Perbandingan Implementasi Algoritma-algoritma Pagerank pada Satu Mesin Komputer

Farhan Herdian Pradana[1], Muhammad Eka Suryana[2], Med Irzal[3]

Program Studi Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Negeri Jakarta, DKI Jakarta, Indonesia
[1]farhan.herdia123@gmail.com, [2]eka-suryana@unj.ac.id, [3]medirzal@unj.ac.id

Abstrak

Algoritma Pagerank merupakan algoritma mengurutkan halaman web pada search engine Google. Masalah pada Algoritma Pagerank adalah memerlukan memori utama yang cukup besar, dan tidak mungkin dilakukan pada satu mesin komputer dengan memori utama yang terbatas. Akan dicari algoritma alternatif dari Algoritma Pagerank Original buatan Google dengan membandingkannya pada algoritma-algoritma Pagerank dari penelitian lainnya dengan membandingkan kecepatan, penggunaan memori utama, dan kemiripan hasil. Penelitian dilakukan dengan melakukan pengkodean terhadap algoritma Pagerank Original, algoritma Distributed Pagerank Computation (DPC), algoritma Modified DPC (MDPC), dan algoritma Random Walker. Semua kode program dijalankan pada dataset dan dibandingkan kecepatan, penggunaan memori utama, dan kemiripan hasil akhirnya. Khusus hasil akhir, hasil dari algoritma Random Walker dijadikan acuan karena dasar dari Algoritma Pagerank adalah Random Walker. Hasilnya Algoritma Pagerank Original unggul dari sisi kecepatan dan hasil yang mirip dengan hasil Algoritma Random Walker. Sedangkan algoritma DPC dan MDPC unggul di penggunaan memori utama yang lebih hemat, sehingga cocok untuk satu mesin komputer yang memiliki memori utama yang terbatas, tetapi dengan catatan mengorbankan kecepatan yang lebih lambat dan hasil yang tidak mirip terhadap Random Walker.

I. PENDAHULUAN

Internet adalah jaringan luas yang membuat jaringan komputer seluruh dunia yang dijalankan oleh perusahaan, pemerintahan, universitas, dan organisasi lainnya untuk dapat saling berkomunikasi (Sample, 2018). Internet memiliki banyak kegunaan. Di bidang komunikasi, Internet melahirkan *Voice over Internet Protocol* (VoIP) dan surat elektronik (*email*). Di bidang pengiriman data, internet memungkinkan pengguna untuk dapat mengunggah berkas ke *file server* untuk dibagikan ke orang lain atau supaya bisa diakses di mana saja. Yang paling populer, selain dari dua bidang tersebut, internet melahirkan World Wide Web (WWW) yang memungkinkan situs web atau biasa disebut *website* untuk bisa diakses oleh semua orang.

Website adalah sekumpulan halaman web yang saling terkait dan berada pada nama domain yang sama. Website dapat dibuat dan dipelihara oleh seorang individu, grup, perusahaan, atau organisasi lain dengan berbagai macam tujuan (Techopedia, 2020). Dengan adanya *website*, kegunaan internet menjadi semakin lebih luas, lebih berkualitas, dan lebih mudah digunakan. *Website* yang menggunakan protokol HTTP, memungkinkan untuk mengirim berkas HTML, CSS, dan JavaScript sehingga internet da-

pat menampilkan visual yang lebih ramah terhadap pengguna dan menjadi media hiburan baru. Tidak heran, *website* terus tumbuh pesat. Pada tahun 1992 hanya terdapat sepuluh *website*, lalu pada tahun 1994 angka ini bertumbuh menjadi 3.000 *website*, dan semakin bertumbuh pesat pada tahun 2021 menjadi kurang lebih 1,88 miliar *website* (Amstrong, 2021).

Dengan meledaknya jumlah halaman web, memunculkan tantangan baru dalam memperoleh informasi dari web. Pengguna biasanya menelusuri web dengan mengunjungi graf *link* yang terdapat pada halaman web, biasanya dimulai pada situs kumpulan index halaman web berkualitas tinggi yang dipelihara oleh manusia seperti Yahoo.com, atau menggunakan *search engine* (Brin dan Page, 1998). Seiring perkembangan zaman, *search engine* Google menjadi *search engine* teratas dengan pengguna terbesar di dunia dengan penguasaan pasar sebesar 91% (GlobalStatCounter, 2022). Google awalnya merupakan proyek Sergey Brin dan Larry Page saat mereka mengambil gelar Ph.D di Universitas Stanford dengan tujuan membuat *search engine* yang lebih berkualitas dibandingkan *search engine* lain (Brin dan Page, 1998). Kunci kesuksesan dari Google terletak pada Pagerank. Pagerank merangking halaman web berdasarkan kepentingan relatif (*relative importance*) suatu halaman web berdasarkan graf tautan web (Page dkk., 1999). Sebelum adanya Pagerank, *search engine* lain biasanya merangking suatu halaman web dengan menghitung banyaknya *backlink* yang merujuk halaman tersebut (Page dkk., 1999). Metode tersebut akan menjadi mudah untuk dimanipulasi pemilik halaman web yang ingin mendapatkan *ranking* teratas dengan membuat halaman web lain yang berisi *link* yang menunjuk pada halaman web tujuan sebanyak-banyaknya. Pagerank menjawab permasalahan tersebut dengan melakukan normalisasi pada jumlah *backlink* suatu halaman web (Brin dan Page, 1998). Hal ini yang membuat hasil pencarian Google menjadi lebih relevan dibandingkan hasil pencarian *search engine* lain.

Setiap *search engine* memiliki arsitektur berbeda-beda. Arsitektur Google dipilih dan dijadikan acuan dalam topik penelitian peningkatan arsitektur *search engine* yang merupakan penelitian induk dari judul penelitian ini karena keunggulannya dibandingkan *search engine* lain. Arsitektur Google dapat dilihat pada gambar **??**. Modul *crawler* dan pendukungnya yang ditandai dengan warna biru muda sudah dibuat pada penelitian sebelumnya yang berjudul "Perancangan *Crawler* Sebagai Pendukung Pada Search Engine" oleh Qoriiba (2021). Pada penelitian tersebut digunakan algoritma *modified similarity based crawling* dan selanjutnya hasil dari *crawling* disimpan kedalam *database* MySQL (Qoriiba, 2021). Selanjutnya penelitian berjudul "Perancangan Arsitektur *Search Engine* dengan Mengintegrasikan *Web Crawler*, Algoritma Page *ranking*, dan Dokumen *ranking*" oleh Khatulistiwa (2022). Pada penelitian tersebut digabungkan modul *crawler* dari penelitian Qoriiba (2021), Pagerank, dan *searcher* pada penelitian lain sebelumnya menjadi *search engine* berbasis konsol (Khatulistiwa, 2022). Pada modul *indexer* dilakukan penelitian oleh Pratama (2022) berjudul "Perancangan Modul Pengindeks pada *Search Engine* Berupa *Induced Generalized Suffix Tree* untuk Keperluan Perangkingan Dokumen" dan Zalghornain (2022) berjudul "Rancang Bangun Sistem Pencarian Teks dengan Menggunakan Model *Continuous-Bag-of-Words* dan Model *Continuous Skip-Gram* pada Koleksi Dokumen".

Dalam melakukan perangkingan halaman web, Pagerank dapat didefinisikan pada persamaan 1. Dimana u adalah halaman web. F_u adalah himpunan halaman u yang menunjuk halaman lain dan B_u adalah himpunan halaman yang menunjuk ke u. $C_u =$

$|F_u|$ adalah jumlah *link* dari u dan c adalah faktor yang digunakan untuk normalisasi (sehingga jumlah total *ranking* semua halaman web adalah konstan) dan $c < 1$. $E(u)$ adalah vektor yang berkorespondensi dengan *ranking* halaman web. $||\pi'||_1 = 1$. Iterasi perhitungan terus dilakukan sampai konvergen. Jika diubah kedalam persamaan matriks, maka persamaan 1 dapat diubah menjadi persamaan 2. Dimana X adalah matriks persegi yang baris dan kolomnya berkorespondensi dengan halaman web, dengan elemen $X_{u,v} = \frac{1}{C_u}$ jika terdapat *link* pada halaman u ke halaman v atau $X_{u,v} = 0$ jika tidak ada.

$$\pi'(u) = c \sum_{v \in B_u} \frac{\pi'(v)}{C_v} + cE(u) \qquad (1)$$

$$\pi' = c(X\pi' + E) \qquad (2)$$

Dasar intuitif dari persamaan 1 adalah *random walks* pada sebuah graf. Anggap pengguna internet sebagai *"random surfer"* yang terus meng-klik *link* selanjutnya secara acak. Namun, jika pengguna terjebak pada sebuah lingkaran halaman web (*link* yang diklik terus menampilkan halaman web yang pernah dikunjungi sebelumnya), tidak mungkin pengguna akan terus mengikuti *link* tersebut, melainkan pengguna akan langsung pindah ke halaman lain. Oleh sebab itu faktor E dipakai untuk memodelkan perilaku ini (Pengguna akan bosan dan langsung lompat ke halaman web lain berdasarkan distribusi pada E) (Page dkk., 1999). E dapat didefinisikan oleh pengguna (*user-defined parameter*) dan nilai elemennya dapat diisi dengan nilai yang seragam atau berbeda-beda. Menariknya, jika nilai elemen E dibuat berbeda-beda, maka dapat membuat Pagerank yang dipersonalisasi (Page dkk., 1999).

Walaupun persamaan Pagerank terlihat sederhana, terdapat masalah dari sisi ruang dan waktu. Dari sisi ruang, misal terdapat 1000 halaman web, maka akan terbentuk 1000x1000 matriks X. Misal tiap sel elemen X memerlukan memori 8 *Byte*, maka untuk membentuk 1000x1000 matriks X, tanpa menghitung alokasi *overhead* memori, memerlukan 8 *Mega Byte* (MB) memori utama (Lihat tabel 1). Di internet terdapat miliaran *website* dan setiap *website* dapat memiliki lebih dari 1 halaman, sehingga untuk bisa membentuk matriks X membutuhkan memori utama dengan kapasitas mencapai *Peta Byte* atau bahkan *Exa Byte*. Hal tersebut sangat tidak mungkin dilakukan pada komputer pribadi biasa yang memori utamanya hanya pada kisaran 4 GB sampai 32 GB, yang berarti ketika program dieksekusi langsung *crash* karena memori yang tidak cukup. Dari sisi waktu, proses *string matching* untuk mengakses nilai *ranking* suatu halaman web berdasarkan *string* URLnya juga memiliki kompleksitas waktu yang besar yakni O(NM), jika dilakukan dengan cara dicari satu-persatu. Beruntung *database* seperti MySQL menggunakan B-Tree dalam mengindeks datanya (MySQL-Doc, 2022). B-tree memiliki kompleksitas waktu kecil yakni hanya O(log(n)) (GeeksForGeeks, 2022).

Table 1: *Tabel alokasi memori utama untuk membentuk matriks X*

Jumlah Halaman Web	Memori (*Byte*)
256	524416
512	2097280
1024	8388736
2048	33554560
4096	134217856
8192	536871040
16384	2147483776
32768	8589934720

Telah dilakukan penelitian tentang Pagerank yang terdistribusi dengan metode *iterative aggregation-disaggregation* (IAD) dengan *Block Jacobi smoothing* (Zhu dkk., 2005). Sederhananya, dilakukan *divide-and-conquer* dengan mengelompokan halaman web

berdasarkan *domain*-nya lalu dihitung Pagerank lokalnya dan disatukan dengan metode komunikasi yang hemat memori dengan sebuah koordinator (Zhu dkk., 2005). Hasilnya, ditemukan sebuah metode Pagerank terdistribusi yang bisa dijalankan pada memori utama kecil dan lebih cepat konvergen sehingga menghemat waktu (Zhu dkk., 2005). Oleh karena itu, akan dicari algoritma Pagerank alternatif yang dapat dijalankan pada satu mesin komputer dengan memori utama terbatas, dengan cara melakukan perbandingan implementasi beberapa algoritma Pagerank pada satu mesin komputer.

II. Kajian Teori

World Wide Web (WWW)

WWW merupakan proyek Tim Berner-lee bersama teman-temannya, yang ditunjukan pada publik pada tahun 1991. WWW didesain untuk membawa sebuah semesta informasi global menggunakan teknologi yang ada. Dengan adanya WWW manusia dapat mengakses seluruh informasi melalui sebuah *platform browsing* apapun. Pada masa itu, sudah ada teknologi serupa yang membuat WWW mungkin untuk dilakukan. Sistem *hypertext* yang sudah ada pada saat itu, terbatas pada sistem *file* lokal atau terdistribusi dan kadang hanya dikembangkan pada *platform* tertentu. Selain itu, juga ada sistem pengambilan dan akses informasi seperti Alex, Gopher, Prospero, dan WAIS yang sudah mencakup area yang luas, tetapi tanpa fungsionalitas *hypertext*. WWW menggabungkan teknik *hypertext*, *information retrieval*, dan *wide area networking* (Berners-Lee dkk., 1992).

Model yang dipakai WWW menggunakan dua paradigma dari *hypertext link* dan pencarian teks yang saling melengkapi. Gambar 1 menunjukkan bagaimana sebuah web yang berisi informasi pribadi terbentuk

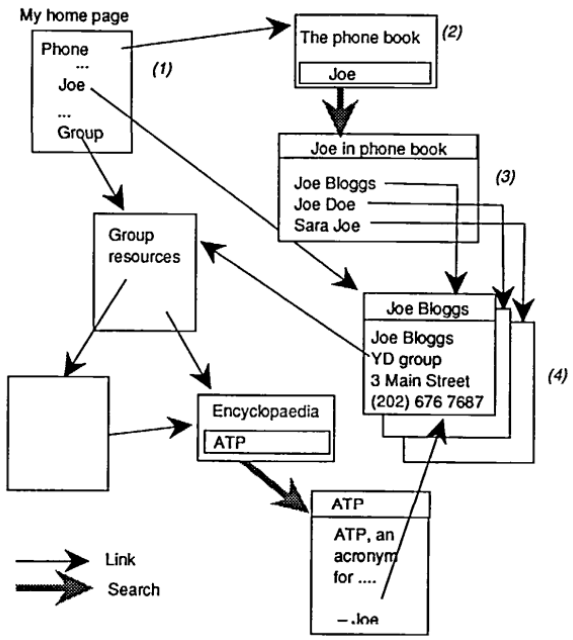

Figure 1: *Gambar sebuah web yang terdiri atas kumpulan link dan indeks (Berners-Lee dkk., 1992)*

pada paradigma ini. Pembaca mulai pada halaman *home* (1) lalu menggunakan *link* grup atau publik untuk mencari bahan informasi. Indeks seperti buku telepon (2) ditampilkan sebagai dokumen yang memungkinkan untuk melakukan input pencarian. Hasil dari pencarian berupa dokumen *hypertext* virtual (3) yang menunjuk pada dokumen yang ditemukan (4) (Berners-Lee dkk., 1992).

Teori Graf

Graf adalah sebuah representasi dari himpunan titik (*node / vertice*) dan bagaimana titik-titik tersebut saling terhubung tanpa memperdulikan sifat metriknya (Wilson, 1996). Pada gambar 2 merupakan contoh graf, dengan P, Q, R, S, T merupakan titik, dan masing-masing terhubung dengan garis (*edge*). Garis yang menghubungkan titik P dan S disebut dengan PS, sedangkan garis yang menghubungkan titik Q dan T disebut dengan QT. Persilangan antara PS dan QT tidak disebut sebagai titik, karena keduanya tidak saling bersilangan, melainkan saling

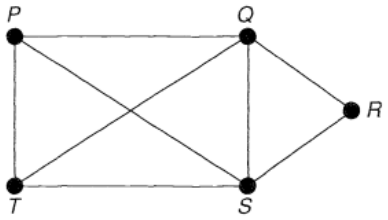

Figure 2: *Contoh graf (Wilson, 1996)*

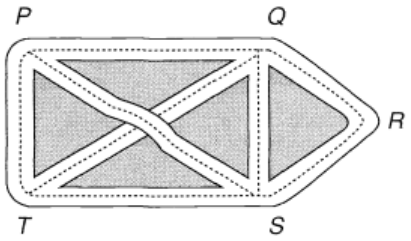

Figure 3: *Contoh peta jalan yang dapat diandaikan sebagai graf (Wilson, 1996)*

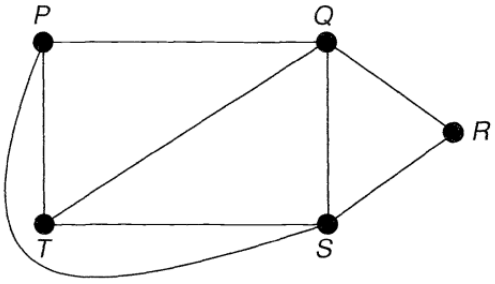

Figure 4

melompati layaknya gambar 3. Selanjutnya, kedua graf dianggap sama, jika dan hanya jika titik yang berkorespondensi sama-sama terhubung dengan garis yang sama dengan garis pada graf lainnya (Wilson, 1996). Sebagai contoh, graf pada gambar 4 merupakan graf yang sama dengan graf pada gambar 2 (Wilson, 1996).

Garis pada graf dapat diberikan arah. Garis pada graf yang berarah disebut sebagai busur (*arc*). Graf yang memiliki arah pada garisnya disebut dengan graf berarah (*directed graph* / *digraph* / digraf) yang terdiri atas himpunan titik dan busur (Wilson, 1996). Pada digraf di gambar 5 terdapat himpunan titik u, v, w, dan z, dengan busur uv, vv, $vw(2\times)$, wv, wu, dan zw. Sebuah digraf disebut sebagai digraf sederhana jika himpunan busurnya tidak ada yang sama (*distinct*) dan tidak memiliki *loop* (contoh: busur vv) (Wilson, 1996). Digraf pada gambar 6 adalah contoh digraf sederhana.

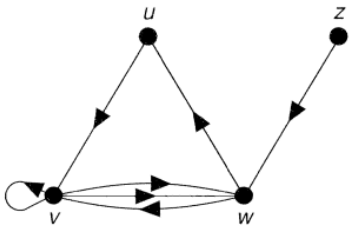

Figure 5: *Contoh digraf (Wilson, 1996)*

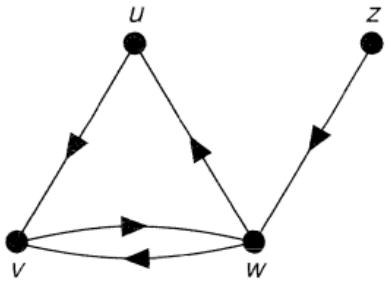

Figure 6: *Contoh digraf sederhana (Wilson, 1996)*

Pagerank

Jika World Wide Web diibaratkan pada sebuah graf berarah, halaman web adalah titik graf, *link* adalah garis. Lalu *link* yang menunjuk keluar dari titik disebut *forward link*, sedangkan *link* yang menunjuk kedalam titik disebut *backlink*. Sangat sulit untuk mengetahui semua *backlink* suatu halaman web, tetapi sangat mudah untuk mengetahui semua *forward link* suatu halaman web yaitu dengan cara mengunduh halaman web tersebut (Page dkk., 1999). Setiap halaman web memiliki jumlah *backlink* yang beragam. Pada saat Pagerank diteliti, halaman *home* NetScape memiliki 62.804 *backlink* dibandingkan halaman web kebanyakan yang hanya memiliki beberapa *backlink*. Secara umum suatu halaman web jika memiliki banyak *backlink* dapat dianggap lebih penting daripada halaman web lain yang memiliki lebih sedikit *backlink*. Perhitungan jumlah sitasi sederhana pernah digunakan

untuk memprediksi pemenang Nobel masa depan (Page dkk., 1999).

$$P_{ij} = \begin{cases} \frac{d}{C_j} + \frac{(1-d)}{N} & j \rightarrow i \\ \frac{(1-d)}{N} & j \not\rightarrow i \text{ dan } C_j \neq 0 \\ \frac{1}{N} & C_j = 0 \end{cases} \quad (3)$$

Graf pada WWW, dapat direpresentasikan sebagai matriks transisi P. Matriks P didefinisikan pada persamaan 3. C_j adalah jumlah *forward link* dari halaman j. $j \rightarrow i$ adalah halaman j memiliki *link* menuju halaman i.

Selanjutnya algoritma Pagerank didefinisikan pada algoritma 1

Algoritma 1 Pagerank (Zhu dkk., 2005)

1: Definisikan π^0 awal-awal
2: $k \leftarrow 0$
3: $\pi^{\sim k+1} \leftarrow P\pi^k$
4: $\pi^{k+1} \leftarrow \frac{\pi^{\sim k+1}}{||\pi^{\sim k+1}||_1}$
5: Jika $||\pi^{k+1} - \pi^k|| < \epsilon$ berhenti dan kembalikan nilai π^{k+1}
6: $k \leftarrow k + 1$
7: Ulangi langkah 3

Distributed Pagerank Computation (DPC)

Telah dijelaskan sebelumnya, masalah dari algoritma Pagerank biasa adalah besarnya memori utama yang dibutuhkan untuk bisa menyimpan matriks X (lihat persamaan 2). Oleh karena itu, dirumuskan algoritma Pagerank terdistribusi. DPC, dirumuskan oleh Zhu dkk. (2005), memakai mekanisme interaksi sederhana antara *cluster* dan lalu lintas komunikasi rendah. Ditinjau dari perspektif matematika, dibuktikan bahwa algoritma DPC setara dengan metode *Iterative Aggregation-Disaggregation* (IAD) dengan *Block Jacobi smoothing*. DPC juga memiliki keunggulan dibandingkan dengan algoritma Pagerank biasa yaitu, matriks-matriks dipecah menjadi matriks agregasi dan matriks lokal sehingga ukurannya cukup kecil untuk disimpan di memori utama, sehingga mempercepat komputasi karena tiap iterasi memerlukan sedikit operasi I/O pada *disk*. Selanjutnya, Vektor Pagerank lokal konvergen lebih cepat pada beberapa *cluster* tertentu, berbeda dengan Pagerank biasa karena terdapat komputasi tidak perlu pada *cluster* yang sudah konvergen (Zhu dkk., 2005).

Jika *link* pada kumpulan web dibuat kedalam graf, maka graf tersebut akan memiliki sebuah struktur menyerupai blok, karena mayoritas dari *link* tersebut bersifat *intra-host*, merujuk halaman yang masih di dalam *host* yang sama. Oleh karena itu, jika dilakukan perjalanan acak pada kumpulan web tersebut dapat dilihat sebagai rantai Markov *Nearly Completely Decomposable* (NCD) (Zhu dkk., 2005). Rantai Markov NCD adalah rantai Markov yang dapat dipartisi sehingga peluang dari keadaan awal ke keadaan selanjutnya lebih sering menunjuk keadaan yang berada di dalam partisinya dibandingkan di luar partisinya (Kontovasilis dan Mitrou, 1995).

Sebelum langsung membahas algoritma DPC, akan didefinisikan beberapa notasi terlebih dahulu. $e = (1, ..., 1)^T$. Misal G adalah himpunan bilangan bulat $\{1, ..., N\}$. Misal $G_1, ...G_n, n \leq N$ adalah grup agregasi dari elemen-elemen di G. Himpunan-himpunan $G_i, i = 1, ..., n$, adalah saling lepas dan $\cup_{i=1}^{n} G_i = G$. Misal N_i adalah ordo dari himpunan G_i, atau jumlah elemen-elemen di G_i (Zhu dkk., 2005).

Misal R adalah matriks agregasi $n \times N$, yang memenuhi persamaan 4 (Zhu dkk., 2005).

$$R_{ij} = \begin{cases} 1 & j \in G_i \\ 0 & \text{lainnya} \end{cases} \quad (4)$$

Dilakukan partisi pada vektor positif π menjadi $(\pi_1^T, \pi_2^T, ..., \pi_n^T)^T$ berdasarkan $\{G_i\}$.

π_i adalah subvektor dengan dimensi N_i (Zhu dkk., 2005).

Maka dapat didefinisikan matriks disagregasi $S(\pi)$ $N \times n$ sebagai persamaan 5 (Zhu dkk., 2005).

$$S(\pi) = \begin{pmatrix} S(\pi)_1 & 0 & \dots & 0 \\ 0 & S(\pi)_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & S(\pi)_n \end{pmatrix} \quad (5)$$

Dimana $S(\pi)_i$ adalah sebuah kolom vektor yang mewakili *censored stationary distribution* dari halaman-halaman di *cluster* G_i. Ingat bahwa $RS(\pi) = I$ (Zhu dkk., 2005).

Matriks transisi P dipartisi menjadi beberapa blok berdasarkan $\{G_i\}$ menjadi seperti persamaan 6 (Zhu dkk., 2005).

$$P = \begin{pmatrix} P_{11} & P_{12} & \dots & P_{1n} \\ P_{21} & P_{22} & \dots & P_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ P_{n1} & P_{n2} & \dots & P_{nn} \end{pmatrix} \quad (6)$$

Dilambangkan blok baris ke-i sebagai persamaan 7

$$P_{i*} \triangleq (P_{i1}, \dots, P_{in}) \quad (7)$$

dan dilambangkan blok kolom ke-i sebagai persamaan 8

$$P_{*i} \triangleq \begin{pmatrix} P_{1i} \\ \vdots \\ P_{ni} \end{pmatrix} \quad (8)$$

Setiap blok diagonal P_{ii} adalah matriks persegi dan merupakan matriks *intra-cluster link* dari *cluster* G_i, sementara blok-blok di luar diagonal merupakan struktur *link* antar-*cluster*. Selanjutnya, matriks agregat $A = RPS(\pi)$ adalah matriks transisi antar *cluster*. Maka dapat dibuat algoritma DPC pada algoritma 2 (Zhu dkk., 2005).

1: Buat matriks transisi lokal untuk tiap *cluster* G_i berdasarkan P

$$Q_i = LocalTransitionMatrix(G_i) \forall G_i \in G \quad (9)$$

2:

$$\pi_i^0 = Pagerank(Q_i, \frac{e}{N_i}, \epsilon) \forall G_i \in G \quad (10)$$

Ket: $e = [1, 1, ..., 1]^T; N_i \rightarrow$ jumlah anggota G_i

3: Inisialisasi $k = 0$

4:

$$A^k = RPS(\pi^k) \quad (11)$$

Ket: $R \rightarrow$ persamaan 4; $P \rightarrow$ persamaan 3; $S(\pi) \rightarrow$ persamaan 5

5:

$$z^k = Pagerank(A^k, \frac{e}{n}, \epsilon) \quad (12)$$

Ket: $n \rightarrow$ banyaknya anggota G

6: $\forall G_i \in G$ buat sebuah *extended local transition matrix* $(N_i + 1) \times (N_i + 1)$. Dimana skalar α^k membuat jumlah nilai kolom dari B_i^k adalah satu

$$B_i^k = \begin{pmatrix} P_{ii} & \frac{(P_{i*}S(\pi^k)z^k - P_{ii}\pi_i^k z_i)}{(1-z_i^k)} \\ e^T P_{*i} & \alpha^k \end{pmatrix} \quad (13)$$

7: Hitung vektor *extended local Pagerank*. Dimana β_i^{k+1} adalah skalar

$$\begin{pmatrix} \omega_i^{k+1} \\ \beta_i^{k+1} \end{pmatrix} = Pagerank(B_i^k, \frac{e}{(N_i+1)}, \epsilon) \quad (14)$$

8:

$$\pi_i^{\sim k+1} = \frac{1 - z_i^k}{\beta_i^{k+1}} \omega_i^{k+1} \quad (15)$$

9:

$$\pi^{k+1} = \frac{\pi^{\sim k+1}}{||\pi^{\sim k+1}||_1} \quad (16)$$

10: $k = k + 1$

11: Jika persamaan 17 terpenuhi, berhenti dan berikan π^k sebagai hasil akhir. Jika sebaliknya, kembali ke langkah 11

$$||\pi^{k+1} - \pi^k|| < \epsilon \quad (17)$$

Algoritma 2 Algoritma DPC (Zhu dkk., 2005)

Table 2: *Example single column table.*

Location		
East Distance	West Distance	Count
100km	200km	422
350km	1000km	1833
600km	1200km	890

Results

Referencing a table using its label: Table 2.

Aenean feugiat pellentesque venenatis. Sed faucibus tristique tortor vel ultrices. Donec consequat tellus sapien. Nam bibendum urna mauris, eget sagittis justo gravida vel. Mauris nisi lacus, malesuada sit amet neque ut, venenatis tempor orci. Curabitur feugiat sagittis molestie. Duis euismod arcu vitae quam scelerisque facilisis. Praesent volutpat eleifend tortor, in malesuada dui egestas id. Donec finibus ac risus sed pellentesque. Donec malesuada non magna nec feugiat. Mauris eget nibh nec orci congue porttitor vitae eu erat. Sed commodo ipsum ipsum, in elementum neque gravida euismod. Cras mi lacus, pulvinar ut sapien ut, rutrum sagittis dui. Donec non est a metus varius finibus. Pellentesque rutrum pellentesque ligula, vitae accumsan nulla hendrerit ut.

Referencing a figure using its label: Figure 7.

Aenean porttitor eros non pharetra congue. Proin in odio in dolor luctus auctor ac et mi. Etiam euismod mi sed lectus fringilla pretium. Phasellus tristique maximus lectus et sodales. Mauris feugiat ligula quis semper luctus. Nam sit amet felis sed leo fermentum aliquet. Mauris arcu dui, posuere id sem eget, cursus pulvinar mi. Donec nec lacus non lectus fermentum scelerisque et at nibh. Sed tristique, metus ac vestibulum porta, tortor lectus placerat lorem, et convallis tellus dolor eget ante. Pellentesque dui ligula, hendrerit a purus et,

Figure 7: *Anther of thale cress (Arabidopsis thaliana), fluorescence micrograph. Source: Heiti Paves, https://commons.wikimedia.org/wiki/File:Tolmukapea.jpg.*

volutpat tempor lectus. Mauris nec purus nec mauris rhoncus pellentesque. Quisque quis diam sed est lacinia congue. Donec magna est, hendrerit sed metus vel, accumsan rutrum nibh.

Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Etiam cursus lectus purus, tempus iaculis quam dictum tristique. Nam interdum sapien nec tempor mattis. Quisque id sapien nisi. Mauris vehicula ornare eros vel efficitur. Nulla consectetur, turpis quis fringilla tincidunt, mi neque iaculis lectus, vel commodo elit odio non ex. Duis facilisis, purus ac viverra iaculis, turpis lectus ultrices ante, ac vestibulum ligula magna in libero. Etiam tristique maximus lacinia. Vestibulum hendrerit, lacus malesuada laoreet blandit, sapien velit sollicitudin nunc, eu porttitor urna ligula at lorem. Aliquam faucibus eros in fermentum venenatis. Fusce consectetur congue pellentesque. Suspendisse at nisi sit amet est porttitor cursus. Cras placerat faucibus nunc, a laoreet justo dignissim sit amet.

International Support

àáâäãåèéêëìíîïòóôöõøùúûüÿýñçčšž
ÀÁÂÄÃÅÈÉÊËÌÍÎÏÒÓÔÖÕØÙÚÛÜŸÝÑ

Table 3: *Example two column table with fixed-width columns.*

Location		
East Distance	West Distance	Count
100km	200km	422
350km	1000km	1833
600km	1200km	890

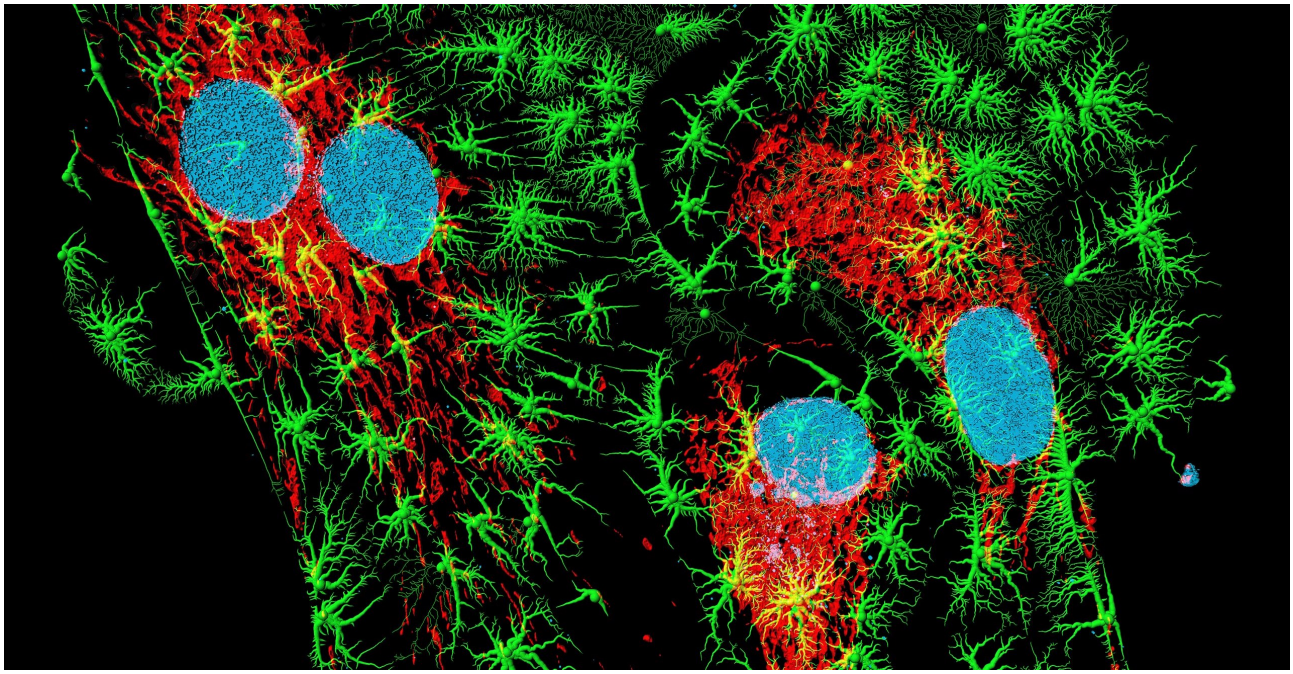

Figure 8: *Bovine pulmonary artery endothelial cells in culture. Blue: nuclei; red: mitochondria; green: microfilaments. Computer generated image from a 3D model based on a confocal laser scanning microscopy using fluorescent marker dyes. Source: Heiti Paves, https://commons.wikimedia.org/wiki/File:Fibroblastid.jpg.*

ßÇŒÆČŠŽ

Links

This is a clickable URL link: LaTeX Templates. This is a clickable email link: vel@latextemplates.com. This is a clickable monospaced URL link: `https://www.LaTeXTemplates.com`.

Discussion

Subsection One

Subsection Two

Subsubsection Example

References

Amstrong, M. (2021). How many websites are there? https://www.statista.com/chart/19058/number-of-websites-online/. Diakses pada 30-07-2022.

Berners-Lee, T., Cailliau, R., Groff, J.-F., dan Pollermann, B. (1992). World-wide web: The information universe. *Internet Research*, 2(1):52–58.

Brin, S. dan Page, L. (1998). The anatomy of a large-scale hypertextual web search engine.

GeeksForGeeks (2022). Introduction of b-tree. https://www.geeksforgeeks.org/introduction-of-b-tree-2/. Diakses pada 28-09-2022.

GlobalStatCounter (2022). Search engine market share worldwide - july 2022. https://gs.statcounter.com/search-engine-market-share#monthly-200901-202208-bar. Diakses pada 28-09-2022.

Khatulistiwa, L. (2022). Perancangan arsitektur search engine dengan mengintegrasikan web crawler, algoritma page ranking, dan document ranking.

Kontovasilis, K. P. dan Mitrou, N. M. (1995). Markov-modulated traffic with nearly complete decomposability characteristics and associated fluid queueing models. *Advances in Applied Probability*, 27(4):1144–1185.

MySQLDoc (2022). How mysql uses indexes. https://dev.mysql.com/doc/refman/8.0/en/mysql-indexes.html. Diakses pada 30-07-2022.

Page, L., Brin, S., Motwani, R., dan Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab.

Pratama, Z. (2022). Perancangan modul pengindeks pada search engine berupa induced generalized suffix tree untuk keperluan perangkingan dokumen.

Qoriiba, M. F. (2021). Perancangan crawler sebagai pendukung pada search engine.

Sample, I. (2018). What is the internet? 13 key questions answered. https://www.theguardian.com/technology/2018/oc is-the-internet-13-key-questions-answered. Diakses pada 30-07-2022.

Techopedia (2020). Website. https://www.techopedia.com/definition/5411/webs Diakses pada 30-07-2022.

Wilson, R. J. (1996). *Introduction to Graph Theory*. Longman Group Ltd.

Zalghornain, M. (2022). Rancang bangun sistem pencarian teks dengan menggunakan model continuous-bag-of-words dan model continuous skip-gram pada koleksi dokumen.

Zhu, Y., Ye, S., dan Li, X. (2005). Distributed pagerank computation based on iterative

aggregation-disaggregation methods. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 578–585.