Implementasi *Single Thread* dan *Multi Thread* pada *Web Crawling*

Yunanti Moga Hasana, dan Budi Setiyono
Departemen Matematika, Institut Teknologi Sepuluh Nopember (ITS)
e-mail: budisetiyono@its.ac.id

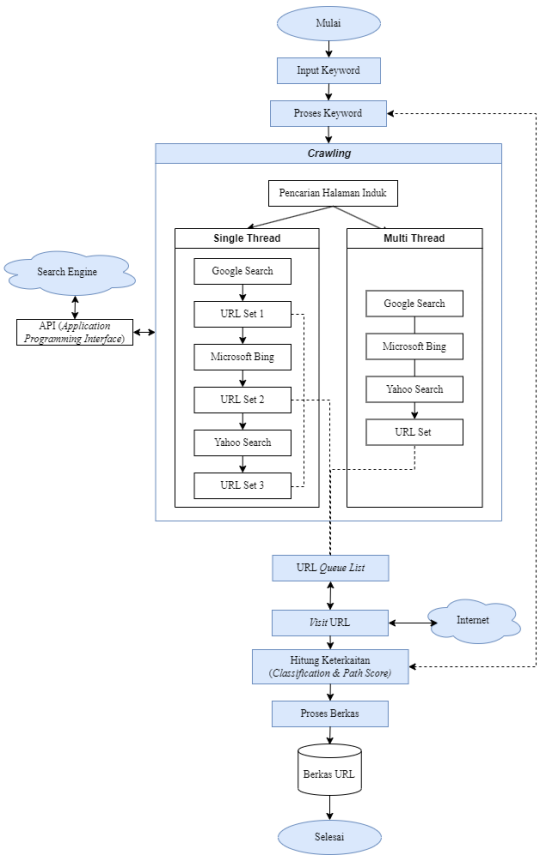
***Abstrak*—Peredaran informasi pada saat ini semakin pesat. Semua informasi dapat dengan mudah diperoleh dari berbagai sumber di internet. Sebagian besar pengguna internet menggunakan mesin pencari dalam memperoleh sebuah informasi. Lebih dari 5 miliar orang di seluruh dunia sekarang menggunakan internet dengan durasi rata-rata online selama 6 jam 53 menit dalam sehari. Sehingga diperlukan mesin pencari yang memiliki kinerja terbaik. Efektivitas mesin pencari tergantung pada *web crawler* dan teknik *crawling* yang digunakan untuk memperoleh data yang diinginkan pengguna. *Web crawling* merupakan elemen penting yang secara otomatis menjelajahi halaman web dan tautan sesuai dengan permintaan pengguna. *Web crawling* dapat diterapkan secara *single thread* maupun *multi thread*. Perbedaan penerapan *web crawling* ini terdapat pada alur kerja dalam menjelajahi sebuah halaman web yang berkaitan dengan *keyword*. *Single thread* akan menjelajahi satu persatu halaman web sehingga dapat lebih cermat pencariannya. Sedangkan *multi thread* menjelajahi halaman web secara bersamaan dalam satu waktu yang mana akan membutuhkan waktu lebih singkat. Oleh karena itu, penelitian ini bertujuan mengimplementasikan *single thread* dan *multi thread* pada *web crawling* untuk mendapatkan metode terbaik dengan menganalisis kinerjanya. Pada penelitian ini dibuat tiga skenario berkaitan dengan banyaknya kata pada kata kunci. Dimana skenario pertama dengan menggunakan satu kata, skenario kedua menggunakan dua kata, dan skenario ketiga menggunakan tiga kata. Hasil terbaik dalam pada penelitian ini adalah metode *Multi Thread* yang memiliki kualitas URL sebesar 62,33% dengan kecepatan selama 59,243 s.**

***Kata Kunci*—*Search Engine*, *Web Crawling*, *Single Thread*, *Multi Thread*.**

I. PENDAHULUAN

PEREDARAN informasi pada saat ini semakin pesat. Semua informasi dapat dengan mudah diperoleh dari berbagai sumber di internet. Lebih dari 5 miliar orang di seluruh dunia sekarang menggunakan internet dengan durasi rata-rata online selama 6 jam 53 menit dalam sehari. Jumlah pengguna ini mewakili 63% dari populasi penduduk dunia yang kini diperkirakan mencapai 7,93 miliar orang.

Sebagian besar pengguna internet menggunakan mesin pencari dalam memperoleh sebuah informasi. Keefektivitasan mesin pencari tersebut tergantung pada *web crawler* dan teknik *crawling* yang digunakan untuk memperoleh data yang diinginkan pengguna [1]. Namun faktanya, internet memiliki data yang sangat besar dimana hampir 30% akan mengalami pemrosesan di waktu yang bersamaan. Dikutip dari laporan International Data Corporation, dunia data global akan tumbuh dari 33 *zettabytes* pada tahun 2018 menjadi 175 *zettabytes* pada tahun 2025 sehingga semua orang yang terhubung setidaknya akan memiliki satu interaksidata setiap 18 detik. Banyaknya interaksi dan data ini membuat sukar menemukan informasi yang sesuai dengan keinginan pengguna.



Gambar 1. Metodologi.

Web crawler merupakan elemen penting yang secara otomatis menjelajahi halaman web dan tautan sesuai dengan permintaan pengguna. *Crawler* menggunakan algoritma *web crawling* yang menemukan aplikasi web secara mekanis, menjelajah *hyperlink*,membuat indeks, dan menyimpan untuk digunakan di masa mendatang. Struktur pengurutan grafik web membuat *crawler* lebih rumit untuk dilintasi [1]. Sehingga diperlukan algoritma yang efisien untuk mengakses konten dan hyperlink agar dapat mengatasi ekspansi data yang cepat di World Wide Web (WWW).Beberapa penelitian metode tentang *web crawling* telah banyak dilakukan, diantaranya mengenai *single thread* dan *multi thread web crawling [1]*. Dimana *web crawling* dalam *single thread* merupakan pengunjungan sebuah halaman dengan menyelesaikan satu tugas terlebih dahulu, lalu dapat menyelesaikan tugas berikutnya. Sedangkan *web crawling* dalam *multi thread* berarti pengunjungan sebuah halaman dengan menyelesaikan semua tugas secara bersamaan dalam satu waktu.

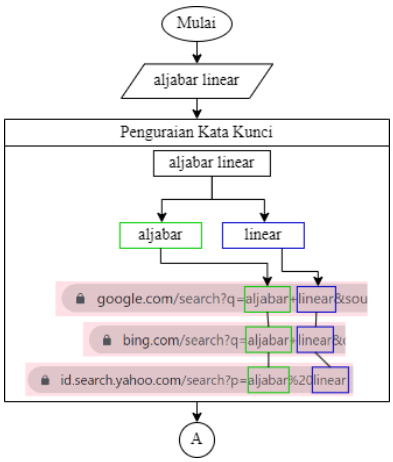
Pada penelitian yang dilakukan oleh Sharma digunakan *web crawling* dengan metode *focused web crawler* yang

Tabel 1.
Rata-rata skenario uji coba 1 *single thread*

Kata Kunci	Jumlah URL	URL Bagus	Kecepatan Komputasi	Kualitas URL
Aljabar	28	25	62817,7	90,91%
Ramadhan	19	13	97610,3	67,02%
Surabaya	20	18	62270	88,73%
Diferensial	30	27	81920,7	90,60%
Kebaya	17	12	36902,5	70,35%

Tabel 2.
Rata-rata skenario uji coba 1 *multi thread*

Kata Kunci	Jumlah URL	URL Bagus	Kecepatan Komputasi	Kualitas URL
Aljabar	104	94	140191,4	90,36%
Ramadhan	17	10	69509,2	62,28%
Surabaya	16	16	42021,8	96,27%
Diferensial	44	31	98343,5	71,17%
Kebaya	25	21	64590	86,23%



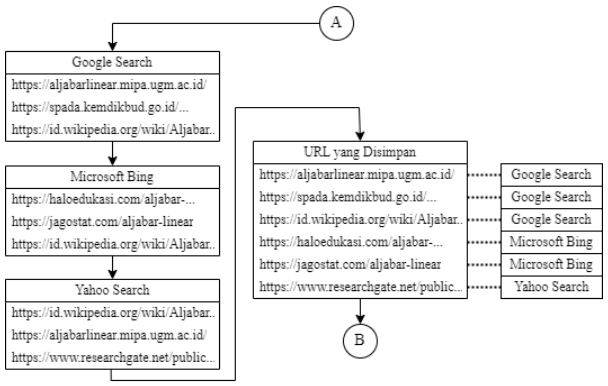
Gambar 2. Penguraian kata kunci pada mesin pencari.

memanfaatkan *Classification and Path Score* untuk menghitung keterkaitansuatu halaman web dengan topik yang diberikan. Pada penelitian ini menerapkan *single thread* karena dalam pengunjungan halamannya dengan menyelesaikan 1 tugas terlebih dahulu lalu dapat menyelesaikan tugas berikutnya. Hasil dari penelitian ini diperoleh tingkat akurasi sebesar 90,1% [2]. Sedangkan pada beberapa penelitian lainnya seperti yang telah ditemukan oleh Vayadande bahwa *multi thread* dapat memberikan efisiensi waktu dan performa lebih baik dalam penerapan *web crawling* [3].

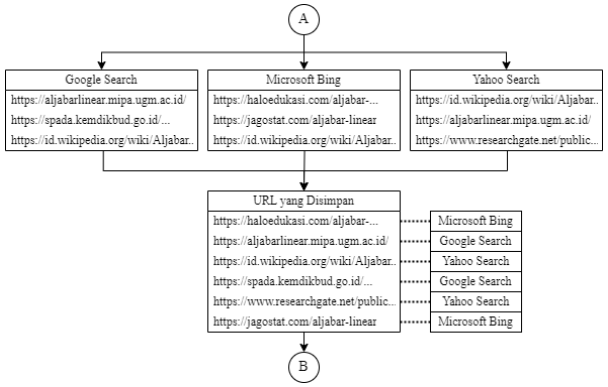
Berdasarkan ulasan di atas, pada penelitian ini dilakukan penelitian tentang implementasi *single thread* dan *multi thread* pada *web crawling*. Penelitian ini dilakukan untuk mendapatkan metode terbaik dalam *web crawling* dengan menganalisis kinerjanya. Adapun inti dari dua metode tersebut yaitu pengambilan data pada beberapa halaman web berdasarkan *keyword* yang diinginkan. *Keyword* dijadikan objek pencarian pada internet dengan menggunakan mesin pencari. Oleh karena itu, dengan objek yang sama diharapkan penelitian ini dapat diperoleh hasil yang tervalidasi dan dapat membantu para pengembang dalam mengoptimalkan kinerja mesin pencari menggunakan *web crawling.*

II. METODOLOGI PENELITIAN

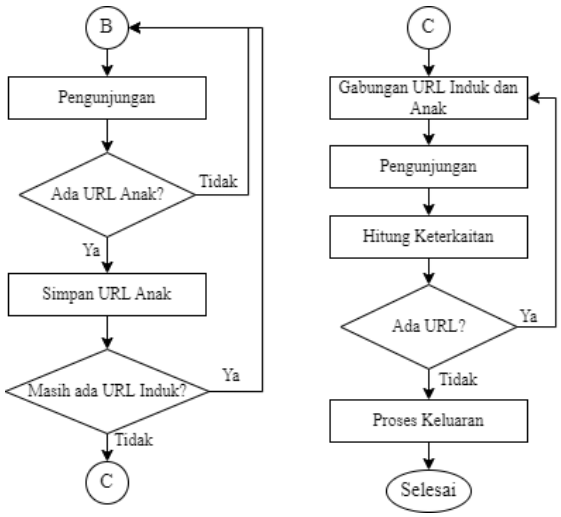
Secara keseluruhan alur metodologi penelitian ini ditampilkan pada Gambar 1. Tahapan metodologi dibagi menjadi beberapa yaitu:



Gambar 3. Contoh pengambilan URL induk pada *single thread*.



Gambar 4. Contoh pengambilan URL pada *multi thread*.



Gambar 5. Perancangan pengambilan URL anak.

A. *Input dan Penguraian Kata Kunci*

Pemrosesan awal dengan menentukan topik (kata kunci) yang berupa kalimat atau kata. Lalu topik akan dideteksi dan diuraikan agar dapat diproses untuk menjadi objek pencarian pada mesin pencari.

B. *Crawling*

Pencarian halaman induk dengan menggunakan *Single Thread* atau *Multi Thread* pada situs web. Situs web yang digunakan yaitu Google Search, Microsoft Bing, dan Yahoo Search. Sehingga menghasilkan kumpulan URL induk (URL *Queue List*).

C. *Pengunjungan dan Hitung Keterkaitan*

Halaman akan dikunjungi dan dihitung keterkaitan isi halaman web tersebut dengan kata kunci yang telah diinputkan menggunakan *classification and path score*.

Tabel 3.
Rata-rata skenario uji coba 2 *single thread*

Kata Kunci	Jumlah URL	URL Bagus	Kecepatan Komputasi	Kualitas URL
Aljabar Linear	19	8	77688,1	39,47%
Persamaan Kuadrat	17	17	59136,1	98,24%
Matematika ITS	11	1	103249	11,43%
Buka Bersama	17	7	47353,1	42,77%
Olahraga Basket	19	5	41557	25%

Tabel 4.
Rata-rata skenario uji coba 2 *multi thread*

Kata Kunci	Jumlah URL	URL Bagus	Kecepatan Komputasi	Kualitas URL
Aljabar Linear	20	12	76932,4	59,30%
Persamaan Kuadrat	18	17	58882,1	98,86%
Matematika ITS	8	2	77993	28%
Buka Bersama	12	7	28612,8	57,26%
Olahraga Basket	14	4	35486,1	27,27%

Setelah dihitung keterkaitan akan dilakukan penyimpanan sementara URL pada program.

D. Proses Berkas Luaran

Setelah tidak ada lagi halaman yang akan dikunjungi, akan dilakukan proses pengunduhan berkas luaran. Berkas tersebut berupa berkas Microsoft Excel berisi kumpulan URL yang berkaitan dengan kata kunci.

III. PERANCANGAN DAN IMPLEMENTASI

Data pada sistem penelitian ini dibagi menjadi tiga yaitu data masukan yang berupa kata atau kalimat, data proses yang digunakan untuk pengolahan data masukan hingga hasil dari proses, dan data keluaran yang berupa data URL dalam bentuk Microsoft Excel (.xls). Pada data proses terdapat perbedaan antara sistem *single thread* dan *multi thread*. Perbedaan tersebut terjadi pada proses *crawling* dan pengambilan URL induk. Adapun perancangan proses dari sistem *single thread* dan *multi thread* dijelaskan sebagai berikut:

A. Perancangan Proses Masukan

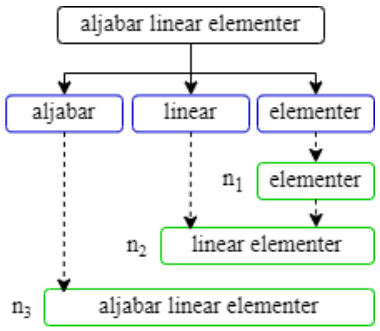
Perancangan proses masukan merupakan proses memasukkan kata kunci yang akan diuraikan berdasarkan "spasi" seperti pada Gambar 2. Penguraian ini bertujuan untuk membentuk URL pencarian halaman induk sesuai dengan *query* masing-masing mesin pencari dan akan disimpan dalam sistem untuk digunakan pada proses perhitungan keterkaitan nanti.

B. Preancangan Proses Penngambilan URL Induk

Dari proses pencarian URL induk melalui penguraian kata kunci diperoleh kumpulan URL induk. URL induk ini yang akan dicatat oleh sistem. Pencarian URL induk memiliki perbedaan pada sistem *single thread* dan *multi thread*. Pada *single thread* akan dilakukan *crawling* satu-persatu pada mesin pencari yaitu dari Google Search terlebih dahulu,

Tabel 5.
Rata-rata skenario uji coba 3 *single thread*

Kata Kunci	Jumlah URL	URL Bagus	Kecepatan Komputasi	Kualitas URL
Aljabar Linear Elementer	14	9	55671,6	64,54%
Persamaan Diferensial Biasa	12	5	31758,2	46,95%
Alat Tulis Kantor	15	9	62247,9	64,14%
Toko Perhiasan Surabaya	19	2	55012,6	11,89%
Tempat Wisata Probolinggo	17	1	76473,6	5,75%



Gambar 2. Penyimpanan kata kunci terurai.

lalu pada Microsoft Bing, dan terakhir pada Yahoo Search. Sehingga URL yang tercatat atau tersimpan pada sistem berurutan dari GoogleSearch, Microsoft Bing, dan Yahoo Search seperti yang diilustrasikan pada Gambar 3.

Sedangkan pada *multi thread* akan dilakukan *crawling* secara bersamaan pada ketiga mesin pencari sehingga hasil URL induk yang tercatat atau tersimpan menjadi acak asal URL-nya seperti yang diilustrasikan pada Gambar 4.

C. Perancangan Proses Pengambilan URL Anak

Setelah diperoleh kumpulan URL induk, URL tersebut akan dikunjungi secara bergantian dan diambil URL di dalamnya seperti yang diilustrasikan pada Gambar 5.

D. Perancangan Proses Pengunjungan dan Perhitungan

Setelah semua URL induk telah dikunjungi dan telah diperoleh semua URL anak yang sesuai dengan kata kunci, akan digabungkan URL induk dan URL anak menjadi kumpulan URL untuk dilakukan pengunjungan dan perhitungan keterkaitan dengan kata kunci. Perhitungan akan dilakukan dengan *Classification and Weight Method*. Pada *classification* digunakan teorema bayes yang dapat disajikan dalam bentuk berikut:

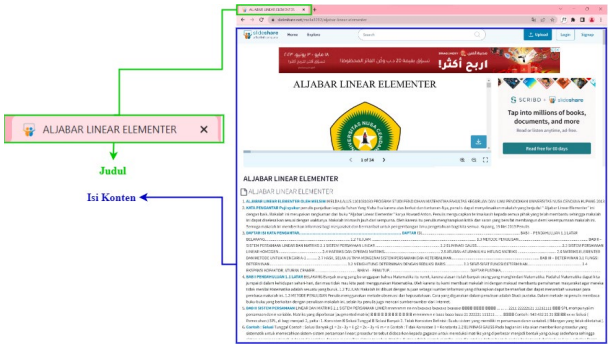
$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (1)$$

dengan $P(B|A)$ mendefinisikan distribusi sebelumnya dari dokumen dan $P(B)$ merupakan total peluang dokumen B.

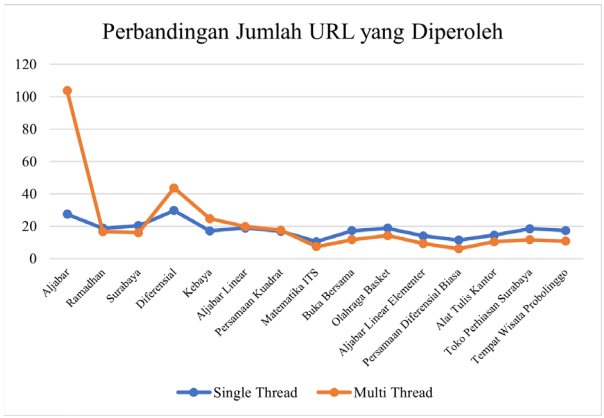
Sehingga $P(A)$ diperoleh dengan pencarian kesamaan kata kunci dengan kata-kata yang ada pada dokumen. Lalu dapat disederhanakan menjadi:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \quad (2)$$

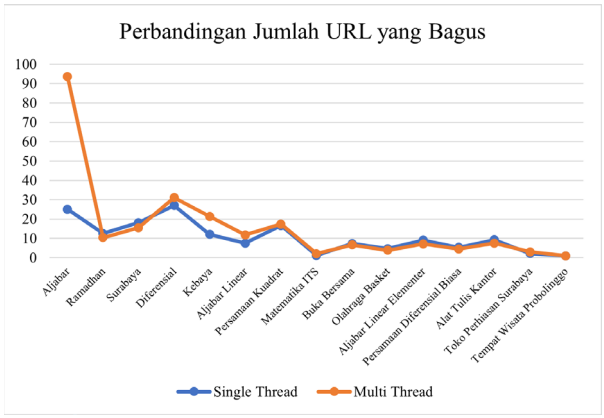
dengan syarat $A \subseteq B$, maka persamaan 2 menjadi:



Gambar 73. Objek perhitungan keterkaitan.



Gambar 84. Perbandingan jumlah URL yang diperoleh.



Gambar 95. Perbandingan jumlah URL yang bagus.

$$P(A|B) = \frac{P(A)}{P(B)} \quad (3)$$

dimana A dan B dilambangkan n yang merupakan indeks kata yang akan dihitung sesuai dengan kata kunci yang telah di uraikan pada proses sebelumnya.

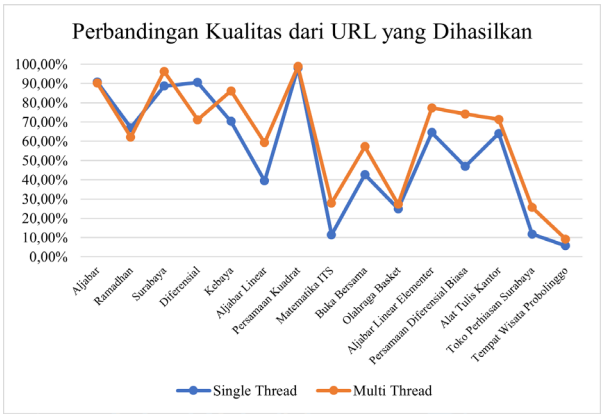
Kata-kata pada kata kunci tersebut akan digunakan seperti pada Gambar 6.

Perhitungan dari peluang kata n dapat diperoleh dari persamaan 4.

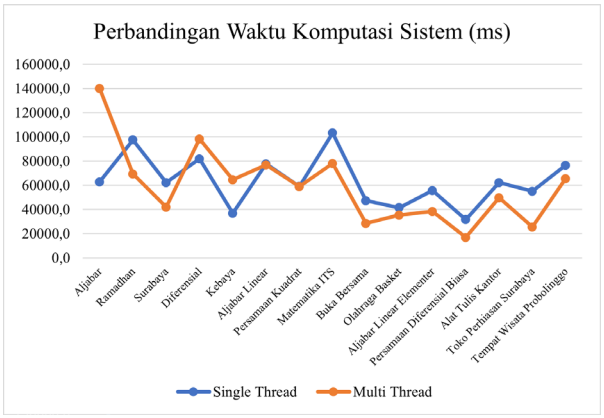
$$P(n) = \begin{cases} \frac{C_n}{E_n} & , C_n < E_n \\ 1 & , C_n \geq E_n \end{cases} \quad (4)$$

dengan C_n merupakan banyaknya kata n yang ditemukan dan E_n merupakan ekspektasi kata n pada kalimat tertentu.

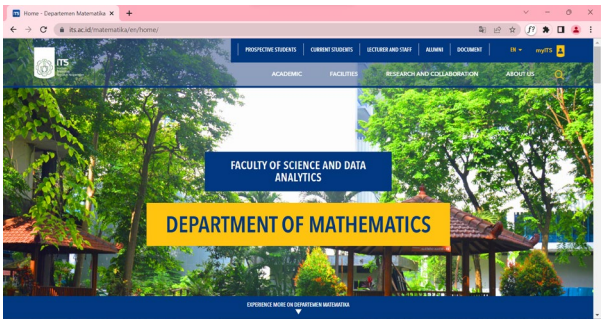
Diasumsikan bahwa nilai ekspektasi suatu kalimat adalah 1. Adapun contoh perhitungan secara manual pada suatu halaman web dengan kata kunci “aljabar linear elementer”. Bagian halaman web yang akan dihitung telihat pada Gambar



Gambar 10. Perbandingan kualitas dari URL yang dihasilkan.



Gambar 11. Perbandingan waktu komputasi sistem.



Gambar 12. Contoh halaman web 1.

7. Pertama akan dihitung pada bagian judul dimana hanya memiliki 1 kalimat dengan ekspektasi kata kunci sebanyak 1 kata.

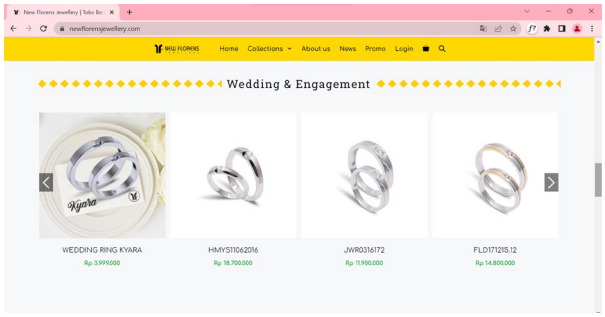
$$P(n_1|n_0) = \frac{P(n_1)}{P(n_0)} = \frac{\frac{C_{n_1}}{E_{n_1}}}{1} = \frac{\frac{1}{1}}{1} = 1$$

$$P\left(n_2|(n_1|n_0)\right) = \frac{P(n_2)}{P(n_1|n_0)} = \frac{\frac{C_{n_2}}{E_{n_2}}}{1} = \frac{\frac{1}{1}}{1} = 1$$

$$P\left(n_3|n_2|(n_1|n_0)\right) = \frac{P(n_3)}{P(n_2|(n_1|n_0))} = \frac{\frac{C_{n_3}}{E_{n_3}}}{1} = \frac{\frac{1}{1}}{1} = 1 \quad (5)$$

dimana $P(n_0) = 1$ karena $P(n_0)$ merupakan total peluang dari halaman web.

Selanjutnya akan dilanjutkan perhitungan peluang pada isis konten dengan kata “elementer” terdapat 20 kata, “linear elementer” terdapat 8 pasangan kata, dan “aljabar linear elementer” terdapat 8 pasangan kata. Serta kalimat pada halaman web Gambar 7 terdapat sebanyak 364 kalimat.



Gambar 6. Contoh halaman web 2.

Gambar 7. Contoh halaman web 3.

$$P(n_1|n_0) = \frac{P(n_1)}{P(n_0)} = \frac{\frac{C_{n_1}}{E_{n_1}}}{1} = \frac{\frac{20}{364}}{1} = \frac{20}{364}$$

$$P\left(n_2|(n_1|n_0)\right) = \frac{P(n_2)}{P(n_1|n_0)} = \frac{\frac{C_{n_2}}{E_{n_2}}}{\frac{20}{364}} = \frac{\frac{8}{364}}{\frac{20}{364}} = \frac{8}{20}$$

$$P\left(n_3|n_2|(n_1|n_0)\right) = \frac{P(n_3)}{P(n_2|(n_1|n_0))} = \frac{\frac{C_{n_3}}{E_{n_3}}}{\frac{8}{20}} = \frac{\frac{8}{364}}{\frac{8}{20}} = \frac{20}{364} \quad (6)$$

Berdasarkan perhitungan diatas diperoleh nilai peluang dari judul yaitu 1 dan nilai peluang dari isis konten yaitu $\frac{20}{364}$. Nilai peluang ini akan digunakan untuk perhitungan bobot dengan menggunakan *Weight Method* yang dapat disederhanakan menjadi:

$$\text{Peluang keterkaitan} = \sum_{t=0}^{n}\left(P(A|B)\right)_i \times k_i \quad (7)$$

dimana *k* merupakan bobot label. Disini akan diberikan bobot label judul dan isi konten sama yaitu 0,5. Sehingga menggunakan persamaan 8 diperoleh:

$$\text{Peluang keterkaitan} = 1 \times 0{,}5 + \frac{20}{364} \times 0{,}5 = 0{,}5275 \quad (8)$$

Sehingga hasil akhir dari nilai peluang keterkaitan halaman web adalah 0,5275. Hasil akhir tersebut akan menjadi acuan penilaian predikat pada sebuah situs web. Predikat itu sendiri terdiri dari "bagus" ($0{,}5 \leq$ nilai peluang keterkaitan ≤ 1) dan "tidak bagus" ($0 \leq$ nilai keterkaitan $< 0{,}5$). Jika halaman web tidak dapat dikunjungi maka predikat yang akan tertulis yaitu "belum diketahui". Predikat ini berfungsi untuk menilai kinerja dari sistem yang sedang diteliti.

E. Perancangan Proses Perbandingan

Pada proses pembandingan, sistem *single thread* dan *multi thread* akan dibandingkan kualitas kinerjanya. Disini peneliti memiliki tiga kriteria yang akan dibandingkan yaitu jumlah URL yang diperoleh, jumlah URL yang bagus, dan lama waktu komputasi. Tiga kriteria tersebut akan dirata-rata dari semua hasil uji coba yang telah dilakukan. Dari rata-rata tersebut akan terlihat manakah sistem yang lebih banyak menghasilkan URL, sistem manakah yang menghasilkan URL bagus lebih banyak, atau sistem manakah yang memiliki tingkat kecepatan komputasi lebih bagus. Proses perbandingan ini akan dilakukan di Microsift Excel untuk mempermudah perhitungan rata-rata dan perbandingannya.

Pada penelitian ini untuk melihat perbandingannya akan dilakukan uji coba sebanyak 10 kali pada setiap kata kunci yang digunakan. Disini akan menggunakan 5 kata kunci dengan 1 kata, 5 kata kunci dengan 2 kata, dan 5 kata kunci dengan 3 kata. Sehingga total dari semuakata kunci yaitu sebanyak 150 kali pengujian pada setiap sistem.

IV. HASIL DAN PEMBAHASAN

Pengujian dilakukan pada sistem *single thread* dan *multi thread* dengan menggunakan 3 skenario yaitu skenario pertama dengan menggunakan kata kunci 1 kata, skenario kedua menggunakan 2 kata, dan skenario ketiga menggunakan 3 kata. Pada setiap skenario akan dilakukan pengujian sebanyak 5 kali dengan kata berbeda dan akan diulang sebanyak 10 kali.

A. Hasil Pengujian Skenario 1

Pada skenario pertama menggunakan kata kunci aljabar, ramadhan, surabaya, diferensial, dan kebaya. Hasil rata-rata dari pengujian skenario 1 pada *single thread* disajikan pada Tabel 1. Sedangkan untuk *multi thread* ditampilkan pada Tabel 2.

B. Hasil Pengujian Skenario 2

Kata kunci yang digunakan pada skenario ketiga ini adalah aljabar linear, persamaan kuadrat, matematika its, buka bersama, dan olahraga basket. Hasil rata-rata dari pengujian skenario 2 pada *single thread* disajikan pada Tabel 3. Sedangkan untuk *multi thread* ditampilkan pada Tabel 4.

C. Hasil Pengujian Skenario 3

Pada skenario ketiga akan digunakan kata kunci aljabar linear elementer, persamaan diferensial biasa, alat tulis kantor, toko perhiasan surabaya, dan tempat wisata probolinggo. Hasil rata-rata dari pengujian skenario 3 pada *single thread* disajikan pada Tabel 5.

D. Analisis Perbandingan Hasil Uji Coba

Berdasarkan hasil uji coba di atas akan dilakukan analisis perbandingan dari kinerja pada sistem *single thread* dan *multi thread*. Pada skenario 1 jika dibandingkan Tabel 1 dan Tabel 2, *multi thread* lebih unggul pada banyaknya URL yang dihasilkan danURL yang bagus yaitu dengan rata-rata jumlah yang dihasilkan 41 URL dan rata-rata URL yang bagus 34 URL. Namun dari sisi kecepatan komputasi dan kualitas URL yang dihasilkan, *single thread* lebih unggul yaitu dengan rata-rata kecepatan komputasi 68304,24 ms dan rata-rata kualitas URL 81,52%.

Pada skenario 2 diperhatikan Tabel 3 dan Tabel 4, *multi thread* lebih unggul dari sisi rata-rata URL yang bagus

sebanyak 8, kecepatan komputasi selama 55581,28 ms, dan kualitas URL yang diperoleh sebesar 51,14%. Sedangkan *single thread* hanya unggul di banyaknya URL yang diperoleh yaitu dengan rata-rata sebanyak 17 dimana hanya lebih 3URL dibanding *multi thread*. Selanjutnya pada skenario 3 dihasilkan Tabel 5 dimana hasil yang diperoleh kurang lebih sama dengan skenario 2. *Single thread* lebih banyak URLyang dihasilkan namun kualitas dan kecepatannya lebih unggul *multi thread* dengan rata-rata kualitas URLnya sebesar 51,59% dan rata-rata kecepatan komputasinya selama 39217,32 ms.

Pada Gambar 8 ditampilkan diagram rata-rata hasil uji coba secara keseluruhan berdasarkan jumlah URL yang diperoleh. Berdasarkan diagram tersebut terlihat bahwa URL yang dihasilkan antara *single thread* dan *multi thread* tidak jauh berbeda kecuali pada kata aljabar. Terbukti bahwa *multi thread* mampu mengambil URL lebih banyak jika kata yang digunakan cukup sering dicari oleh pengguna. Sedangkan *single thread* baik itu kata yang sering dicari atau tidak, *single thread* hanya mampu mengambil URL sedikit.

Terlihat dari banyaknya URL yang dihasilkan *multi thread*, ternyata *multi thread* tidak hanya asal mengambil URL yang ada. URL yang dihasilkan juga banyak yang memiliki predikat bagus menurut perhitungan menggunakan persamaan 6. Hal tersebut dapat dilihat pada Gambar 9 yang hasilnya kurang lebih sama dengan Gambar 8. Oleh karena itu diperolehlah diagram perbandingan kualitas dari URL yang didapatkan pada Gambar 10. Dapat terlihat bahwa *multi thread* lebih dominan menghasilkan URL yang berkualitas dibandingkan *single thread*. Selain itu dapat terlihat bahwa hasil yang diperoleh dari skenario 1 sampai 3 dapat disimpulkan bahwa semakin detail kata yangdigunakan, maka semakin sedikit URL yangakan diperoleh.

Selanjutnya akan dibandingkan kecepatan waktu komputasi kedua sistem. Terlihat pada diagram waktu komputasi di Gambar 11 bahwa kecepatan komputasi dalam pengambilan URL berbanding lurus dengan banyaknya URL yang didapatkan jika dilihat sekilas. Namun jika diperhatikan pada beberapa titik yang hampir bersamaan antara *single thread* dan *multi thread*, semua terlihat bahwa *multi thread* selalu memiliki kecepatan yang lebih tinggi walaupun URL yang dihasilkan dominan lebih banyak.

E. Analisis Hasil

Berdasarkan hasil dari percobaan skenario 1 hingga 3 terdapat hasil yang kurang lazim pada hasil beberapa kata kunci. Hasil tersebut diantaranya pada kata kunci "Matematika ITS", "Toko Perhiasan Surabaya", dan "Tempat Wisata Probolinggo". Pada ketiga kata kunci ini dimana URL yang bagus terlihat pada Tabel 3 hingga Tabel 5 sangat sedikit dibandingkan dengan jumlah URL yang diperoleh. Oleh karena itu, dilakukan pengecekan kembali pada hasil URL yang ada. Pada pengecekan tersebut ditemukannya URL yang sebenarnya bagus atau relevan dengan kata kunci, namun pada perhitungan keterkaitan diperoleh predikat "tidak bagus".

Terlihat pada Gambar 12 dengan kata kunci "Matematika ITS" halaman tersebut terhitung "tidak bagus" dengan menggunakan perhitungan *classification and weight method*. Sedangkan dari penilaian peneliti dengan membaca isi halaman web tersebut, halaman tersebut dapat dianggap bagus atau relevan dengan kata kunci yang digunakan. Begitu pula dengan kata kunci "Toko Perhiasan Surabaya" dan "Tempat Wisata Probolinggo". Terlihat pada Gambar 13 dan Gambar 14 dapat dikatakan bagus atau relevan dengan kata kunci jika dibaca langsung isi halaman web tersebut.

Berdasarkan pengamatan tersebut terdapat ketidaksesuaian antara hasil perhitungan dengan pengamatan langsung. Hal ini dikarenakan kata pada judul dan isi kontennya tidak sama seperti kata kunci yang dimasukkan. Sehingga dari kasus ini dapat disimpulkan perhitungan *classification* dan *weight method* tidak dapat diterapkan pada kata kunci yang umum. Dimana yang dimaksud kata umum tersebut yaitu kata yang dapat diketahui atau dimengerti oleh masyarakat luas.

V. KESIMPULAN/RINGKASAN

Berdasarkan hasil pengujian dan pembahasan yang telah didapatkan, diperoleh kesimpulan dari penelitian ini sebagai berikut: (1) Telah berhasil dilakukan implementasi *single thread* dan *multi thread* pada *web crawling*. (2) Berdasarkan analisis kinerja antara *single thread* dan *multi thread* pada *web crawling* diperoleh bahwa metode *multi thread* lebih baik daripada *single thread* dimana kualitas URL yang dihasilkan lebih besar yaitu 62,33%, sedangkan pada *single thread* untuk kualitas URL yang dihasilkan hanya sebesar 54,52%. Selain itu berdasarkan kecepatan komputasinya, *multi thread* juga lebih unggul dengan rata-rata waktu komputasi selama 59,243 s, sedangkan pada *single thread* selama 63,445 s. (3) Perhitungan *classification* dan *weight method* tidak dapat diterapkan pada kata kunci yang umum.

DAFTAR PUSTAKA

[1] A. K. Sharma, V. Shrivastava, and H. Singh, “Experimental performance analysis of web crawlers using single and multi-threaded web crawling and indexing algorithm for the application of smart web contents,” *Mater Today Proc*, vol. 37, pp. 1403–1408, 2021, doi: 10.1016/j.matpr.2020.06.596.

[2] A. S. Putra, “Implementasi Focused Web Crawling untuk Akuisisi Data pada Situs Web,” Departemen Matematika, Institut Teknologi Sepuluh Nopember, Surabaya, 2019.

[3] K. Vayadande, R. Shaikh, T. Narnaware, S. Rothe, N. Bhavar, and S. Deshmukh, “Designing web crawler based on multi-threaded approach for authentication of web links on internet,” in *2022 6th International Conference on Electronics, Communication and Aerospace Technology*, IEEE, Dec. 2022, pp. 1469–1473. doi: 10.1109/ICECA55336.2022.10009614.