terjadi karena ada *dumping factor* yang akan terus menuju 1, meskipun seberapa besarnya PageRank awal dari setiap *page*.

#### D. Graf dan PageRank



Gambar 3.4 Graf *webpage*

(source : <http://www.sirgroane.net/google-page-rank/>)

Dari gambar di atas, yang dapat langsung dihitung secara cepat adalah PageRank dari *page D*. Hal ini dikarenakan *page D* tidak memiliki *backlink* atau tautan yang menuju dirinya. Sehingga :

$$PR(D) = 0.15 + 0.85 * 0 = 0.15$$

Selanjutnya gunakan rumus PageRank untuk setiap *page* dengan menggunakan algoritma di upabab B.

Hal ini dapat dipermudah dengan menggunakan bantuan *program* atau *script* yang banyak tersedia di internet.

Sehingga dari hasil pemrosesan secara iteratif dengan menggunakan program, didapatkan hasil :

$$PR(A) = 1.49$$

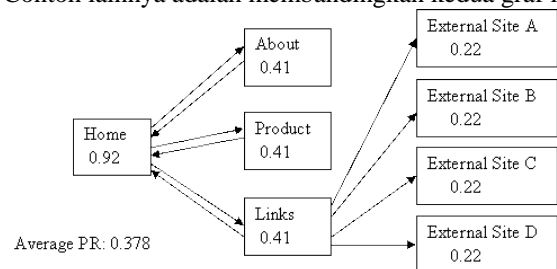
$$PR(B) = 0.78$$

$$PR(C) = 1.58$$

$$PR(D) = 0.15$$

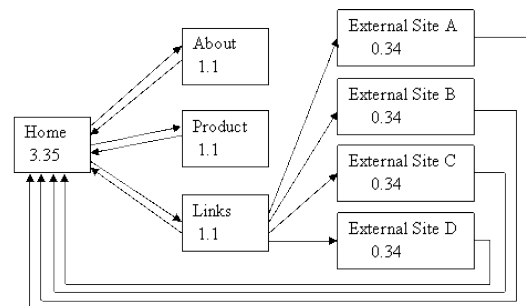
Yang didapatkan rata-ratanya senilai 1. Hal ini sudah sesuai dengan tujuan algoritma yang dikemukakan pada upabab B.

Contoh lainnya adalah membandingkan kedua graf ini



Gambar 3.5 Graf *webpage* tanpa optimisasi

(source : <http://www.sirgroane.net/google-page-rank/>)



Average PR: 1.000

Gambar 3.6 Graf *webpage* dengan menambahkan *edge*

(source : <http://www.sirgroane.net/google-page-rank/>)

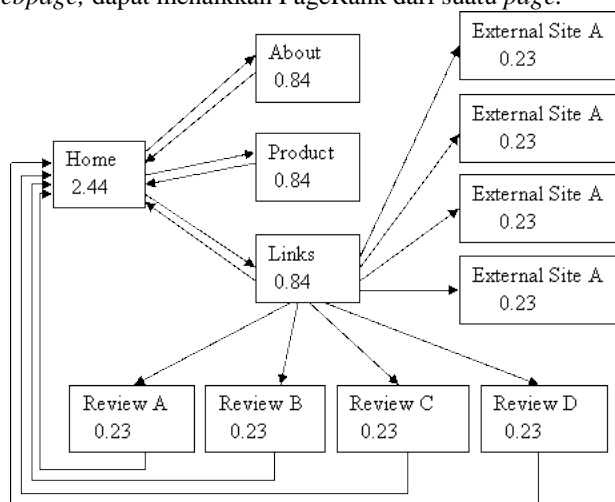
Jika pernah mendengar istilah *Search Engine Optimisation (SEO)* maka dengan mengetahui algoritma dan graf dari PageRank, kita dapat melakukan optimisasi dari *webpage-webpage* yang ada di *website* kita.

Lihat gambar 3.5 dan gambar 3.6 yang merupakan graf berarah. Meskipun graf pada gambar 3.5, rata-rata dari PageRank nya belum mencapai 1, tapi PageRank dari *page Home* sangatlah kecil, mengingat jika kita memiliki suatu *website*, pastilah kita berharap bahwa *homepage* kita memiliki PageRank yang sangat tinggi sehingga dapat menempati urutan yang cukup baik pada *Google Search*.

Ketika kita tinjau gambar 3.5 dan gambar 3.6, perbedaan yang mencolok dari kedua graf tersebut adalah adanya *edge* atau sisi baru di gambar 3.6 yaitu sisi dari *External Site A* ke *Home*, *External Site B* ke *Home*, *External Site C* ke *Home*, dan *External Site D* ke *Home*.

Dan lihat apa yang terjadi pada PageRank *page Home*! Terbukti dengan menambahkan beberapa *edge* atau sisi ke *page* yang ingin dinaikkan PageRank nya sangatlah berguna. Dalam hal ini, PageRank *page Home* naik tiga kali lipat dari nilai sebelumnya! *Edge* atau sisi pada graf *webpage* menunjukkan *link* atau tautan pada suatu *page*.

Tidak lupa juga dalam beberapa kasus, menambahkan *node* atau simpul yang pada graf ini berarti menambahkan *webpage*, dapat menaikkan PageRank dari suatu *page*.

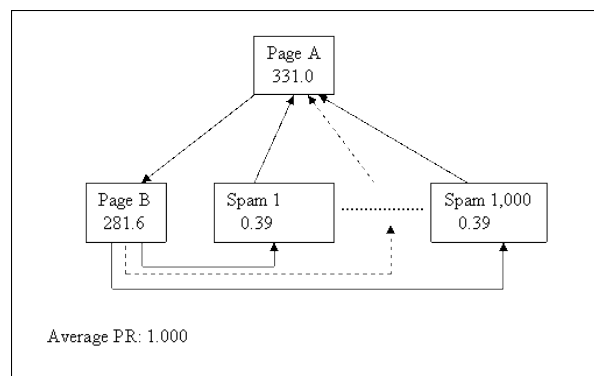


Gambar 3.7 Graf *webpage* dengan menambahkan node (source : <http://www.sirgroane.net/google-page-rank/>)

Jika kita perhatikan, gambar 3.7 menunjukkan penambahan *node* atau *webpage* baru efektif menaikkan nilai PageRank dari *Home*, hal ini juga dikarenakan adanya penambahan *edge* atau sisi baru. *Node-node* baru itu adalah *Review A*, *Review B*, *Review C*, dan *Review D*.

Terbukti, jika ada yang me-review *website* milik kita, yakinlah PageRank pasti akan bertambah!

Cara lain yang sekarang sudah banyak penjual jasanya adalah *backlink generator* dimana *link* dari *website* kita akan di-“spam” di berbagai *website* agar PageRank meningkat.



Gambar 3.7 Graf *webpage* dengan menambahkan banyak *backlink* (source : <http://www.sirgroane.net/google-page-rank/>)

Dengan cara ini, PageRank *webpage* bisa bertambah sangat tinggi! Lihat saja PageRank dari *page A* yang mencapai 331.0! Hal ini lah makanya jasa yang ditawarkan pun membutuhkan biaya, karena Google hanya mengecek *backlink-backlink* yang berkualitas dan valid yang layak dijadikan perhitungan pada PageRank.

*Website-website* yang memiliki banyak *backlink* sudah pastilah merupakan *website* yang akan “disukai” Google untuk di-*rank*. Namun, yang perlu diketahui, banyaknya *link* atau tautan bukan satu-satunya penilaian pada PageRank. Isi konten yang orisinal dan banyaknya *visitor* (termasuk *unique visitor*) sangat memengaruhi penilaian PageRank suatu *website*.

#### IV. KESIMPULAN

PageRank merupakan algoritma unik dan sangat berguna bagi setiap pemilik *website* aktif. Mengetahui graf dan algoritma pada PageRank akan menunjang peningkatan PageRank dari suatu *page* atau halaman *website*. Hal inilah yang dipergunakan banyak orang dengan istilah *Search Engine Optimisation (SEO)* dengan memetakan *backlink-backlink* yang ada di internet dan membuatnya menuju pada *link* yang ingin ditingkatkan PageRank nya.

## V. UCAPAN TERIMA KASIH

Pertama – tama saya mengucapkan terima kasih kepada Tuhan Yang Maha Esa yang telah melimpahkan berkat dan kasih-Nya sehingga makalah Matematika Diskrit ini dapat diselesaikan tepat waktu. Saya juga mengucapkan terima kasih kepada kedua orang tua saya yang selalu memberi dukungan dan doa restu kepada saya sehingga dapat menempuh pendidikan sampai saat ini. Tak lupa saya juga mengucapkan terima kasih kepada Bapak Dr. Ir. Rinaldi Munir, MT. serta Ibu Dra. Harlili S., M.Sc. yang berperan sebagai dosen mata kuliah IF 2120 Matematika Diskrit sehingga dengan ilmu pengetahuan seputar Matematika Diskrit, saya dapat membuat dan menyelesaikan makalah ini. Saya juga tidak lupa mengucapkan terima kasih kepada teman-teman dalam membantu saya menentukan topik untuk makalah ini. Terima kasih.

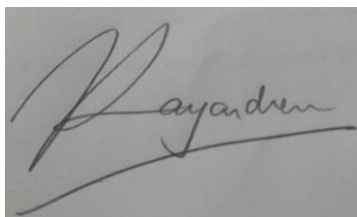
## REFERENSI

- [1] *"Facts about Google and Competition"*. Disadur original tanggal 4 November 2011 dan diakses pada 6 Desember 2016. <https://web.archive.org/web/20111104131332/https://www.google.com/competition/howgooglesearchworks.html>
- [2] Munir, Rinaldi. Matematika Diskrit Edisi 3, Penerbit Informatika, 2010.
- [3] <http://www.sirgroane.net/google-page-rank/>. Diakses tanggal 6 Desember 2016.
- [4] <http://www.cs.princeton.edu/~chazelle/courses/BIB/pagerank.htm> . Diakses tanggal 7 Desember 2016.
- [5] <http://www.math.cornell.edu/~mec/Winter2009/RalucaRemus/Lecture3/lecture3.html>. Diakses tanggal 7 Desember 2016.
- [6] <https://moz.com/blog/a-universe-of-graphs>. Diakses tanggal 7 Desember 2016.
- [7] <http://www.opte.org/the-internet>. Diakses tanggal 7 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, 8 Desember 2016



Ray Andrew Obaja Sinurat – 13515073  
Teknik Informatika 2015  
Institut Teknologi Bandung