Aplikasi Teori Graf untuk Mengurutkan Web Pages Menggunakan Markov Chain dalam Algoritma PageRank

Raden Rafly Hanggaraksa Budiarto- 13522014[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13522014@std.stei.itb.ac.id

***Abstrak*—PageRank merupakan salah satu penerapan aplikasi Teori Graf yang membuat penciptanya mendirikan perusahaan teknologi bernama Google. Google sendiri membangun teknologi search engine mereka di sekitar algoritma PageRank tersebut. PageRank ini melakukan perhitungan dengan sudut pandang yang berbeda. Alih-alih melihat frekuensi kata kunci, meta tags, dan popularitas link, PageRank memberi peringkat berdasarkan jumlah dan kualitas dari link yang menunjuk ke situs tersebut. Algoritma PageRank ini memanfaatkan Teori Graf dengan Markov Chain sebagai landasannya.**

***Keywords*—PageRank, Markov Chain, Stationary Distribution, Situs**

I. PENDAHULUAN

Pada awal abad ke 20, teknologi informasi berkembang secara eksponensial dalam jangka waktu yang tidak lama. Penyebaran dan pengaksesan informasi sudah tidak menjadi tantangan yang susah. Setiap orang dapat membuat konten sendiri di dalam situs web yang ia buat.

Jumlah dari informasi yang disimpan dalam web bertumbuh secara cepat. Pada awalnya, website yang tercatat pada tahun 2008 hanyalah sekitar 70.000.000, tetapi pada awal tahun 2023 sudah tercatat sekitar 1.000.000.000 situs web [1].

Di antara semua informasi tersebut, pengguna belum tentu memerlukan semuanya. Dibutuhkan suatu sistem untuk menyaring dan mengurutkan informasi berdasarkan masukan oleh pengguna. Sistem ini bernama search engine. Search engine yang kita ketahui pada umumnya adalah Google, Bing, Yahoo, dan lainnya.

A. Search Engine Sebelum Google

Google bukanlah salah satu search engine dan bukan pula yang pertama. Search engine pertama yang dapat diakses oleh Masyarakat luas adalah "Archie". Akan tetapi, Archie bukanlah search engine yang kita ketahui pada umumnya. Melainkan, Archie merupakan suatu alat untuk mengindex dan menerima file dari situs FTP (File Transfer Protocol). Archie menyediakan fondasi umum bagi search engine modern yang kita ketahui sekarang.

Cara kerja dari search engine tersebut terbagi menjadi berbagai tahap. Tahap pertama adalah Web Crawling. Search engine menggunakan Web Crawlers untuk menavigasi seluruh situs yang ada dan menerima informasi dari situs tersebut. Tahap selanjutnya adalah indexing. Data yang telah terkumpul akan di index dan disusun sedemikian rupa agar search engine dapat mengaksesnya dengan mudah saat pengguna memasukan query. Tahap terakhir adalah algoritma ranking. Algoritma ini akan menentukan posisi situs yang didapat guna menentukan relevansi dari query yang diterima. Beberapa faktor seperti frekuensi kata kunci, meta tags, dan popularitas link diperhatikan dalam algoritma ini.

B. Algoritma PageRank Google

Salah satu faktor yang membedakan Google dari search engine yang lain adalah algoritma rankingnya. Algoritma ini dikembangkan oleh Larry Page dan Sergei Brin, yang merupakan pendiri Google, dengan nama "PageRank". PageRank ini melakukan perhitungan dengan sudut pandang yang berbeda. Alih-alih melihat frekuensi kata kunci, meta tags, dan popularitas link, PageRank memberi peringkat berdasarkan jumlah dan kualitas dari link yang menunjuk ke situs tersebut. Hal ini membantu Google memprioritaskan laman yang lebih relevan dan ketat. Algoritma PageRank ini memanfaatkan Teori Graf dengan Markov Chain sebagai landasannya.

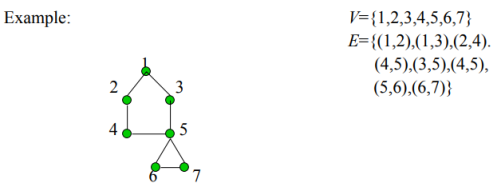
II. LANDASAN TEORI

A. Teori Graf

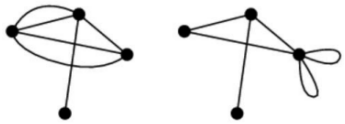
Teori Graf merupakan cabang dari Matematika diskrit yang mempelajari relasi antar objek. Objek, yang biasa diberi nama vertices, terhubung oleh edges atau arc. Sebuah graf G memiliki Kumpulan set vertices V dan edge E. Setiap edge merupakan pasangan dari (a,b) dengan a dan b adalah vertices.

Berdasarkan ada tidaknya gelang atau sisi ganda pada suatu graf, graf tergolongkan menjadi **Graf Sederhana** dan **Graf tidak sederhana**.

1. **Graf Sederhana** adalah graf yang tidak memiliki gelang atau sisi ganda.
2. **Graf tidak sederhana** adalah graf yang mengandung gelang atau sisi ganda.



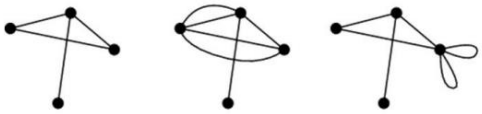
Gambar 1 Graf Sederhana [2]



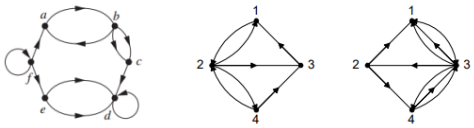
Gambar 2 Graf tidak sederhana [2]

Tidak seperti **Gambar 1** dan **Gambar 2**, sisi pada graf dapat juga memiliki arah. Graf dapat dibagikan kembali berdasarkan arah dari sisi/edgenya.

1. Graf yang edgenya tidak memiliki orientasi arah disebut **Graf tak-berarah**.
2. Graf yang edgenya memiliki orientasi arah disebut **Graf berarah**.



Gambar 3 Graf tak-berarah [2]



Gambar 4 Graf berarah [2]

B. Representasi Graf

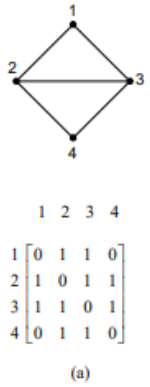
Graf dapat direpresentasikan dalam bentuk matriks. Ada dua jenis matriks yang digunakan untuk merepresentasikan suatu graf. Matriks tersebut antara lain Matriks Ketetanggaan / Adjacency Matrix dan Matriks Bersisian / Incidency Matrix.

1. Matriks Ketetanggaan / Adjacency Matrix
   Adjacency Matrix adalah sebuah matriks kotak yang digunakan untuk merepresentasikan graf berdasarkan ketetanggannya. Baris dan kolom dari matriks ini merupakan vertices dari graph yang ingin direpresentasikan. Nilai elemen dari matriks ini mengindikasikan apakah pasangan dari vertices tersebut bertetangaan atau tidak. Adjacency Matrix A dari sebuah graf G dengan n vertices adalah sebuah matriks nxn dengan:

   $A[i][j] = 1$, jika terdapat edge diantar vertices i dan j
   $A[i][j] = 0$, jika tidak terdapat edge diantar vertices i dan j

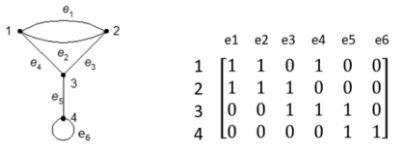
   Untuk graf tidak berarah, adjacency matrix-nya simetris, sehingga

   $A[i][j] = A[j][i]$ untuk semua i dan j.



Gambar 5 Graf dengan representasi Adjacency Matrix [2]

2. Matriks Bersisian / Incidency Matrix
   Incidency Matrix adalah sebuah matriks yang digunakan untuk merepresentasikan graf berdasarkan hubungan edge dengan verticesnya. Incidency Matrix M dari sebuah graf G dengan n vertices dan m edges adalah sebuah matriks n x m dengan,
   $M[i][j] = 1$ jika vertex i bersisian dengan edge j
   $M[i][j] = 0$ jika vertex I tidak bersisian dengan edge j



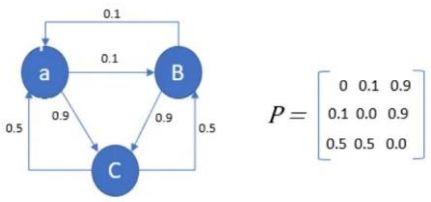
Gambar 6 Graf dengan representasi Incidency Matrix [2]

C. Markov Chain

Markov Chain adalah model Matematika yang mendeskripsikan sebuah sistem yang melakukan transisi antara state berdasarkan suatu aturan probabilitas. Karakteristik dari Markov Chain adalah Properti Markov. Properti Markov menyatakan bahwa peluang transisi ke state lain bergantung pada state terkini dan waktu/langkah yang telah ditempuh. Dengan kata lain, masa depan sifat dari sistem ini bebas dari masa lalunya.

Model matematis dari Properti Markov adalah sebagai berikut:

$$P[Xl+1 = s \mid Xl = sl, Xl-1 = sl-1, \cdots, X \circ = s \circ] = P[Xl+1 = s \mid Xl = sl] \quad (1)$$

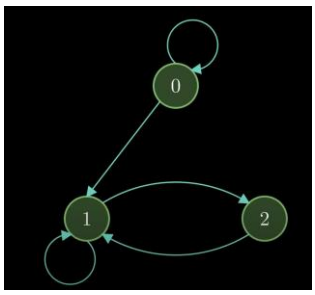


Gambar 7 Markov Chain dengan Representasi Transition Probabilites dalam Adjacency Matrix [3]

Beberapa terminology dari Markov Chain adalah sebagai berikut:

1. States
   Kumpulan dari seluruh state terbatas yang bisa digunakan oleh sistem. State biasanya diwakilkan sebagai vertices dalam graf. Pada **Gambar 7**, Statesnya adalah {a,b,c}
2. Transition Probabilites
   Peluang transisi dari suatu state ke state lainnya bagi suatu pasang state. Transition Probabilites ini biasanya direpresentasikan dengan Adjancency Matrix dengan nilai elemennya berupa peluang dari sepasang state. Pada **Gambar 7**, Transition Probabilites merupakan nilai pada edge yang ada dan sudah direprestasikan dalam matriks P.
3. Initial State Probability Vector (π_0)
   Sebuah vector yang menyatakan peluang ke dalam setiap state pada awal proses. Nilai dari peluang awal bebas dimasukkan. [4]

D. Reducible Chain dan Irreducible Chain



Gambar 8 Markov Chain tanpa Transition Probablity [5]

Transient State adalah sebuah state yang memiliki kemungkinan untuk keluar dari state tersebut dan tidak pernah kembali lagi ke state tersebut. Pada **Gambar 8**, Transient State dari Markov Chain tersebut adalah state 0. Hal in dibuktikan jika state 0 melakukan transisi ke transisi state 1, maka "pejalan" tidak mungkin akan kembali ke state 0.

Recurrent State adalah sebuah state yang memiliki kemungkinan untuk kembali lagi ke state tersebut setelah melalui beberapa kali proses. Pada **Gambar 8**, Recurrent State dari Markov Chain tersebut adalah state 1 dan state 2.

Markov Chain yang memiliki Transient State disebut Reducible Chain sementara yang tidak memiliki Transient State disebut Irreducible Chain.

E. State Probability

Pada waktu ke-n, persebaran peluang setiap state akan berubah. Perubahan ini dapat kita modelkan dalam persamaan:

$$\boxed{\vec{\pi}_{n+1} = \vec{\pi}_n P} \qquad (2)$$

F. Stationary Distribution

Stationary Distribution pada Markov Chain adalah sebuah distribusi peluang yang tidak akan berubah setelah setiap transisi / proses pada graf. Dengan kata lain, jika suat Markov Chain memulai Initial State Probabilitynya menggunakan Stationary Distribution, maka nilai dari peluangnya tidak akan berubah seiring berjalannya waktu. Stationary Distribution dapat dimodelkan dengan persamaan:

$$\boxed{\vec{\pi} = \vec{\pi}A} \qquad (3)$$

Dalam mencari Stationary Distribution, dapat digunakan nilai eigen dan vector eigen:

$$\vec{\pi} = A^T\vec{\pi} \qquad (4)$$

$$\lambda\vec{\pi} = A^T\vec{\pi} \qquad (5)$$

Didapatkan nilai $\vec{\pi}$ bila kita mengasumsikan $\lambda = 1$.

Tidak semua jenis Markov Chain memiliki nilai Staionary Distribution. Menurut Teorema Ergodic, hanya Markov Chain yang irreducible dan aperiodic.

G. Random Surfer Model

Random Surfer Model merupakan suatu model graf yang menjelaskan kemungkinan pengguna acak untuk mengunjungi suatu situs web. Model ini merupakan penerapan teori Markov Chain yang akan digunakan kedalam algoritma PageRank Google.

III. METODE

A. Deskripsi Umum dari Kalkulasi Menggunakan PageRank

PageRank, pada dasarnya, mengguna sistem 'vote' oleh situs yang lain dalam menentukan tingkat kepentingan suatu situs. Tautan menuju situs diasumsikan menjadi suatu "suara" terhadap situs tersebut. Apabila tidak terdapat tautan apa pun, page tersebut diasumsikan tidak memberikan suara apa – apa terhadap situs yang lain.

Mengutip dari karya ilmiah orisinal buatan Google:

> *We assume page A has pages TI...Tn which point to it (i.e., are citations). The parameter d is a damping factor which can be set between 0 and 1. We usually set d to 0.85. There are more details about d in the next section. Also C(A) is defined as the number of links going out of page A.*

Suatu nilai PageRank dari sebuah situs A diberikan sebagai berikut:

$$\boxed{PR(A) = (1-d) + d.(\frac{PR(T1)}{C(T1)} + \cdots + \frac{PR(Tn)}{C(Tn)})} \qquad (6)$$

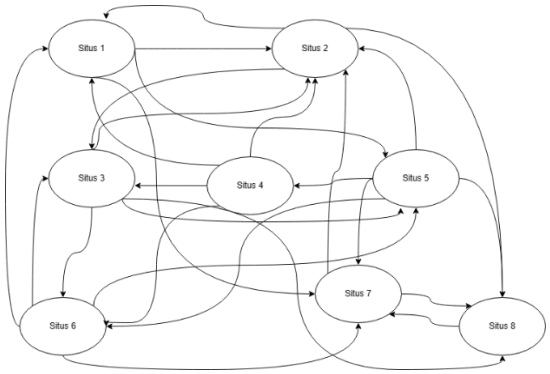
> *Note that the PageRanks form a probability distribution over web pages, so the sum of all web pages PageRanks will be one.*

> *PageRank or PR(A) can be calculated using a simple iterative algorithm, and corresponds to the principal eigenvector of the normalized link matrix of the web. [6]*

Pada **Persamaan 6** tersebut:

1. PR(Tn)
   Setiap situs memiliki nilai PageRank-nya masing – masing. PR(T1) mengartikan nilai PageRank dari situs pertama hingga PR(Tn) mengartikan nilai PageRank situs terakhir.
2. C(Tn)
   Setiap situs memberikan "suara" secara merata kepada situs yang ia berikan. C(T1) mengartikan derajat keluar vertices / state dari situs 1.
3. PR(Tn)/C(Tn)
   Bagian ini mengartikan jika situs kita dirujuk oleh suatu situs n, situs kita akan mendapatkan suara sebesar PR(Tn) / C(Tn).
4. d(….
   Semua pecahan suara akan dijumlahkan. Namun, untuk mencegah suatu situs memiliki pengaruh yang sangat besar, jumlah suara akan diredam dengan cara mengalikannya dengan d=0,85.
5. (1-d)
   Bagian ini bertujuan agar total PageRank seluruh situs bernilai satu. Nilai ini juga berguna untuk menangani kasus apabila Teorema Ergodic tidak terpenuhi. [4]

B. Pembentukan Graf Berbobot



Gambar 9 Markov Chain yang Telah Terbentuk oleh Web Crawler

Untuk membuat hubungan antara page dengan graf seperti di atas, dibutuhkan suatu web crawler yang dapat secara acak menelusuri suatu link pada page tertentu. Web crawler ini diluar pembahasan dari karya ilmiah ini sehingga kita asumsikan web crawler sudah melakukan kerjanya dan kita telah disajikan data tersebut. Pengguna dapat secara bebas memilih page apa pun sebagai state awalnya.

C. Pengisian Nilai PageRank

Berdasarkan **Gambar 9**, kita mendapatkan suatu Graf dengan representasi Adjacency Matrix sebagai berikut

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Gambar 10 Representasi Graf pada Gambar 9 dengan Matrix

Initial state probability-nya diisi dengan angka bebas. Pada kesempatan ini, penulis mengujikan dengan vector:

$$\overrightarrow{\pi_0} = [0.5\ ,0.1\ ,0.25\ ,1\ ,0.006\ ,2\ ,0.1\ ,1.9] \qquad (7)$$

D. Program Sederhana untuk Mengaplikasikan Model PageRank

Dalam pemrosesan model PageRank, penulis membuat program sederhana berbahasa python. Program tersebut menerima masukan berupa banyaknya page, hubungan antar page, Initial state probability, dan threshold. Program yang dibuat penulis bersifat brute force sehingga dibutuhkan suatu Batasan / threshold untuk menentukan keluaran.

Program ini akan melakukan iterasi terus menerus untuk memperbaru nilai PageRank setiap page sehingga tercapai suatu Stationary Distribution. Stationary Distribution dianggap tercapai apabila mutlak selisih peluang pada saat n+1 dan pada saat n lebih kecil daripada nilai threshold.

```
================ KEADAAN STATE ================
Adjacency Matriks:
0 1 0 0 0 0 0 0
1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 1 1 1 1 1 0 1
1 1 1 1 1 1 1 0

Page Rank:
Situs1: 1.1214285714285714
Situs2: 1.366309523809524
Situs3: 0.4130952380952381
Situs4: 0.4130952380952381
Situs5: 0.4130952380952381
Situs6: 0.4130952380952381
Situs7: 0.27142857142857146
Situs8: 0.18845238095238098

Average PR: 0.575
Threshold: 0.0001

Iterasi ke-1
Threshold is not satisfied
```

Gambar 11 Tampilan Output Sederhana Program

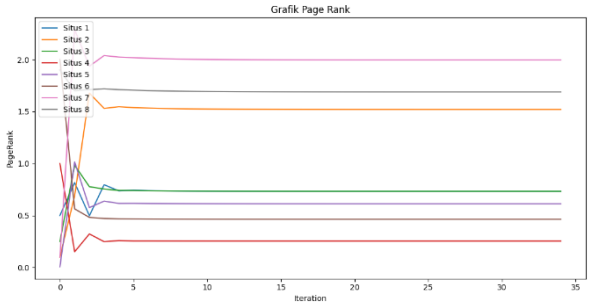
E. Pemodelan Data Menggunakan Program Python

Berdasarkan program yang dibuat penulis [7], diperoleh:

Tabel 1. Daftar nilai PageRank setiap situs setiap proses / iterasi

Situs	PageRank pada saat-n					
	0	1	2	5	15	35
Situs 1	0.5	0.8158333	0.4974319	0.7423612	0.7335149	0.7331237
Situs 2	0.1	0.6902977	1.6770952	1.5374748	1.5205788	1.5196739
Situs 3	0.25	0.9830843	0.7770246	0.7398189	0.7334248	0.7331235
Situs 4	1	0.1510200	0.3225598	0.2547667	0.2540993	0.2540519
Situs 5	0.006	1.0150582	0.5758126	0.6168192	0.6122765	0.6120703
Situs 6	2	0.5635570	0.4815498	0.4662087	0.4639358	0.4638267
Situs 7	0.1	2.2884685	1.9358580	2.0180791	1.9967227	1.9955959
Situs 8	1.9	1.6996488	1.7109225	1.7053722	1.6893776	1.6885432
Rata – Rata PR		1.025	0.997	1.010	1.000	1.000

Dari **Tabel 1**, terlihat bahwa setiap situs memiliki nilai PageRank yang berbeda hingga sampai pada n = 15. Pada langkah 15 keatas, nilai PageRank mulai menjadi statis karena mendekati Stationary Distributionnya. Keadaan ini lebih jelas terlihat apabila data tersebut dijadikan grafik sebagaimana **Gambar 12** dibawah berikut.



Gambar 12 Plot PageRank seiring berjalannya waktu

Berdasarkan hasil grafik tersebut, nilai PageRank mulai menjadi statis ketika iterasi kelima. Apabila iterasi terus dilanjutkan hingga tak hingga, nilai PageRank tidak akan berubah.

IV. Hasil dan Pembahasan

Keadaan awal probabilitas dari setiap state memiliki pengaruh yang sangat kecil terhadap Stationary Distributionnya. Ini dibuktikan dengan **Gambar 12** yang menunjukan bahwa setiap nilai akan mengalami perubahan nilai yang sangat drastis agar memenuhi teorema Markov Chain.

Berdasarkan hasil percobaan pada graf di **Gambar 9**, terdapat urutan situs sesuai dengan relevansinya. Situs 7 menduduki puncak relevansi dengan nilai PageRank sebesar 1,9959. Situs 4 memiliki nilai PageRank paling kecil sebesar 0.254. Menurut model PageRank, situs yang memiliki nilai PageRank yang tertinggi akan diprioritaskan muncul paling awal ketika pengguna melakukan search didalam suatu search engine.

Meskipun Situs 7 mendapati rujukan terbanyak, hal tersebut bukanlah faktor utama mengapa nilai PageRank Situs 7 sangat tinggi. Situs 7 mendapati rujukan oleh situs 8 yang memiliki PageRank yang tinggi sehingga bobot “suara” yang diberikan lebih besar. Sistem seperti ini berguna untuk mencegah seseorang untuk membuat page yang banyak untuk mendorong page yang ia miliki agar menaiki puncak peringkat. Ini dikarenakan dibutuhkan juga page yang relevan agar suatu situs dapat terdorong nilainya

Pada dasarnya, PageRank menggunakan teori Markov Chain dengan sedikit modifikasi agar Teorema Ergodic selalu terpenuhi. Adanya damping factor, disimbolkan d pada persamaan, membuat asumsi baru jika pengguna masuk kedalam Transient State, pengguna akan keluar dari state dan masuk ke state yang lain secara acak.

$$\boxed{\vec{\pi}_{n+1} = \vec{\pi}_n \hat{P}} \qquad (8)$$

$$\boxed{\hat{P} = (1-\alpha)P + \frac{\alpha}{N}\left(N \times Nmatrixofall1's\right)} \qquad (9)$$

Dengan probabilitas $\frac{\alpha}{N}$, pengguna akan transisi ke state yang acak.

V. Kesimpulan

PageRank merupakan salah satu penerapan aplikasi Teori Graf yang membuat penciptanya mendirikan perusahaan teknologi bernama Google. Google sendiri membangun teknologi search engine mereka disekitar algoritma PageRank tersebut.

Berdasarkan hasil percobaan yang telah dilakukan, model ini dapat diskalabilitas bersamaan dengan menigkatnya kemajuan teknologi. Biaya waktu yang diperlukan dalam melakukan kalkulasi dari program ini dapat dikurangi. Perlu diingat pula, bahwa program yang penulis buat berdasarkan sumber resmi yang diterbitkan pihak pertama, seperti Google, dan pihak lainnya yang meneliti hal yang sama.

VII. Ucapan Terima Kasih

Penulis bersyukur kepada Tuhan Yang Maha Esa atas kelancaran dalam penulisan makalah berjudul "Aplikasi Teori Graf untuk Mengurutkan Web Pages Menggunakan Markov Chain dalam Algoritma PageRank". Penulis juga mengucapkan terima kasih kepada Dr. Nur Ulva Maulidevi, S.T., M.Sc., sebagai dosen dan pembimbing dalam mata kuliah IF2120 Matematika Diskrit pada tahun ajaran 2023/2024. Tak lupa, penulis juga mengucapkan terima kasih kepada keluarga, teman-teman, dan pihak lain yang telah memberikan kontribusi untuk menyelesaikan makalah ini. Semoga Tuhan Yang Maha Esa membalas semua kebaikan dengan kebaikan yang berlipat ganda.

Referensi

[1] “siteefy,” [Online]. Available: https://siteefy.com/how-many-websites-are-there/. [Diakses 9 Desember 2023].

[2] R. Munir, “Matematika Diskrit,” 2023. [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2023-2024/19-Graf-Bagian1-2023.pdf. [Diakses 9 Desember 2023].

[3] H. Scheepers, “Markov Chain Analysis and Simulation using Python,” 20 November 2019. [Online]. Available: https://towardsdatascience.com/markov-chain-analysis-and-simulation-using-python-4507cee0b06e. [Diakses 10 December 2023].

[4] I. Rogers, “The Google Pagerank algorithm and how it works.,” 2002.

[5] N. Nerd, “Markov Chains: Recurrence, Irreducibility, Classes,” 3 November 2020. [Online]. Available: https://www.youtube.com/watch?v=VNHeFp6zXKU. [Diakses 10 Desember 2023].

[6] S. Brin dan L. Page, “The anatomy of large-scale hypertextual Web search engine,” *Computer Networks and ISDN Systems 30,* pp. 107-117, 1998.

[7] R. Hanggaraksa, “PageRank-Simulator,” 11 December 2023. [Online]. Available: https://github.com/raflyhangga/PageRank-Simulator. [Diakses 11 Desember 2023].

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, 9 Desember 2023

Raden Rafly Hanggaraksa Budiarto
13522014