

IMPLEMENTASI ALGORITMA PAGERANK PADA PEMROGRAMAN BERORIENTASI OBJEK (OOP) (Studi kasus Menggunakan Bahasa Pemrograman Java)

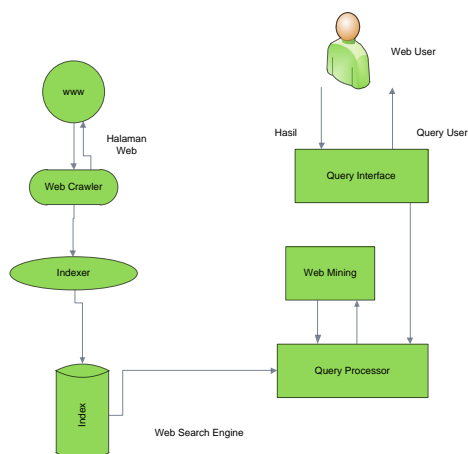
Erwin Renaldhy *)
Muhammad Rizal H **)

Abstract : *Abstract-There are many google pagerank checker available on the internet for free, most of the methods, techniques and calculations used are the same values that produce a close approximation to the actual value of the ranking / rating website. This paper will design the application architecture Pagerank calculator gui (graphic user interface) based object / visual applications available compared to the internet for free. Built using Java programming language.*

Keywords: *Oop, pagerank algorithm, PageRank calculator java*

PENDAHULUAN

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, Gambar berikut menunjukkan bagaimana unjuk kerja dari search engine, dimana menampilkan flow graph untuk pencarian query oleh pengguna web [1,2,4].



Gambar 1. Kerja Mesin Pencari

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 1-10.

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.

Google page rank checker begitu banyak tersedia di internet yang bisa digunakan dengan instant oleh pengguna website hanya dengan memasukkan alamat *Url (Uniform Resource Locator)*, aplikasi Pagerank calculator rata-rata memiliki kesamaan fungsi, adapun perbedaan terdiri dari :

1. Hanya menampilkan PR skala 1-10
2. Menampilkan PR berdasarkan iterasi
3. Memasukkan lebih dari 1 link url untuk membandingkan PR antara masing-masing link [3,5,7].

Tujuan dari makalah ini adalah bagaimana menerapkan algoritma pagerank kedalam bahasa pemrograman objek java, melihat arsitektur aplikasi dan implementasi kalkulasi pagerank kedalam bahasa java [9].

TINJAUAN PUSTAKA

Overviews

Pagerank diciptakan Sergey Brin dan Larry Page pada Konferensi Web Dunia Internasional Ketujuh (www-Conference) pada bulan April 1998. Pagerank adalah sebuah algoritma pencarian peringkat menggunakan hyperlink pada Web. Pagerank disebut juga sebagai salah satu fitur utama mesin pencari google [6].

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 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.

Pagerank menghasilkan peringkat statis halaman web dalam arti bahwa nilai

Pagerank dihitung untuk setiap halaman dan tidak tergantung pada permintaan pencarian. Algoritma bergantung pada keterbukaan hal. Web dengan kekuatan pada struktur link yg dijadikan indikator kualitas.

B. Objek Oriented Programming (Pemrograman Berorientasi Objek)

Merupakan paradigma pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus dalam kelas-kelas atau objek-objek. Bandingkan dengan logika pemrograman terstruktur. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya, Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik piranti lunak skala besar. Lebih jauh lagi, pendukung OOP mengklaim bahwa OOP lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan OOP lebih mudah dikembangkan dan dirawat.

Konsep dasar pemrograman berorientasi Objek menekankan konsep berikut:

- *Kelas* : kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebagai contoh 'class of dog' adalah suatu unit yang terdiri atas definisi-definisi data dan fungsi-fungsi yang menunjuk pada berbagai macam perilaku/turunan dari anjing. Sebuah class adalah dasar dari modularitas dan struktur dalam pemrograman berorientasi object. Sebuah class secara tipikal sebaiknya dapat dikenali oleh seorang non-programmer sekalipun terkait dengan domain permasalahan yang ada, dan kode yang terdapat dalam sebuah class sebaiknya (relatif) bersifat mandiri dan independen (sebagaimana kode tersebut digunakan jika tidak menggunakan OOP). Dengan modularitas, struktur dari

sebuah program akan terkait dengan aspek-aspek dalam masalah yang akan diselesaikan melalui program tersebut. Cara seperti ini akan menyederhanakan pemetaan dari masalah ke sebuah program ataupun sebaliknya.

- *Objek* : Membungkus data dan fungsi bersama menjadi suatu unit dalam sebuah program komputer, objek merupakan dasar dari modularitas dan struktur dalam sebuah program komputer berorientasi objek.
- *Abstraksi* : Kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti. Setiap objek dalam sistem melayani sebagai model dari "pelaku" abstrak yang dapat melakukan kerja, laporan dan perubahan keadaannya, dan berkomunikasi dengan objek lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan. Proses, fungsi atau metode dapat juga dibuat abstrak, dan beberapa teknik digunakan untuk mengembangkan sebuah pengabstrakan.
- *Enkapsulasi* : Memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam dari sebuah objek dengan cara yang tidak layak; hanya metode dalam objek tersebut yang diberi izin untuk mengakses keadaannya. Setiap objek mengakses interface yang menyebutkan bagaimana objek lainnya dapat berinteraksi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.
- *Polimorfisme* : Melalui pengiriman pesan. Tidak bergantung kepada pemanggilan subrutin, bahasa orientasi objek dapat mengirim pesan; metode tertentu yang berhubungan dengan sebuah pengiriman pesan tergantung kepada objek tertentu di mana pesa tersebut dikirim.

Contohnya, bila sebuah burung menerima pesan "gerak cepat", dia akan menggerakkan sayapnya dan terbang. Bila seekor singa menerima pesan yang sama, dia akan menggerakkan kakinya dan berlari. Keduanya menjawab sebuah pesan yang sama, namun yang sesuai dengan kemampuan hewan tersebut. Ini disebut polimorfisme karena sebuah variabel tunggal dalam program dapat memegang berbagai jenis objek yang berbeda selagi program berjalan, dan teks program yang sama dapat memanggil beberapa metode yang berbeda di saat yang berbeda dalam pemanggilan yang sama. Hal ini berlawanan dengan bahasa fungsional yang mencapai polimorfisme melalui penggunaan fungsi kelas-pertama.

Dengan menggunakan OOP maka dalam melakukan pemecahan suatu masalah kita tidak melihat bagaimana cara menyelesaikan suatu masalah tersebut (terstruktur) tetapi objek-objek apa yang dapat melakukan pemecahan masalah tersebut. Sebagai contoh anggap kita memiliki sebuah departemen yang memiliki manager, sekretaris, petugas administrasi data dan lainnya. Misal manager tersebut ingin memperoleh data dari bag administrasi maka manager tersebut tidak harus mengambilnya langsung tetapi dapat menyuruh petugas bag administrasi untuk mengambilnya. Pada kasus tersebut seorang manager tidak harus mengetahui bagaimana cara mengambil data tersebut tetapi manager bisa mendapatkan data tersebut melalui objek petugas administrasi. Jadi untuk menyelesaikan suatu masalah dengan kolaborasi antar objek-objek yang ada karena setiap objek memiliki deskripsi tugasnya sendiri [5,7,9].

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 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 .*

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

$$PR(A) = (1-d) + d \left(\left(\frac{PR(T1)}{C(T1)} \right) + \dots + \left(\frac{PR(Tn)}{C(Tn)} \right) \right)$$

Salah satu algoritma lain yang dipublikasikan

$$PR(A) = (1-d) / N + d \left(\left(\frac{PR(T1)}{C(T1)} \right) + \dots + \left(\frac{PR(Tn)}{C(Tn)} \right) \right)$$

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.

D. Kalkulasi Pagerank

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

Tabel 1. 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 halaman Anda memiliki 4 halaman yang menghubungkan ke sebagai berikut:

Tabel 2. Contoh estimasi pr

Estimate of Toolbar PageRank vs. True PageRank Based on Log Base 5 Scale	
Toolbar PageRank	True PageRank
0	.15-5
1	5-25
2	25-125
3	125-625
4	625-3,125
5	3,125-15,625
6	15,625-78,125
7	78,125-390,625
8	390,625-1,953,125
9	1,953,125-9,765,625
10	9,765,625-48,828,125

Page 1: Pagerank Benar 110, Links Outgoing Total 5

Page 2: Benar Pagerank 1000, Links Outgoing Total 45

Page 3: Pagerank Benar 2000, Links Outgoing Total 100

Page 4: Pagerank Benar 100, Outgoing Links Total 1

Pagerank dari halaman = $.15 + .85 (PR(T1) / C(T1) + PR(T2) / C(T2) + \dots + PR(Tn) / C(Tn))$

Pagerank dari halaman = $.15 + .85 (110/5 + 1000/45 + 2000/100 + 100/1)$.

Pagerank dari halaman = $.15 + .85 (22 + 22,22 + 20 + 100)$

Pagerank dari halaman = 121,37

Toolbar Perkiraan Peringkat = 2

Sekarang apa yang terjadi jika kita mengganti Page 3 dengan halaman lain dari Pagerank 2000 tetapi hanya 3 link keluar:

Pagerank dari halaman = $.15 + .85 (110/5 + 1000/45 + 2000/3 + 100/1)$

Pagerank dari halaman = $.15 + .85 (22 + 22,22 + 666,66 + 100)$

Pagerank dari halaman = 689,39

Toolbar estimasi Peringkat = 4

Dari contoh perhitungan diatas dapat disimpulkan bahwa outlink sangat mempengaruhi pertumbuhan *pr* website.

Ilustrasi :

Untuk merumuskan ide-ide tersebut, kita memperlakukan Web sebagai graph, $G = (V, E)$, di mana V adalah himpunan simpul atau node, yaitu, himpunan semua halaman, dan E adalah himpunan tepi (edges) diarahkan dalam graph, i, e, \dots , hyperlink. jumlah halaman di Web menjadi

n (yaitu, $i.e., n = |V|$).

Nilai Pagerank dari halaman i (dilambangkan dengan $P(i)$) didefinisikan oleh:

$$P(i) = \sum_{(j,i) \in E} \frac{P(j)}{o_j}$$

Dimana o_j adalah jumlah out-link dari halaman j . Secara matematis, memiliki sistem persamaan linear n dengan variabel n tidak diketahui. Kita dapat menggunakan matriks untuk mewakili semua persamaan. P menjadi n -dimensi vektor kolom nilai Pagerank : $P = (P(1), P(2), \dots, P(n))^T$

Misalkan A adalah matriks adjacency dari graph kami dengan :

$$A_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E \\ 0 & \end{cases}$$

Kita bisa menulis sistem persamaan n dengan $P = A^T P$

Eigenvalue terbesar dan Pagerank vektor \mathbf{P} adalah **eigenvektor** utama. Sebuah teknik matematika yang terkenal disebut kekuatan iterasi dapat digunakan untuk menemukan \mathbf{P} . Namun, masalahnya adalah persamaan itu tidak cukup karena graph web tidak memenuhi persyaratan. Bahkan juga dapat diturunkan berdasarkan rantai Markov. Kemudian beberapa hasil teoritis dari rantai Markov dapat diterapkan.

Setelah menambah graph Web untuk memenuhi kondisi, maka menggunakan persamaan Pagerankat.

$$\mathbf{P} = (\mathbf{1} - d)\mathbf{e} + d \mathbf{A}^T \mathbf{P}$$

```

PageRank-Iterate(G)
  P0 ← e/n
  k ← 1
  repeat
    Pk ← (1 - d)e + dATPk-1;
    k ← k + 1;
  until ||Pk - Pk-1||1 < ε
  return Pk

```

Rumus setiap Halaman i :

$$\mathbf{P}(i) = (\mathbf{1} - d) + d \sum_{j=1}^n \mathbf{A}_{ji} \mathbf{P}(j)$$

Sehingga diperoleh asumsi, antara lain:

- **PR** dari setiap halaman tergantung dari halaman PR yang merujuk.
- Tapi kita tidak akan tahu apa PR halaman tersebut telah sampai untuk dihitung.
- Yang kita lakukan adalah menebak atau melakukan perkiraan.

PAGERANK CALCULATOR

Bagi pemilik website atau blog tidak perlu khawatir apabila belum mengetahui PR website/blognya ini dikarenakan saat ini sudah tersedia aplikasi online checker PR yang bisa digunakan secara instant hanya dengan memasukkan link lalu akan dilakukan perhitungan PR.

Berikut beberapa penyedia aplikasi pr checker gratis, antara lain :

1. www.prchecker.info

2. www.checkpagerank.net
3. www.cekpr.com
4. www.cekpagerank.info
5. www.checkwebrank.info
6. www.getrank.org
7. www.cpagerank.com
8. www.multipagerank.com
9. www.pagerankim.com
10. www.pagerank-tools.com

Cara menggunakan aplikasi pagerank calculator hampir sama, yaitu dengan memasukkan nama domain, mengisi permintaan security code/chapcha lalu akan menampilkan skala 1-10 PR, peringkat/rangking website.

A. Bahasa Pemrograman Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems. Bahasa ini banyak mengadopsi sintaks yang terdapat pada C dan C++ namun dengan sintak model objek yang lebih sederhana serta dukungan rutin-rutin yang minimal. Aplikasi-aplikasi berbasis java umumnya dikompilasi ke dalam p-code (*bytecode*) dan dapat dijalankan pada berbagai Mesin Virtual Java (*JVM*). Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (*general purpose*), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda, saat ini java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis web.

```

1 // Import packages
import com.remedy.arsys.api

2 {
// Create UserSession object
ARServerUser context = new ARServerUser ("Demo, "<password>",
"<language>", "<server name>");

3 try {
// create an instance of <server object>, for example, Entry,
// using its factory class.
<server object>-Factory x = <server object>-Factory.getFactory
<server object>- x = (<server object>) x.newInstance();

4 // Set the context for the operation
x.setContext(context);

5 // get/set calls as needed
x.get
x.get
x.get

// One of the AR System persistence method
// calls (for example, load, create and so on)
x.load

6 // Release the instance of the server or entry object
// using the factory class and final cleanup.
<server object>-Factory.releaseinstance (x);
}
}
// Create another ARServerUser object...
    
```

Gambar 3. Struktur Listing Code Java

Package contoh

Class Pertama	Class Kedua
Methode aaa	Methode aaa
Instruksi Variabel Lokal	Instruksi Variabel Lokal
Methode zzzz	Methode zzzz
Instruksi Variabel Lokal	Instruksi Variabel Lokal
Class Instance Variabel	Class Instance Variabel

Gambar 2. Struktur Program Java

B. Aplikasi Pagerank Calculator

Pada kasus ini beberapa struktur file program aplikasi pagerank calculator ini adalah sebagai berikut.

Package page_rank yang terdiri dari class :

1. Main.java
2. Link.java
3. PageRankCal.java
4. App.Form.java
5. ResultForm.java

- Main.java berisi deklarasi PR sebagai variabel publik

```

PageRankCalc pr=new
PageRankCalc();
pr.addLink("a", "b");
pr.addLink("c", "b");
pr.addLink("c", "a");
pr.addLink("d", "a");
pr.addLink("c", "d");
pr.addLink("e", "c");
pr.addLink("f", "a");
pr.addLink("g", "a");
pr.addLink("g", "a");
pr.addLink("h", "a");
pr.addLink("b", "a");
pr.addLink("a", "h");
pr.addLink("a", "z");
pr.addLink("b", "z");
pr.addLink("c", "z");
pr.addLink("d", "z");
pr.addLink("e", "z");
pr.calculate();
    
```

```

System.out.println(pr.getPa
geRank("h"));
    
```

- Link.java adalah class untuk menampung variabel url yang akan dimasukkan untuk kemudian dilakukan perhitungan PR berdasarkan nilai masuk string. Adapun proses yang dilakukan di class tersebut, antara lain :

```

Public Class Link {
Public String From, To;
    
```

```

Public String getForm() {...}
Public String getTo() {...}
Public set void setForm(String From) {...}
Public SetTo(String To) {...}
    
```

- PageRankCal.java adalah class utama yang melakukan proses kalkulasi PR berdasar input nilai pada form masukan (appForm.java) untuk kemudian outputnya ditampilkan (resultForm.java), struktur class pageRankCalc.java adalah:

1. Create Vector public
2. Menambahkan link / addLink
3. Hapus link /removeLink

4. `getUniqueElements() = Vector`
5. For Iteration vektor
6. Sum pagerank

Berikut bagian utama dari class PageRankCalc.java

```
package page_rank;

import java.util.*;

public class PageRankCalc {

    public Vector links;
    public Vector uniqueElements;
    public double[] pageRanks;

    public PageRankCalc() {}

    public void addLink(String from, String to) {}

    public void removeLink(String from, String to) {}

    public void getUniqueElements() {}

    public int howManyLinksSends(String element) {}

    public int howManyLinksReceives(String element) {}

    public boolean getLinkFrom (String element, String from) {}
}
```

Gambar 4. Class PageRankCalc.java

HASIL SIMULASI

Pada bagian ini akan dibandingkan aplikasi pagerank calculator online berbasis website dengan program pagerank calculator yang dibuat menggunakan bahasa pemrograman java *gui(graphic user interface)*.

C. Pagerank calculator Online

Penulis akan menggunakan pagerank checker online di <http://www.markhorrell.com/seo/pagerank.asp> dimana akan pr akan dikalkulasi berdasarkan nilai iterasi minimal 2 iterasi

[Note: The Web tools section of this website contains legacy content which is no longer being supported. Although the tools will remain online for the foreseeable future, they are not being updated and may not be current.]

The Google PageRank Calculator

[The PageRank Algorithm](#) - Click here for some thoughts on PageRank, and operating instructions for the PageRank Calculator.

[About the PageRank Calculator](#) - A brief explanation and some further resources.

Step One: Please input the number of interlinking pages to be analysed (Max.50)

[Click here to start again.](#)

Gambar 5. Google PR Calculator

Step berikutnya anda diminta memasukkan nama page atau nama domain contohnya www.stimednp.ac.id akan muncul dumping factor = 0.85 dengan iterasi 20, klik tombol submit tampil jumlah iterasi beserta hasil kalkulasi PR.

[Note: The Web tools section of this website contains legacy content which is no longer being supported. Although the tools will remain online for the foreseeable future, they are not being updated and may not be current.]

The Google PageRank Calculator

[The PageRank Algorithm](#) - Click here for some thoughts on PageRank, and operating instructions for the PageRank Calculator.

[About the PageRank Calculator](#) - A brief explanation and some further resources.

Calculating PageRank....

Iteration	www.stimednp.ac.id	www.stimednp.ac.id	Total
0	1	1	2
1	0.15	0.15	0.3
2	0.15	0.15	0.3
3	0.15	0.15	0.3
4	0.15	0.15	0.3
5	0.15	0.15	0.3
6	0.15	0.15	0.3

Gambar 6. Hasil kalkulasi PR

www.stimednp.ac.id

D. PR Aplikasi Java

Perlu diingat bahwa sebelum menggunakan aplikasi tersebut disyaratkan untuk menggunakan internet karena kita akan mengkalkulasi PR dari url yang dimasukkan pada kolom site.

Langkah-langkah pengujian yang dibutuhkan antara lain :

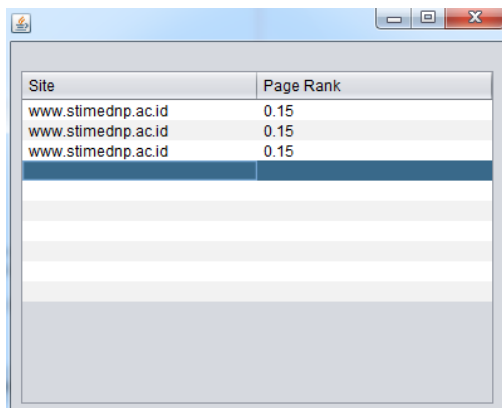
1. Masukkan alamat url site atau nama domain yang akan dikalkulasi

Site	Links To
www.stimednp.ac.id	2

Calculate Clean

Gambar 7. Input url/domain

2. Kolom links to untuk nilai iterasi
3. Klik calculate maka proses perhitungan dilaksanakan (pastikan untuk terhubung ke internet)
4. Program akan running beberapa saat untuk mencari data.
5. Bila sukses akan tampil hasil kalkulasi page rank



Site	Page Rank
www.stimednp.ac.id	0.15
www.stimednp.ac.id	0.15
www.stimednp.ac.id	0.15

Gambar 8. Hasil kalkulasi PR

KESIMPULAN

Makalah ini mencakup dasar-dasar perhitungan algoritma pagerank, perbandingan pagerank calculator hanya bersifat untuk eksplorasi algoritma pagerank untuk model *gui*, 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

- Dilip Kumar Sharma dan A. K. Sharma, "A Comparative Analysis of Web Page Ranking Algorithms", (IJCS) International Journal on Computer Science and Engineering Vol. 02, No. 08, 2010.
- Ashutosh Kumar Singh, Ravi Kumar P, "A Comparative Study of Page Ranking Algorithms for Information Retrieval", International Journal of Electrical

and Computer Engineering 4:7 2009

Sung Jin Kim and Sang Ho Lee, "An Improved Computation of the PageRank Algorithm", School of Computing, Soongsil University, Korea.

Ahmad Dahlan and Benhard Sitohang, "Combining PageRank and Citation Analysis to Measure Information Credibility in Internet", Proceedings of ii WAS 2007.

Amy N. Langvilletand Carl D. Meyer, "Deeper Inside PageRank", October 20, 2004.

Catherine Benincasa and Adena Calden, "Page Rank Algorithm", May 12, 2006.

Joni Pajarinen, Joni.Pajarinen@tkk.fi, "The PageRank/HITS algorithms", Maret 19, 2008.

Laxmi Choudhary and Bhawani Shankar Burdak, "Role of Ranking Algorithms for Information Retrieval", International Journal of Artificial Intelligence & Applications (IJAA), Vol.3, No.4, July 2012.

Ibnu Rahman Chalid, "Aplikasi Audio Steganografi Untuk melindungi Data Menggunakan Bahasa Pemograman Java", 8 Agustus 2009.

*) **Penulis adalah Fakultas Teknik Elektro Konsentrasi Teknik Informatika Universitas Hasanuddin**

***) **Penulis adalah Fakultas Teknik Elektro Konsentrasi Teknik Informatika Unirversitas Hasanuddin, Makassar**