

Laporan Penelitian

IMPLEMENTASI ALGORITMA *PAGERANK*

Karya Ilmiah untuk Melengkapi Syarat Pengajuan Edukatif
Pada Fakultas Sains dan Teknologi UIN Sumatera Utara
Program Studi Matematika

Oleh:

Rima Aprilia, M.Si

NIDN. 0130048801



**PRODI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA
MEDAN
2017**

KATA PENGANTAR

Bismillahirrahmanirrahim...

Dengan segala kerendahan hati, penulis sampaikan puji syukur kepada Allah SWT berkat Rahmat dan Hidayah-Nya memberi kesehatan, pengetahuan dan kesempatan kepada penulis sehingga dapat menyelesaikan penelitian ini yang berjudul "**Implementasi Algoritma PageRank**".

Dalam menyelesaikan penelitian ini banyak bantuan bimbingan dari berbagai pihak, baik berupa materil, spiritual, maupun informasi. Sehingga penelitian ini dapat diselesaikan. Maka selayaknya penulis mengucapkan terima kasih kepada :

1. Bapak Dr. H.M. Jamil, MA selaku Dekan Fakultas Sains dan Teknologi UIN Sumatera Utara Medan
2. Ibu Dr. Rina Filia Sari, M.Si selaku Wakil Dekan I Fakultas Sains dan Teknologi UIN Sumatera Utara Medan
3. Ibu Dr. Sajaratud Dur, MT selaku Kaprodi Matematika Fakultas Sains dan Teknologi UIN Sumatera Utara Medan sekaligus Konsultan pada penelitian ini
4. Bapak / ibu rekan-rekan dosen tetap Fakultas Sains dan Teknologi UIN Sumatera Utara Medan

Menyadari kekurangan dan keterbatasan pada penelitian ini, maka penulis tetap mengharapkan kritik dan saran dari berbagai pihak agar penelitian ini dapat dikembangkan dikemudian hari. Akhir kata semoga penelitian ini bermanfaat bagi kita semua dan Semoga Allah SWT berkenan memberkan berkahnya sehingga semua harapan dan cita-cita penulis dapat terkabulkan. Amin

Medan, September 2017

Rima Aprilia, M.Si

REKOMENDASI

Setelah membaca dan menelaah hasil penelitian yang berjudul **“Implementasi Algoritma Pagarank pada Matlab”**. Yang dilakukan oleh Rima Aprilia, M.Si maka saya berkesimpulan bahwa hasil penelitian ini dapat diterima sebagai karya tulis berupa hasil penelitian. Demikianlah rekomendasi diberikan kepada yang bersangkutan untuk dapat dipergunakan sebagaimana mestinya.

Medan, April 2018
Konsultan

Dr. Rina Filia Sari, M.Si
NIP. 197703012005012002

DAFTAR ISI

KATA PENGANTAR	ii
REKOMENDASI	iii
DAFTAR ISI	iv
ABSTRAK	v
BAB I. PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian	3
1.4 Manfaat Penelitian	3
BAB II. TINJAUAN PUSTAKA	4
2.1 <i>Pagerank</i>	4
2.2 Algoritma <i>Pagerank</i>	5
2.3 Prestasi Peringkat	6
2.4 Matlab	8
BAB III. METODOLOGI PENELITIAN	10
3.1 Tempat dan waktu penelitian	10
3.2 Manfaat Penelitian	10
BAB IV. HASIL DAN PEMBAHASAN	14
4.1 HASIL	14
4.2 Prestasi Peringkat	15
4.3 Kalkulasi <i>Pagerank</i>	17
4.4 Implementasi	18
BAB V. KESIMPULAN DAN SARAN	19
5.1 Kesimpulan	19
5.2 Saran	19
DAFTAR PUSTAKA	20
Lampiran Program <i>PageRank</i>	21

ABSTRAK

PENDAHULUAN

Algoritma PageRank adalah algoritma yang dikeluarkan dan dikembangkan oleh pendiri Google sebagai alat untuk menentukan ranking suatu web. Dalam penentuan ranking suatu web, Link masuk dan link keluar juga random surfer model merupakan salah satu faktor penentu dalam menentukan ranking suatu web. Implementasi PageRank pada komputer dibentuk pada sebuah program dimana program dikerjakan di dalam m-file menggunakan Matlab.

Kata Kunci : Algoritma PageRank, Ranking Web, Matlab

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan volume informasi di internet terus meningkat dari hari ke hari, hal ini menjadi tantangan tersendiri bagi para pemilik website untuk bisa menyajikan informasi yang tepat, relevan bagi para pengguna internet. Banyak cara digunakan search engine dalam menentukan kualitas/rangking sebuah halaman web, mulai dari penggunaan meta tags, isi dokumen, penekanan pada content dan masih banyak teknik lain atau gabungan teknik yang mungkin digunakan. *Link popularity*, sebuah teknologi yang dikembangkan untuk memperbaiki kekurangan dari teknologi lain (*Meta Keywords, Meta Description*) yang bisa dicurangi dengan halaman yang khusus di desain untuk search engine atau biasa disebut doorway pages.

Dengan algoritma *PageRank* ini, dalam setiap halaman akan diperhitungkan inbound link (link masuk) dan outbound link (link keuar) dari setiap halaman web. Sebuah situs akan semakin populer jika semakin banyak situs lain yang meletakkan link yang mengarah ke situsnya, dengan asumsi isi/content situs tersebut lebih berguna dari isi/content situs lain. *PageRank* dihitung dengan skala 0-10, di mana semakin besar nilai *PageRank* sebuah situs, maka semakin tinggi rankingnya. Pada *PageRank*, ranking tertinggi adalah 10, sementara situs-situs pemula berada pada *PageRank* 0.

Contoh: Sebuah situs yang mempunyai *PageRank* 9 akan diurutkan lebih dahulu dalam list pencarian Google daripada situs yang mempunyai *PageRank* 8 dan kemudian seterusnya yang lebih kecil.

Perubahan algoritma *PageRank* dari waktu ke waktu disesuaikan dengan perubahan hardware dan perkembangan teknologi informasi menyebabkan pengetahuan setiap orang berbeda mengenai algoritma *PageRank*. Lawrence Page dan Sergey Brin mempublikasikan algoritma *PageRank* dalam salah satu publikasinya pada www.wikipedia.org dituliskan sebagai berikut:

Algoritma *PageRank* ;

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

Salah satu algoritma lain yang dipublikasikan

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

- $PR(A)$ adalah *PageRank* halaman A
- $PR(T1)$ adalah *PageRank* halaman T1 yang mengacu ke halaman A
- $C(T1)$ adalah jumlah link keluar (*outbound link*) pada halaman T1
- d adalah damping faktor yang bisa diberi antara 0 dan 1.
- N adalah jumlah keseluruhan halaman web (yang terindeks oleh Google)

Dari Algoritma di atas dapat diketahui bahwa algoritma *PageRank* ditentukan untuk setiap halaman suatu web dari

keseluruhan halaman web yang ada. Pada dasarnya algoritma *PageRank* mengevaluasi dua hal, yaitu:

- a. Berapa link yang ada pada halaman web lain yang menuju halaman web tersebut.
- b. Bagaimana kualitas situs yang melink halaman web tersebut.

1.2 Rumusan Masalah

Permasalahan pada penelitian ini adalah mengimplementasikan algoritma *PageRank* pada web dengan menggunakan software MATLAB.

1.3 Tujuan Penelitian

Penelitian ini bertujuan untuk mengetahui penggunaan algoritma *PageRank* pada software MATLAB, penelitian dilakukan dengan memperhatikan pada faktor-faktor yang mempengaruhi kinerja dari algoritma tersebut.

1.4 Manfaat Penelitian

Dengan tercapainya tujuan penelitian di atas maka diperoleh manfaat dari penelitian. Penelitian ini menjadi referensi, menambah pengetahuan dan pemahaman bagi peneliti dan pembaca untuk memahami bagaimana implementasi dan penggunaan algoritma *PageRank* pada software MATLAB.

Bab II

TINJAUAN PUSTAKA

2.1 PageRank

PageRank adalah sebuah algoritma yang telah dipatenkan yang berfungsi menentukan situs web mana yang lebih penting / populer. *PageRank* merupakan salah satu fitur utama mesin pencari Google dan diciptakan oleh pendirinya, Larry Page dan Sergey Brin yang merupakan mahasiswa Ph.D. Universitas Stanford. Banyak cara digunakan *search engine* dalam menentukan kualitas/rangking sebuah halaman web, mulai dari penggunaan *META Tags*, isi dokumen, penekanan pada *content* dan masih banyak teknik lain atau gabungan teknik yang mungkin digunakan. *Link popularity*, sebuah teknologi yang dikembangkan untuk memperbaiki kekurangan dari teknologi lain (*Meta Keywords, Meta Description*) yang bisa dicurangi dengan halaman yang khusus di desain untuk *search engine* atau biasa disebut *doorway pages*. Dengan algoritma '*PageRank*' ini, dalam setiap halaman akan diperhitungkan *inbound link* (link masuk) dan *outbound link* (link keuar) dari setiap halaman web.

PageRank, memiliki konsep dasar yang sama dengan *link popularity*, tetapi tidak hanya memperhitungkan "jumlah" *inbound* dan *outbound link*. Pendekatan yang digunakan

adalah sebuah halaman akan dianggap penting jika halaman lain memiliki link ke halaman tersebut. Sebuah halaman juga akan menjadi semakin penting jika halaman lain yang memiliki rangking (*PageRank*) tinggi mengacu ke halaman tersebut.

Dengan pendekatan yang digunakan *PageRank*, proses terjadi secara rekursif dimana sebuah rangking akan ditentukan oleh rangking dari halaman web yang rangkingnya ditentukan oleh rangking halaman web lain yang memiliki link ke halaman tersebut. Proses ini berarti suatu proses yang berulang (rekursif). Di dunia maya, ada jutaan bahkan milyaran halaman web. Oleh karena itu sebuah rangking halaman web ditentukan dari struktur link dari keseluruhan halaman web yang ada di dunia maya. Sebuah proses yang sangat besar dan kompleks.

2.2. Algoritma Pagerank

Pagerank adalah distribusi probabilitas yang digunakan untuk mewakili kemungkinan bahwa seseorang secara acak mengklik link akan tiba pada suatu halaman tertentu. Pagerank dapat dihitung untuk koleksi dokumen dari berbagai ukuran. Hal ini diasumsikan dalam beberapa penelitian bahwa distribusi yang merata dibagi di antara semua dokumen dalam koleksi pada awal proses komputasi. Perhitungan Pagerank melewati "iterasi", melalui koleksi untuk menyesuaikan nilai Pagerank perkiraan untuk lebih mencerminkan nilai sebenarnya. Sebuah probabilitas dinyatakan sebagai nilai numerik antara 0 dan 1. Sebuah probabilitas 0,5 umumnya dinyatakan sebagai "peluang 50%". Oleh karena itu, Pagerank dari

Salah satu algoritma lain yang dipublikasikan

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

- $PR(A)$ adalah *PageRank* halaman A
- $PR(T1)$ adalah *PageRank* halaman T1 yang mengacu ke halaman A
- $C(T1)$ adalah jumlah link keluar (*outbound link*) pada halaman T1
- d adalah damping faktor yang bisa diberi antara 0 dan 1.
- N adalah jumlah keseluruhan halaman web (yang terindeks oleh Google)

Dari algoritma di atas dapat dilihat bahwa *PageRank* ditentukan untuk setiap halaman bukan keseluruhan situs web. *PageRank* sebuah halaman ditentukan dari *PageRank* halaman yang mengacu kepadanya yang juga menjalani proses penentuan *PageRank* dengan cara yang sama, jadi proses ini akan berulang sampai ditemukan hasil yang tepat.

Akan tetapi *PageRank* halaman A tidak langsung diberikan kepada halaman yang dituju, akan tetapi sebelumnya dibagi dengan jumlah link yang ada pada halaman T1 (*outbound link*), dan *PageRank* itu akan dibagi rata kepada setiap link yang ada pada halaman tersebut. Demikian juga dengan setiap halaman lain "Tn" yang mengacu ke halaman "A".

Setelah semua Pagerank yang didapat dari halaman-halaman lain yang mengacu ke halaman "A" dijumlahkan, nilai itu kemudian dikalikan dengan damping factor yang bernilai antara 0 sampai 1. Hal ini dilakukan agar tidak keseluruhan nilai Pagerank halaman T didistribusikan ke halaman A.

2.4. MATLAB

Matlab adalah singkatan dari Matrix Laboratory, diciptakan pada akhir tahun 1970-an. Kekuatan Matlab terletak pada:

1. Kemudahan manipulasi struktur matriks.
2. Jumlah routine-routine powerful yang berlimpah yang terus berkembang.
3. Kekuatan fasilitas grafik tiga dimensi yang sangat memadai.
4. Sistem scripting yang memberikan keleluasaan bagi pengguna untuk mengembangkan dan memodifikasi *software* untuk kebutuhan sendiri.
5. Kemampuan *interface* (misal dengan bahasa c, word dan matematika)
6. Dilengkapi dengan toolbox, simulink, stateflow dan sebagainya, serta mulai melimpahkan *source code* di internet yang dibuat dalam Matlab (contoh toolbox misalnya: *signal processing, control system, neural networks* dan sebagainya).

Semua itu merupakan perangkat yang powerful untuk menyelesaikan permasalahan sains dan teknik terutama untuk wilayah dimana komputasi numeric harus dibuat.

2.4.1 Fungsi M.File

Penulisan barisan ekspresi dalam MATLAB *command window* biasanya dilakukan baris perbaris dan biasanya untuk menyimpan barisan perintah dan hasil outputnya dengan menggunakan *command diary*. Hal ini sangatlah tidak efisien dikarenakan barisan yang telah tersimpan di *diary* tidak dapat dipanggil kembali seandainya telah keluar dari MATLAB. Untuk itu MATLAB menyediakan suatu struktur untuk membuat fungsi yang diinginkan atau suatu tehnik pemograman dalam bentuk m-file.

M-file selain dipakai sebagai penamaan file juga bisa dipakai untuk menamakan fungsi, sehingga fungsi yang di buat di *command window* bisa di simpan dengan ekstensi .m sama dengan file yang kita panggil di *command window*. Saat kita menggunakan fungsi MATLAB seperti *inv*, *abs*, *cos*, *sin* dan *sqrt*, MATLAB menerima variabel yang kita berikan.

Fungsi m-file mirip dengan *script* file dimana keduanya merupakan file teks dengan ekstensi .m sebagaimana *script* m-file, fungsi m-file tidak dimasukkan dalma *command window* tetapi pada file tersendiri yang dibuat dengan editor teks.

BAB III

METODOLOGI PENELITIAN

3.1 Tempat dan Waktu Penelitian

Tempat penelitian di Universitas Islam Negeri Sumatera Utara Meda.. Waktu kegiatan penelitian dimulai bulan Juni 2016- Januari 2017

3.2 Metode dan Desain Penelitian

1. Metode Penelitian

Metode penelitian ini adalah Studi literatur.

2. Desain Penelitian

a. Studi Literatur

Studi literatur adalah mencari referensi teori yang relevan dengan kasus atau permasalahan yang ditemukan. Referensi tersebut berisikan tentang:

1. Algoritma secara umum.
2. Perumusan PageRank
3. Cara kerja PageRank
4. Penggunaan *Software* Matlab.

Referensi ini dapat dicari dari buku, jurnal, artikel laporan penelitian, dan situs-situs di internet. Output dari studi literatur ini adalah terkoleksinya referensi yang relevan dengan perumusan masalah. Tujuannya adalah untuk memperkuat permasalahan serta sebagai dasar teori dalam melakukan studi dan juga menjadi dasar untuk melakukan desain dan implementasi Algoritma PageRank.

b. Pengumpulan Data

Data sekunder yang dibutuhkan untuk dalam penelitian ini adalah :

1. Proses perhitungan Google PageRank.
2. Cara kerja Google PageRank

Data ini dapat diperoleh dengan studi literatur Hasil dari pengumpulan data ini bisa dipakai sebagai bahan untuk membuat desain program Algoritma PageRank dan visualisasinya.

c. Desain Program

Desain program PageRank menggunakan *Software* Matlab. Program tersebut dibagi dalam beberapa prosedur. Program dikerjakan di dalam m-file pada *software* Matlab, kemudian program disimpan pada folder yang diinginkan dalam file yang bernama PageRank.m.

Output dari desain ini adalah simulasi sederhana bagaimana proses Algoritma PageRank pada web dalam penentuan ranking web.

d. Menjalankan Program

Program simulasi yang telah jadi dari pendesainan dirunning dengan memasukkan variable input yang dibutuhkan program. Input ini dapat divariasikan sesuai dengan keinginan pengguna program.

e. Hasil dan Pembahasan

Dari output program yang telah dibuat, perlu dilakukan analisa terlebih dahulu. Apakah program telah sesuai dengan perencanaan, jika tidak, maka harus dicek kembali parameter-parameter desain program yang dimasukkan dalam listing program *Software Matlab*. Kalau data output telah sesuai dengan perencanaan maka data tersebut tinggal dianalisa.

Pada hasil dan pembahasan akan dijelaskan cara kerja algoritma *PageRank* dan diberikan ilustrasi sehingga akan lebih mudah untuk dipahami cara kerja algoritma *PageRank*.

f. Kesimpulan dan Saran

Pada tahap ini merupakan proses untuk menarik kesimpulan dan saran atas apa yang dilakukan selama pengerjaan penelitian. Dasar

pengambilan kesimpulan dan saran diantaranya adalah hasil analisa dan pembahasan. Dan bisa berisikan rekomendasi teknis yang berhubungan dengan Implementasi Algoritma PageRank pada *Software Matlab*.

Pagerank adalah distribusi probabilitas yang digunakan untuk mewakili kemungkinan bahwa seseorang secara acak mengklik link akan tiba pada suatu halaman tertentu. *Pagerank* dapat dihitung untuk koleksi dokumen dari berbagai ukuran. Hal ini diasumsikan dalam beberapa penelitian bahwa distribusi yang merata dibagi di antara semua dokumen dalam koleksi pada awal proses komputasi.

Perhitungan *Pagerank* melewati "iterasi", dilakukan koleksi untuk menyesuaikan nilai *Pagerank* perkiraan untuk lebih mencerminkan nilai sebenarnya. Sebuah probabilitas dinyatakan sebagai nilai numerik antara 0 dan 1. Sebuah probabilitas 0,5 umumnya dinyatakan sebagai "peluang 50%". Oleh karena itu, *Pagerank* dari 0,5 berarti ada kemungkinan 50% bahwa seseorang mengklik pada link acak akan diarahkan ke dokumen dengan *Pagerank* 0,5 [8].

In Link 1 : adalah *hyperlink* yang mengarah ke hal. 1 dari halaman lain.

Out Link 1 : adalah *hyperlink* yang menunjuk ke halaman lain dari hal. 1.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil

Pagerank adalah distribusi probabilitas yang digunakan untuk mewakili kemungkinan bahwa seseorang secara acak mengklik link akan tiba pada suatu halaman tertentu. *Pagerank* dapat dihitung untuk koleksi dokumen dari berbagai ukuran. Hal ini diasumsikan dalam beberapa penelitian bahwa distribusi yang merata dibagi di antara semua dokumen dalam koleksi pada awal proses komputasi.

Perhitungan *Pagerank* melewati "iterasi", melalui koleksi untuk menyesuaikan nilai *Pagerank* perkiraan untuk lebih mencerminkan nilai sebenarnya. Sebuah probabilitas dinyatakan sebagai nilai numerik antara 0 dan 1. Sebuah probabilitas 0,5 umumnya dinyatakan sebagai "peluang 50%". Oleh karena itu, *Pagerank* dari 0,5 berarti ada kemungkinan 50% bahwa seseorang mengklik pada link acak akan diarahkan ke dokumen dengan *Pagerank* 0,5 [8].

In Link i : adalah hyperlink yang mengarah ke hal. i dari halaman lain.

Out Link i : adalah hyperlink yang menunjuk ke halaman lain dari hal. i .

4.2 Prestasi Peringkat:

1. Sebuah hyperlink dari halaman menunjuk ke halaman lain adalah alat angkut implisit yang memiliki otoritas ke halaman target.
2. Sebuah halaman dengan tinggi skor prestasi yang menunjuk ke i adalah lebih penting daripada halaman dengan skor prestasi lebih rendah menunjuk ke i .

Menurut prestasi peringkat, pentingnya skor halaman i ditentukan dengan menjumlahkan nilai Pagerank dari semua halaman yang mengarah ke i . Karena Halaman i dapat menunjuk ke halaman lain, nilai prestasi harus dibagi di antara semua halaman yang menunjuk ke halaman lain.

Dari pendekatan yang sudah dijelaskan sebelumnya mengenai konsep Pagerank, algoritma Pagerank dapat dirumuskan seperti di bawah ini :

Berapa nilai PR yang benar adalah hak hitung dari google

Rumus Pagerank Calculator adalah :

Algoritma awal

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

Salah satu algoritma lain yang dipublikasikan

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

PR(A) adalah Pagerank halaman A

PR(T1) adalah Pagerank halaman T1 yang mengacu ke halaman A

C(T1) adalah jumlah link keluar (outbound link) pada halaman T1

d adalah damping factor yang bisa diberi antara 0 dan 1.

N adalah jumlah keseluruhan halaman web (yang terindeks oleh Google).

Dari algoritma di atas dapat dilihat bahwa Pagerank ditentukan untuk setiap halaman bukan keseluruhan situs web. Pagerank sebuah halaman ditentukan dari Pagerank halaman yang mengacu kepadanya yang juga menjalani proses penentuan Pagerank dengan cara yang sama, jadi proses ini akan berulang sampai ditemukan hasil yang tepat. Akan tetapi Pagerank halaman A tidak langsung diberikan kepada halaman yang dituju, akan tetapi sebelumnya dibagi dengan jumlah link yang ada pada halaman T1 (outbound link), dan Pagerank itu akan dibagi rata kepada setiap link yang ada pada halaman tersebut. Demikian juga dengan setiap halaman lain "Tn" yang mengacu ke halaman "A".

Setelah semua Pagerank yang didapat dari halaman-halaman lain yang mengacu ke halaman "A" dijumlahkan, nilai itu kemudian dikalikan dengan damping factor yang bernilai antara 0 sampai 1. Hal ini dilakukan agar tidak keseluruhan nilai Pagerank halaman T didistribusikan ke halaman A.

4.3. Kalkulasi Pagerank

Nilai Pagerank tinggi masih merupakan faktor umum untuk menilai otoritas sebuah website.

Tabel . Perhitungan PR secara manual

PR	Links for PR3	Links for PR4	Links for PR5	Links for PR6	Links for PR7	Links for PR 8
PR 1	555	3.055	16.803	92.414	508.277	2.795.522
PR 2	101	555	3.055	16.803	92.414	508.277
PR 3	18.5	101	555	3.055	16.803	92.414
PR 4	3.5	18.5	101	555	3.055	16.803
PR 5	1	3.5	18.5	101	555	3.055
PR 6	0.5	1	3.5	18.5	101	555
PR 7	0.5	0.5	1	3.5	18.5	101
PR 8	0.5	0.5	0.5	1	3.5	18.5
PR 9	0.5	0.5	0.5	0.5	1	3.5
PR 10	0.5	0.5	0.5	0.5	0.5	1

Dasar Pagerank cukup mudah, yang perlu diketahui dalam perhitungan pr adalah:

1. Berapa incoming link ?
2. Berapa banyak total halaman lain terhubung ke website anda?
3. Bagaimana PR hal website yg terlink ke website anda?

Contoh kalkulasi Pagerank sederhana:

Mari kita berasumsi bahwa seluruh web hanya terdiri atas empat website A, B, C, dan D, masing-masing memiliki nilai *PageRank* "1". Jumlahnya sama dengan jumlah website. Website B, C dan D masing-masing memiliki sebuah link ke website A dan tidak ada link lainnya. Apabila faktor peredam diabaikan, hasilnya

adalah rumus : $PR(A) = 1/1 + 1/1$ dan $1/1$, *PageRank* website adalah 3.

Contoh yang lebih rumit: Website A memiliki link ke website B dan C. B hanya memiliki sebuah link ke A. C memiliki link ke A, B dan D. D hanya memiliki link ke B. Rumus untuk A akan menjadi $PR(A) = 1/1 + 1/3$. Link dari B bernilai 1, sementara dari C hanya $1/3$ dengan jumlah links 3, hasilnya adalah 1,33.

- Untuk B : $PR(B) = 1/2 + 1/3 + 1/1$ hasilnya 1,83
- Untuk C : $PR(C) = 1/2$ hasilnya 0,5
- Untuk D : $PR(D) = 1/3$ hasilnya 0,33

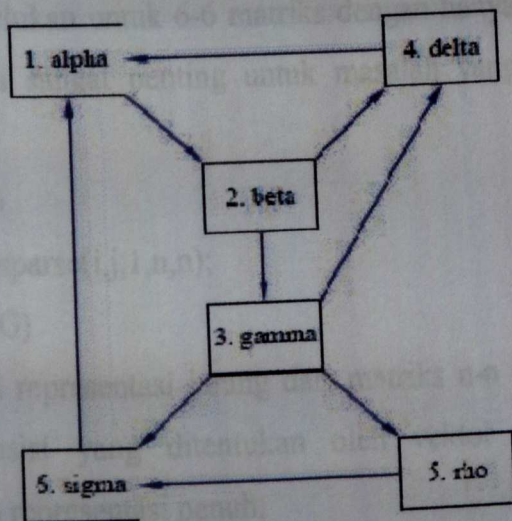
Jumlah *PageRank* website A, B, C dan D seharusnya sama dengan jumlah website $1,33 + 1,83 + 0,5 + 0,33 = 3,99$. Kekurangan 0,1 disebabkan oleh adanya pembulatan. Dalam kalkulasi ini masih ada yang kurang. *PageRank* setiap website tidak disertakan. Contoh berikutnya adalah website b. Apabila kalkulasi disesuaikan dengan *PageRank* yang didapat dari langkah pertama . $PR(B) = 1/2 + 1/3 + 1/1$. Didapat term berikut : $PR(B) = 1,33/2 + 0,5/3 + 0,33/1$ hasilnya adalah 1,62. Tentu saja kalkulasi baru *PageRank* website B mengubah *PageRank* website A, C dan D. Nilai baru website D kembali mengubah nilai website B.

4.4 Implementasi *PageRank* pada MATLAB

Implementasi *PageRank* pada MATLAB dibentuk pada sebuah program yang tujuannya menggunakan aplikasi algoritma

PageRank. Program dikerjakan di dalam m-file pada MATLAB kemudian program disimpan pada folder yang diinginkan dalam file bernama *PageRank.m*. pada saat menjalankan program, pada *command window*, diketikkan nama file yaitu *PageRank*, maka MATLAB akan membuka fungsi file dan mengkompilasi perintah-perintah didalamnya, sehingga akan muncul hasil program seperti yang ada pada lampiran.

Berikut akan ditampilkan ilustrasi program yang dirancang dengan menggunakan software MATLAB, dengan menggunakan enam buah url yaitu alpha, beta, gamma, delta, rho dan sigma.



Gambar . Ilustrasi web pada matlab

Dua jenis pengindeksan ke array sel yang mungkin. Kurung menunjukkan subarrays, termasuk sel-sel individual, dan kurung kurawal menunjukkan isi dari sel. Jika k adalah skalar, maka $U(k)$ adalah 1-1 array sel yang terdiri dari sel k di U , sedangkan U_k

adalah string dalam sel itu. Jadi U (1) adalah sel tunggal dan U1 adalah string alpha. Anggap kotak surat dengan alamat pada jalan kota. B (502) adalah kotak di nomor 502, sedangkan B502 adalah surat dalam kotak itu. Kita bisa menghasilkan konektivitas matriks dengan menetapkan pasang indeks (i, j) dari elemen nol. Karena ada Link ke beta dari alpha, yang (2,1) unsur G adalah nol. Sembilan koneksi dijelaskan oleh

$$i = [2 3 4 4 5 6 1 6 1]$$

$$j = [1 2 2 3 3 3 4 5 6]$$

Sebuah matriks jarang disimpan dalam struktur data yang membutuhkan memori hanya untuk elemen nol dan indeks mereka. Ini tidak diperlukan untuk 6-6 matriks dengan hanya 27 nol entri, tetapi menjadi sangat penting untuk masalah yang lebih besar.

Pernyataan

$$n = 6$$

$$G = \text{sparse}(i,j,1,n,n);$$

$$\text{full}(G)$$

menghasilkan representasi jarang dari matriks n-n dengan orang-orang di posisi yang ditentukan oleh vektor i dan j dan menampilkan representasi penuh.

$$0 \ 0 \ 0 \ 1 \ 0 \ 1$$

$$1 \ 0 \ 0 \ 0 \ 0 \ 0$$

$$0 \ 1 \ 0 \ 0 \ 0 \ 0$$

$$0 \ 1 \ 1 \ 0 \ 0 \ 0$$

$$0 \ 0 \ 1 \ 0 \ 0 \ 0$$

0 0 1 0 1 0

Pernyataan

$$c = \text{full}(\text{sum}(G))$$

menghitung jumlah kolom

c =

1 2 3 1 1 1

Matriks diagonal I dan D dihitung seperti sebelumnya

$$I = \text{speye}(n,n)$$

$$D = \text{spdiags}(1./c', 0, n, n)$$

Pernyataan

$$x = (I - p * G * D) \setminus (\text{delta} * e)$$

kemudian memecahkan sistim persamaan linier dan menghasilkan

x =

0.2675

0.2524

0.1323

0.1697

0.0625

0.1156

Untuk contoh yang kecil, elemen terkecil transisi markov matriks adalah $\delta = .15 = 6 = .0250$.

$$A = p * G * D + \text{delta}$$

A =

0.0250 0.0250 0.0250 0.8750 0.0250 0.8750

0.8750 0.0250 0.0250 0.0250 0.0250 0.0250

0.0250 0.4500 0.0250 0.0250 0.0250 0.0250

0.0250 0.4500 0.3083 0.0250 0.0250 0.0250

0.0250 0.0250 0.3083 0.0250 0.0250 0.0250

0.0250 0.0250 0.3083 0.0250 0.8750 0.0250

Jika diurutkan dalam *shorthes PageRank* dengan nilai masuk dan nilai keluar hasilnya adalah sebagai berikut:

PageRank	masuk	keluar	url
1 0.2675	2	1	alpha
2 0.2524	1	2	beta
4 0.1697	2	1	delta
3 0.1323	1	3	gamma
6 0.1156	2	1	sigma
5 0.0625	1	1	rho

Dapat dilihat bahwa alpha memiliki *PageRank* lebih tinggi dari delta atau sigma, bahkan meskipun semuanya memiliki jumlah link yang sama, dan beta pada peringkat kedua karena berada dalam perlindungan alpha itu. Sebuah surfer acak akan mengunjungi alpha hampir 27% dari waktu dan Rho hanya sekitar 6% dari waktu.

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian dapat ditarik kesimpulan bahwa algoritma *PageRank* dapat diimplementasikan pada software MATLAB, dalam menentukan urutan tinggi suatu web dengan algoritma *PageRank* digunakan banyak faktor yaitu model peselancar acak, filter, link masuk, link keluar dan damping faktor yang telah ditetapkan yaitu 0-1.

5.2 Saran

Pengujian masih bersifat sederhana belum menggunakan data yang lebih besar, masih dibutuhkan penelitian-penelitian lanjutan untuk memecahkan masalah perhitungan pagerank yang lebih kompleks

DAFTAR PUSTAKA

- Ahmad Dahlan and Benhard Sitohang, 2007. "Combining PageRank and Citation Analysis to Measure Information Credibility in Internet", Proceedings of ii WAS
- Alamsyah, Fahrijal dan Smitdev Community. 2008. *Mendulang Dolar dengan Text Link Ads*. Jakarta. Elex Media komputindo.
- Catherine Benincasa and Adena Calden, 2006. "Page Rank Algorithm",
- Dilip Kumar Sharma dan A. K. Sharma. 2010 "A Comparative Analysis of Web Page Ranking Algorithms", (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 08
- Febrian, Jack. 2008. *Menjelajah dunia dengan Google Mesin Pencarian Informasi di Internet Terbesar sedunia*. Cetakan Kcdua. Bandung. Informatika.
- Perangin-angin, Kasiman. 2006. *Pengenalan Matlab*. Yogyakarta. Andi.
- Putra, Rahmat, 2006, *Rahasia Menjadi Top 10 on Google*. Jakarta. Dian rakyat.

Lampiran 3 Program *PageRank*

```
=====
% program sederhana pagerank
=====

function pagerank(steps)

Graph = graphinit;
rand('state',0);
n = size(Graph.G,1);

help pagerank

%inisialisasi menghitung halaman
hits = zeros(1,n);
oldwhere = 1;
where = 1;
hits(where) = 1;
set(Graph.node(where),'FaceColor',[001]);

p = 0.85; %probabilitas sebuah link yang diikuti
c = sum(Graph.G); %link keluar

link = cell(1,n);
for k = 1:n
    link{k} = find(Graph.G(:,k));
end

follow_link = 0;
if(nargin < 1)
    input('tekan enter untuk memulai titik alpha:');
end

%menulis statistic pada grafik
set(Graph.nodelabel(where),'string',...
    sprintf('%s%d(%3.1f%%)',Graph.nodes{where},hits(where),...
```

```

100*hits(where)/sum(hits)));
if(nargin < 1)
    input('tekan enter untuk mengambil satu tahap:');
    steps = 1;
end
%repeat
while(1)

    %menyiapkan warna tua dan tanda panah tua
    set(Graph.nodc(whcrc),'FaceColor',[0 1 0]);
    if(follow_link)
        set(Graph.arrows(where,oldwhere),'LineWidth',2);
        set(Graph.arrows(where,oldwhere),'Color',[0 0 0]);
    end

    %determinan where menuju tahap selanjutnya
    oldwhere = where;
    if(c(where)==0|rand > p)
        %bukan link keluar, atau mengabaikan link
        follow_link = 0;
        where = floor(n*rand+1);
        set(Graph.node(where),'FaceColor',[0 0 1]);
    else
        %bergerak sepanjang link
        follow_link=1;
        where=link{where}(floor(c(where)*rand+1));
        set(Graph.node(where),'FaceColor',[0 0 1]);
        set(Graph.arrows(where,oldwhere),'LineWidth',5);
        set(Graph.arrows(where,oldwhere),'Color',[1 0 0]);
    end

    %menghitung kenaikan hits
    hits(where)=hits(where)+1;

    %menulis statistik pada grafik
    for k = 1:n
        set(Graph.nodelabel(k),'String',...
            sprintf('%s%d(%3.1f%%)',Graph.nodes{k},hits(k),...

```



```

    100*hits(k)/sum(hits));
end

drawnow

%menuju tahap selanjutnya
steps=steps - 1;
if(steps<=0)
    if(nargin>0)
        break;
    end
    step=input...
        ('number of steps to make(default1,control-C to quit):');
    if(steps==0)
        break;
    end
    if isempty(steps)
        steps=1;
    end
end
end
end

```

```

function Graph = graphinit
%GRAPHINIT membuat web kecil..
%contoh
%G=graphinit

```

```

figure(1)
clf

```

```

nodes={'alpha','beta','gamma','delta','rho','sigma'};

```

```

xy=[
    0 4
    1 3
    1 2
    2 4
    2 0
    2 0

```

```

0 0
];
x=xy(:,1);
y=xy(:,2);

%skala x dan y untuk range 0.1 sampai 0.9

x=0.8*x/2+.1;
y=0.8*y/4+.1;
xy=[xy];

xy_delta=[
    .08 .04 0
    -.03 -.02 -1
    .04 0 0
    -.05 .04 -1
    -.03 0 -1
    .03 0 0
];

xd=xy_delta(:,1);
yd=xy_delta(:,2);
tjust=xy_delta(:,3);

G=[
    0 0 0 1 0 1
    1 0 0 0 0 0
    0 1 0 0 0 0
    0 1 1 0 0 0
    0 0 1 0 0 0
    1 0 1 0 0 0];

clf

n=size(G,1);

axes('Position',[0 0 1 1],'Visible','off');
node = zeros(n,1);

```



```

nodelabel=zeros(n,1);
for k=1:n
    node(k)=node('ellipse',[x(k)-.025 y(k)-.025 .05 .05]);
    set(node(k),'LineWidth',2);
    set(node(k),'FaceColor',[0 1 0]);
    nodelabel(k) = text(x(k)+xd(k),y(k)+yd(k),nodes{k},...
        'Units','normalized','FontSize',16);
    if(tjust(k)<0)
        set(nodelabel(k),'HorizontalAlignment','right');
    end
end
end

axis off

arrows = zeros(n,n);

[ij]=find(G)
for k=1:length(i)
    %mendapatkan pusat dari dua simpul
    figx=[x(j(k)) x(i(k))];
    figy=[y(j(k)) y(i(k))];
    %[figx figy]=dsxy2figxy(gca, axx, axy);
    %mempersingkat anak panah oleh s unit pada setiap akhir
    s=0.03;
    len=sqrt(diff(figx)^2+diff(figy)^2);
    fy(1)=diff(figy)*(s/len)+figy(1);
    fy(2)=diff(figy)*(1-s/len)+figy(1);
    fx(1)=diff(figx)*(s/len)+figy(1);
    fx(2)=diff(figx)*(1-s/len)+figy(1);
    arrows(i(k),j(k))= annotation ('arrow',fx,fy);
    set(arrows(i(k),j(k)),'LineWidth',2);
    set(arrows(i(k),j(k)),'HeadLength',20);
    set(arrows(i(k),j(k)),'HeadWidth',20);
end

Graph.G = G;
Graph.nodes = nodes;
Graph.node = node;
Graph.xy = xy;

```

```
Graph.xy_delta=xy_delta;  
Graph.nodelabel=nodelabel;  
Graph.arrows=arrows;
```