ZERO – JURNAL SAINS MATEMATIKA DAN TERAPAN
Volume 1 No. 1 2017
Page : 33-40
P-ISSN: 2580-569X
E-ISSN: 2580-5754

IMPLEMENTATION OF PAGERANK ALGORITHM IN MATLAB

Rima Aprilia[1], Rina Filia Sari[2]

Department of Mathematics
UINSU Medan
rimaaprilia@gmail.com

Abstract. *Implementation of the PageRank algorithm to rank web generally only contain static and dynamic pages, with the rapid url users then needed an algorithm for calculating web rankings. In determining the ranking of a web, links incoming and outgoing links are also random surfer model is one decisive factor in determining the ranking of a web. Implementation of PageRank on MATLAB formed on a program in the m-file.*

Key Word*: Algoritma PageRank, Matlab, Implementation*

Abstrak. Implementasi algoritma PageRank untuk menentukan peringkat web umumnya hanya berisi halaman statis dan dinamis, dengan pengguna url cepat diperlukan algoritma untuk menghitung peringkat web. Dalam menentukan rangking sebuah web, link masuk dan keluar juga merupakan model surfer acak merupakan salah satu faktor penentu dalam menentukan rangking sebuah web. Implementasi PageRank pada MATLAB terbentuk pada sebuah program di m-file.

1. Pendahuluan

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 keluar) 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 di urutkan lebih dahulu dalam list pencarian Google daripada situs yang mempunyai *PageRank* 8 dan kemudian seterusnya yang lebih kecil.

Algoritma *PageRank*

PageRank adalah sebuah algoritma 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.

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 diangap 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.

Dari pendekatan yang sudah dijelaskan pada artikel konsep *PageRank*, Lawrence Page and Sergey Brin membuat algoritma *PageRank* seperti di bawah:

Algoritma awal

PR(A) = (1-d) + d ((PR(T1)/C (T1))+ . . .+ (PR(Tn) / C(Tn)))

Salah satu algoritma lain yang dipublikasikan

PR(A) = (1-d) / N + d ((PR(T1) / C(T1)) + … + (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 anda 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.

2.2. 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 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.

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". Jumlahmnya 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.

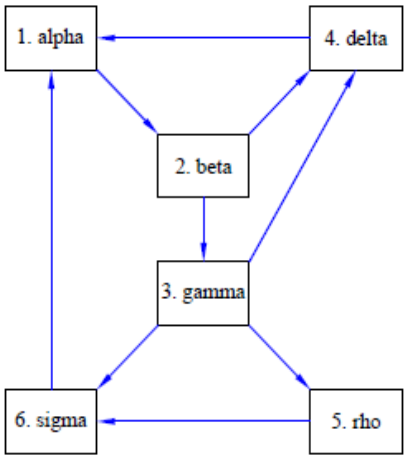
a. Untuk B : PR(B) = 1/2 + 1/3 + 1/1 hasilnya 1, 83
b. Untuk C : PR(C) = 1/2 hasilnya 0,5
c. Untuk D : PR(D) = 1/3 hasilnya 0,33

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) = ½ +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.

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 url 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 Uk 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 = sparse(i,j,1,n,n);
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 = full(sum(G))

menghitung jumlah kolom

c =1 2 3 1 1 1

Matriks diagonal I dan D dihitung seperti sebelumnya

I = speye(n,n)

D = spdiags(1./c',0,n,n)

Pernyataan

x = (I - p*G*D)\(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 δ = .15 = 6 = .0250.

A = p*G*D + 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.

4. Kesimpulan

Berdasarkan hasil penelitian dapat ditarik kesimpulan bahwa algoritma *PageRank* dapat diimplementasikan pada software MATLAB, dalam menentukan urutan tinggi suatu url dengan algoritma *PageRank* digunakan banyak faktor yaitu model peselancar acak, filter, link masuk, link keluar dan damping faktor yang telah ditetapkan yaitu 0-1. Pengujian masih bersifat sederhana belum menggunakan data yang lebih besar, masih dibutuhkan penelitan-penelitian lanjutan untuk memecahkan masalah perhitungan *PageRank* yang lebih kompleks

5. Daftar Pustaka

[1] Ahmad Dahlan and Benhard Sitohang, 2007.“*Combining PageRank and Citation Analysis to Measure Information Credibility in Internet* ”, Proceedings of ii WAS

[2] Alamsyah, Fahrijal dan Smitdev Community. 2008. *Mendulang Dolar dengan Text Link Ads.* Jakarta. Elex Media komputindo.

[3] Catherine Benincasa and Adena Calden, 2006. ” *Page Rank Algorithm*”,

[4] 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

[5] Febrian, Jack. 2008. *Menjelajah dunia dengan Google Mesin Pencarian Informasi di Internet Terbesar sedunia.* Cetakan Kedua. Bandung. Informatika.

[6] Perangin-angin, Kasiman. 2006. *Pengenalan Matlab.* Yogyakarta. Andi.

[7] Putra, Rahmat, 2006, *Rahasia Menjadi Top 10 on Google.* Jakarta. Dian rakyat.