

Penerapan Web Scraping Sebagai Media Pencarian dan Menyimpan Artikel Ilmiah Secara Otomatis Berdasarkan Keyword

Veronica Ambassador Flores¹, Putri Agung Permatasari², Lie Jasa³

[Submission: 28-05-2020, Accepted: 17-12-2020]

Abstract— A scientific article is a reference that usually used by a student or researcher to make a study. Scientific articles can be searched freely on the Internet, but the search process is often time-consuming due with the large number of scientific articles on the Internet. This search can be done automatically by utilizing the Web Scraping Technique as a data collection technique on a website. Depth First Search (DFS) method is a search method that can be used to do web Scraping on the Internet. This system works by processing the Keywords about the scientific article topic given by the user to be searched for. Then the system will send a request to the server to look for links related with these Keywords. After the search results found, the system will automatically download and save the article with .pdf format to user's computer. The results of this study indicate from 134 articles were successfully downloaded and this system can work very well to search the scientific articles with a precision value of 0.87 and a recall value of 0.99.

Keywords —Scientific Article, Web Scraping, Depth First Search

Intisari—Artikel ilmiah adalah sebuah referensi yang biasanya digunakan oleh mahasiswa atau peneliti untuk membuat sebuah penelitian. Artikel Ilmiah dapat dicari melalui Internet secara bebas, namun proses pencarian ini seringkali memakan waktu dikarenakan banyaknya artikel ilmiah yang ada di Internet. Pencarian ini dapat dilakukan secara otomatis dengan memanfaatkan Teknik *Web Scraping* sebagai teknik pengambilan data pada suatu situs *web*. Metode *Depth First Search* (DFS) adalah metode pencarian yang dapat dimanfaatkan untuk melakukan *Web Scraping* di Internet. Sistem ini bekerja dengan mengolah *Keyword* mengenai topik artikel ilmiah yang diberikan oleh pengguna. Kemudian sistem akan mengirimkan *request* kepada *server* untuk mencari tautan yang berhubungan dengan *Keyword* tersebut. Setelah ditemukan, sistem secara otomatis akan mengunduh artikel yang berformat .pdf ke komputer pengguna. Hasil uji coba dari 134 artikel yang berhasil diunduh menunjukkan bahwa sistem ini dapat bekerja dengan sangat baik untuk melakukan pencarian artikel ilmiah dengan nilai *precision* sebesar 0.87 dan *recall* sebesar 0.99.

^{1, 2} Mahasiswa, Magister Teknik Elektro Universitas Udayana Gedung Pascasarjana Universitas Udayana, Jl. PB Sudirman Denpasar-Bali 80234 INDONESIA (telp: 0361-555225; e-mail: veronicaambassador@gmail.com, agungputri.p@gmail.com)

³ Staff Pengajar, Magister Teknik Elektro Universitas Udayana Gedung Pascasarjana Universitas Udayana, Jl. PB Sudirman Denpasar-Bali 80234 INDONESIA (telp: 0361-555225; e-mail: liejasa@unud.ac.id)

Kata Kunci—Artikel Ilmiah, *Web Scraping*, *Depth First Search*

I. PENDAHULUAN

Seorang mahasiswa atau seorang peneliti seringkali diwajibkan untuk melakukan sebuah penelitian yang berkaitan dengan studi mereka. Penelitian dapat dilakukan jika ditunjang oleh berbagai referensi yang berupa penelitian-penelitian sebelumnya atau berupa teori-teori yang mendukung tujuan penelitian tersebut. Referensi berupa artikel ilmiah dapat ditemukan dengan sangat mudah melalui Internet, karena Internet telah menjadi sesuatu yang penting dalam perkembangan dunia komunikasi dan pendidikan [1]. Menurut Survei, dari total 260 mahasiswa dan pekerja akademisi, sebanyak 152 orang mencari artikel ilmiah langsung pada kolom pencarian Google [2]. Sebuah penelitian memperkirakan bahwa setidaknya 114 juta dokumen ilmiah berbahasa Inggris tersedia dan dapat diakses di Internet secara bebas [3]. Begitu banyak artikel ilmiah yang tersedia di Internet, membuat pencarian artikel ilmiah di mesin pencari akan memakan waktu yang sangat lama untuk seseorang dalam menemukan artikel yang tepat [4]. Saat mahasiswa atau peneliti sudah menemukan artikel yang dicari, terkadang artikel ini masih tidak tersedia, hal ini dikarenakan terkadang status dari artikel yang ada di Internet sudah terblokir atau tidak dapat dimuat, atau membutuhkan pembayaran lebih lanjut untuk pengaksesannya [5].

Artikel ilmiah yang ada di Internet dapat secara otomatis didapatkan dengan menggunakan Teknik *Web Scraping*. *Web Scraping* merupakan suatu teknik yang digunakan untuk mendapatkan suatu data atau informasi pada suatu *website* secara otomatis. Informasi tersebut dapat berupa, teks, tautan, video, audio ataupun dokumen.

Penelitian [6] memanfaatkan Teknik *Web Scraping* untuk melakukan rekapitulasi atau perhitungan artikel ilmiah pada situs *Google Scholar* berdasarkan nama peneliti di sebuah lembaga. Rekapitulasi ini berfungsi untuk mengetahui kinerja penelitian mereka secara kolektif. Hasil rekapitulasi disimpan kedalam format *.xlsx atau *.pdf lengkap dengan daftar peneliti, daftar judul artikel yang pernah ditulis, daftar kutipan, dan daftar afiliasi. Penelitian ini menyimpulkan bahwa Teknik *Web Scraping* dapat digunakan untuk mengambil dan mengumpulkan data artikel ilmiah dari situs *Google Scholar*.

Penelitian [7] membangun sebuah sistem yang dapat memberikan perbandingan sebuah produk dengan memanfaatkan Teknik *Web Scraping* sebagai media untuk membandingkan produk yang ada pada situs *web ecommerce*. Pengguna hanya perlu memasukkan nama barang yang dicari, kemudian Teknik *Web Scraping* akan bekerja dengan



mengambil data produk berupa nama barang, deskripsi produk, harga, ulasan, *rating*, dan jumlah produk yang terjual melalui *website* Bukalapak, JD.id, dan Eleveni.

Penelitian [8] menerapkan Teknik *Web Scraping* untuk melakukan pencarian artikel ilmiah pada situs Portal Garuda, Google Scholar dan Indonesian Scientific Journal Database (ISJD). Penelitian ini membangun sebuah sistem menggunakan Bahasa Pemrograman PHP. Cara kerja sistem ini yaitu, pengguna hanya perlu memasukkan *keyword* dari artikel yang dicari kemudian menentukan situs publikasi jurnalnya. Daftar artikel akan ditampilkan lengkap dengan *link* untuk mengunduh artikel tersebut.

Penelitian [9] menggunakan Metode *Depth First Search* untuk mencari suatu dokumen di memori utama komputer dan juga pada media *external* seperti *disk* berdasarkan *keyword* yang dimasukkan oleh pengguna. Kelebihan Metode *Depth First Search* yaitu pencarian dengan metode ini membutuhkan waktu yang cepat dan lebih efisien untuk masuk ke ruang-ruang atau folder pencarian, hal ini dikarenakan Metode *Depth First Search* tidak akan mengeksekusi seluruh simpul yang ada pada suatu level.

Penelitian [10] menerapkan Metode *Depth First Search* untuk melakukan *Web Scraping* pada situs berita Liputan6, Detik.com, dan CNN Indonesia. Penelitian ini bertujuan untuk mencari tahu berapa banyak dari situs-situs ini yang mengeluarkan berita dengan *keyword* "PENDIDIKAN". Hasil menunjukkan bahwa situs Detik.com adalah situs terbanyak yang mempublikasikan berita *online* terkait "PENDIDIKAN". Penelitian ini menyimpulkan bahwa hasil dari *Web Scraping* akan sangat ditentukan berdasarkan *keyword* yang lebih spesifik. Metode *Depth First Search* juga mempengaruhi hasil *Web Scraping* secara signifikan, karena metode ini dapat memproses *keyword* dan menampilkan berita yang mirip bersama dengan URL berita tersebut.

Metode pencarian pada Teknik *Web Crawling* dapat menggunakan Algoritma *Depth First Search* (DFS) dan *Breadth First Search* (BFS). *Web Crawling* adalah sebuah teknik yang berguna untuk menjelajahi, membaca dan mengambil data pada suatu situs *web*. Penelitian [11] melakukan perbandingan terhadap dua algoritma tersebut. Alur sistem pada penelitian ini yaitu, pengguna memasukkan URL, kemudian proses *crawling* akan dijalankan secara bersamaan menggunakan Metode *Breadth First Search* dan *Depth First Search*, kemudian URL yang ditemukan dalam proses BFS dan DFS akan diekstrak dan dikalkulasikan waktu pemrosesannya. Data yang diujikan pada penelitian ini adalah berdasarkan jumlah *Branching Factors* (jumlah anak di setiap node). Hasil menunjukkan bahwa pada jumlah *Branching Factor* 250, Metode BFS memerlukan waktu pemrosesan selama 6 menit 21 detik, sedangkan Metode DFS memerlukan waktu 3 menit 6 detik. Penelitian ini menunjukkan bahwa Metode DFS dapat melakukan proses *crawling* dengan lebih cepat.

Berdasarkan penelitian sebelumnya, maka penelitian ini akan menggunakan Teknik *Web Scraping* untuk mencari artikel ilmiah pada pencarian Google (semua situs) secara otomatis menggunakan Metode *Depth First Search*. Metode *Depth First Search* digunakan sebagai metode dalam melakukan pencarian tautan artikel ilmiah yang berupa pdf pada suatu situs *web*. Tiap situs *web* yang didapatkan pada

kolom pencarian Google akan ditelusuri tautan per tautan sampai mendapatkan artikel ilmiah yang dicari.

Penelitian ini bertujuan untuk membantu mahasiswa atau para peneliti untuk mencari artikel mengenai topik tertentu tanpa harus masuk kedalam situs artikel tersebut, dan menampilkan daftar artikel ilmiah pada situs artikel yang bersifat *open-access* secara otomatis, dan mengunduh semua artikel ilmiah yang ditemukan kedalam lokal komputer.

II. LANDASAN TEORI

Penelitian ini dilandasi oleh beberapa teori pendukung yang berasal dari berbagai sumber.

A. Artikel Ilmiah

Artikel ilmiah adalah laporan tercetak yang menggambarkan hasil dari sebuah studi [12]. Ciri-ciri dari sebuah artikel yaitu mencakup bidang studi tertentu, diterbitkan secara teratur (mingguan, bulanan, triwulanan), dan berisi artikel, ulasan atau konten editorial. Artikel biasanya dijadikan sebagai sebuah referensi dalam sebuah penelitian [13]. Terdapat empat jenis model pada sebuah penulisan artikel. Sebagian besar makalah menyajikan alternatif pendekatan baru dengan masalah yang sudah ada atau sudah pernah dibahas. Artikel lainnya menyajikan analisis mengenai teori baru, dan ada juga makalah survei yang menyajikan tinjauan komprehensif dari bidang studi yang diberikan. Artikel yang terakhir yaitu artikel yang menyatakan mengenai sebuah masalah baru [14].

B. Web Scraping

Web Scraping atau yang dikenal sebagai ekstraksi *web* adalah teknik untuk mengekstraksi data dari *World Wide Web* (WWW) dan menyimpannya ke *file* sistem atau basis data untuk dijadikan analisis data. *Web Scraping* dapat dilakukan baik secara manual oleh seorang pengguna atau secara otomatis oleh *bot* atau *crawler web*. Proses *Web Scraping* dari Internet dapat dibagi menjadi dua langkah berurutan, yaitu mengakuisisi sumber daya *web* dan kemudian mengekstraksi informasi yang diinginkan dari data yang diperoleh. Secara khusus, program *Web Scraping* dimulai dengan meminta HTTP untuk memperoleh sumber daya dari yang ditargetkan oleh situs *web*. Permintaan ini dapat diformat kedalam URL yang berisi permintaan GET atau HTTP yang berisi POST. Setelah permintaan berhasil diterima dan diproses oleh situs *web* yang ditargetkan, sumber daya yang diminta akan diambil dari situs *web* dan kemudian dikirim kembali ke program *Web Scraping*. Sumber daya ini bisa dalam berbagai format, seperti halaman *web* yang dibangun dengan HTML, XML atau JSON, atau data multimedia seperti gambar, audio, atau video. Terdapat dua modul penting dari *Web Scraping* - modul untuk menulis permintaan HTTP, seperti *Urllib2* atau *selenium* dan satu lagi untuk *parsing* dan mengekstraksi informasi dari kode HTML mentah, seperti *BeautifulSoup* atau *Pyquery* [15].

C. Legalitas Web Scraping

Pada akhir 2019, Pengadilan AS menolak banding dari permintaan LinkedIn kepada HiQ, sebuah perusahaan analitik, dari penggoresan data LinkedIn. Keputusan itu adalah momen bersejarah dalam era privasi dan regulasi data yang

menunjukkan bahwa data apa pun yang tersedia untuk umum dan tidak dilindungi hak cipta adalah adil untuk para *web scraping*.

Namun, keputusan tersebut tidak memberikan kebebasan bagi *web scraping* untuk menggunakan data yang diperoleh untuk tujuan komersial. Misalnya, *web scraping* akan diizinkan untuk mencari judul video di Youtube, tetapi tidak dapat memposting ulang video Youtube di situsya sendiri, karena video tersebut memiliki hak cipta. Youtube sendiri merupakan aplikasi berbagi video dengan jumlah pengguna aktif yang cukup besar yang mengijikan pengguna untuk saling berinteraksi di platform tersebut [16] [17].

Keputusan ini juga tidak memberikan kebebasan kepada *web scraping* untuk mendapatkan data dari situs yang membutuhkan otentikasi. Misalnya, *web scraping* masuk ke Facebook dan mengunduh data pengguna adalah ilegal.

Berikut beberapa pedoman umum yang harus dipatuhi oleh para *web scrapper*

- Tidak ada kata sandi atau hambatan yang dilanggar untuk mengambil konten.
- Konten yang diambil tidak boleh dilindungi hak cipta.
- Jika dilindungi hak cipta, konten yang diambil harus mematuhi standar yang adil.
- Tindakan *scraping* tidak boleh membebani layanan situs *webnya*.
- *Scraping* tidak boleh melanggar ketentuan penggunaan situs yang sedang diambil.
- *Scraping* tidak boleh mengumpulkan informasi pengguna yang sensitif.

D. Depth First Search

Dalam *Depth First Search* (DFS), ekspansi dimulai dari simpul paling awal dan mengeksplorasinya ke simpul kiri yang paling dalam kemudian naik ke simpul kanan sampai mencapai simpul tujuannya. Metode ini juga disebut metode berbasis tepi dan bekerja dengan mode rekursif di mana simpul dieksplorasi di sepanjang tepinya. Algoritma ini dapat diimplementasikan menggunakan prinsip *Last In First Out* (LIFO) [18] [19].

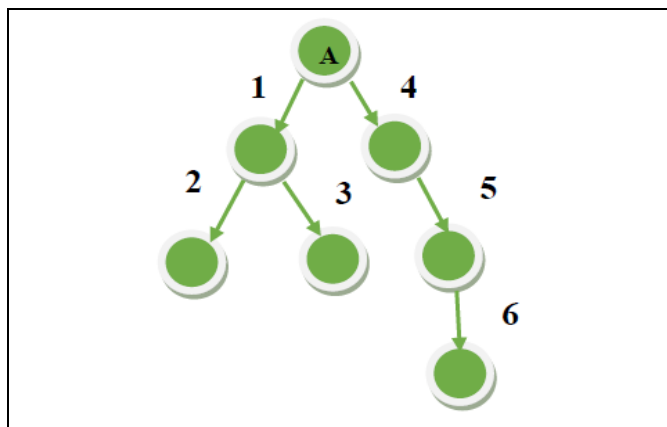
Metode *Depth First Search* memiliki beberapa kelebihan [20], diantaranya yaitu :

- Metode ini menghabiskan lebih sedikit ruang memori jika dibandingkan dengan Metode *Breadth First Search*, hal ini dikarenakan pada *Breadth First Search* semua node yang ada dalam satu pohon.
- Jika target yang dicari berada di posisi level yang paling kiri, maka target akan dapat ditemukan dengan cepat.

Selain kelebihan, Metode *Depth First Search* juga memiliki beberapa kekurangan [20], yaitu:

- Jika struktur pohon yang akan ditelusuri memiliki level yang dalam (kedalam yang tak terhingga), maka terdapat kemungkinan jika target tidak akan ditemukan.

- Jika ternyata terdapat lebih dari satu target di level yang berbeda, maka terdapat kemungkinan jika target yang lainnya tidak akan ditemukan



Gambar 1: Pola *Depth First Search* [21]

Gambar 1 menunjukkan bagaimana menemukan tujuan dengan menggunakan algoritma DFS. Pencarian dimulai dari simpul awal "A" dan kemudian bergerak ke node 1 (paling kiri), kemudian menelusuri cabang yang ada dibawahnya yaitu Node 2. Jika tujuannya belum ditemukan pada Node 2, maka pencarian akan berpindah ke node sebelah kanan yaitu Node 3, dan pencarian akan naik ke atas ke cabang yang ada disebelah kanan yaitu Node 4 dan begitu seterusnya.

E. Breadth First Search

Breadth First Search adalah algoritma pencarian yang dimulai dalam satu deret level baru kemudian di lanjutkan ke satu deret level selanjutnya di mulai dari node yang paling kanan.

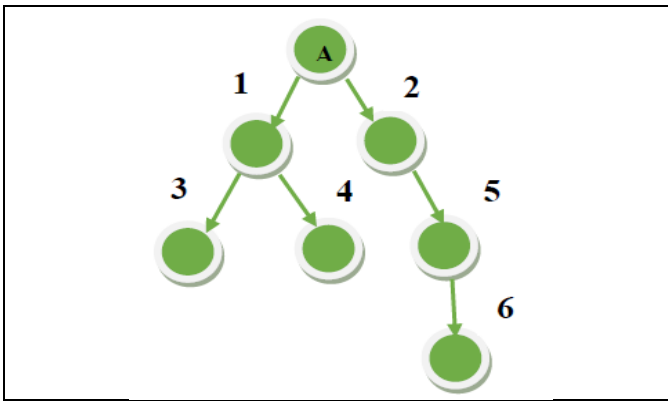
Kelebihan yang dimiliki oleh Metode *Breadth First Search* adalah sebagai berikut [20]:

- Tidak akan bertemu dengan jalan buntu
- Jika target hanya berada di satu posisi, maka metode ini akan menemukannya, namun jika target yang dicari berada dilebih dari satu posisi maka target minimum akan ditemukan

Selain kelebihan, Metode *Breadth First Search* juga memiliki beberapa kekurangan [20], yaitu:

- Metode *Breadth First Search* menghabiskan lebih banyak ruang memori, hal ini dikarenakan metode ini akan menyimpan semua node yang ada dalam satu pohon.





Gambar 2: Pola Breadth First Search [21]

Gambar 2 menunjukkan bagaimana menemukan tujuan dengan menggunakan algoritma BFS. Pencarian dimulai dengan node awal A dan melakukan pencarian ke satu deret node yang ada di bawahnya yaitu level 1 dan level 2, kemudian di lanjutkan ke satu deret level dibawah Node 1 dan seterusnya.

F. Pengujian Akurasi

Pengujian akurasi berfungsi untuk mengetahui seberapa akurat sebuah aplikasi dalam mencari dan menemukan target yang diinginkan. Metode pengujian yang dapat digunakan untuk mengukur tingkat akurasi dan efektifitas metode yang dipakai adalah *Precision* dan *Recall* [22]. *Precision* merupakan tingkat propabilitas yang menunjukkan kerelevanan dari hasil dokumen dengan seluruh dokumen yang berhasil terambil oleh aplikasi [23]. Cara menghitung nilai *Precision* ditunjukkan pada Persamaan 1.

$$Precision = \frac{(Relevant\ document\ retrieved)}{(Retrieved\ document)} \quad (1)$$

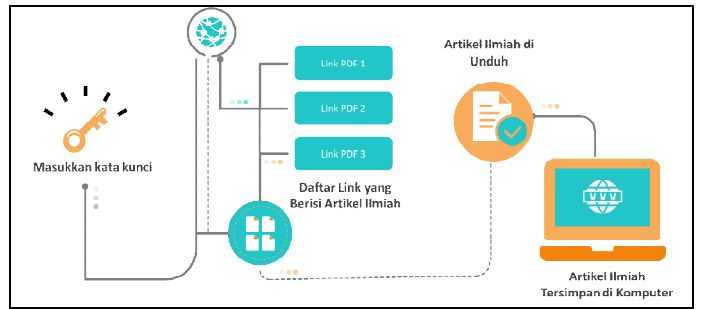
Recall merupakan perbandingan dari dokumen relevan yang dipilih terhadap seluruh dokumen relevan yang berhasil terambil [23]. Perhitungan *Recall* ditunjukkan pada Persamaan 2.

$$Recall = \frac{(Relevant\ document\ retrieved)}{(Relevant\ document)} \quad (2)$$

Proses pengujian tingkat akurasi pada penelirfrian ini akan menggunakan 2 metode, yaitu metode *Precision* yang berfungsi untuk mengetahui performa dari aplikasi yang telah dibangun dengan membandingkan jumlah dokumen valid yang berhasil diambil dengan seluruh jumlah dokumen yang berhasil di ambil oleh sistem. Sedangkan metode *Recall* berfungsi untuk mengetahui tingkat akurasi dari alikasi yang dibangun berdasarkan jumlah dokumen valid yang berhasil diambil dibandingkan dengan jumlah dokumen valid yang seharusnya diambil [22].

III. METODE PENELITIAN

Metodologi yang digunakan dalam perancangan dan implementasi dari *web scraping* ini antara lain:



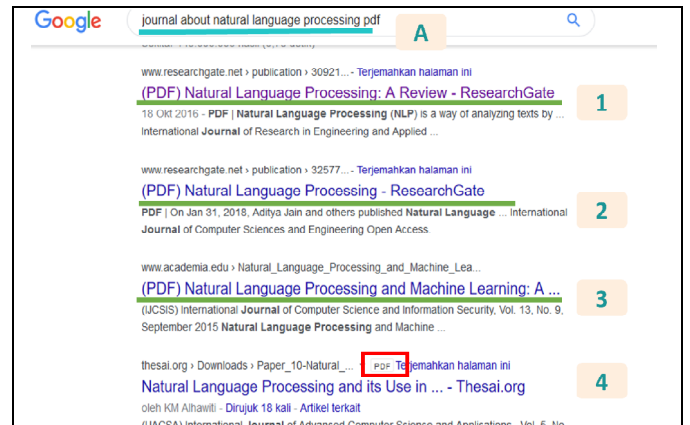
Gambar 3: Gambaran Umum

Berikut alur kerja dari “Penerapan *Web Scraping* Sebagai Media Pencarian dan Menyimpan Artikel Ilmiah Secara Otomatis Berdasarkan *Keyword*”

- Masukkan kata kunci dari artikel yang dicari
- Sistem melakukan *request URL* pada Google berdasarkan kata kunci yang diberikan.
- *Request* diproses oleh *server* Google.
- Hasil dari *request URL* adalah beberapa tautan pencarian pada halaman Google
- Tautan ini kemudian ditelusuri satu persatu menggunakan Metode *Depth First Search*
- Saat tautan yang berekstensi pdf, maka *file* tersebut akan diunduh dan disimpan di lokal komputer.
- Setelah itu pencarian akan dilanjutkan ke node yang baru.

A. Penerapan *Depth First Search*

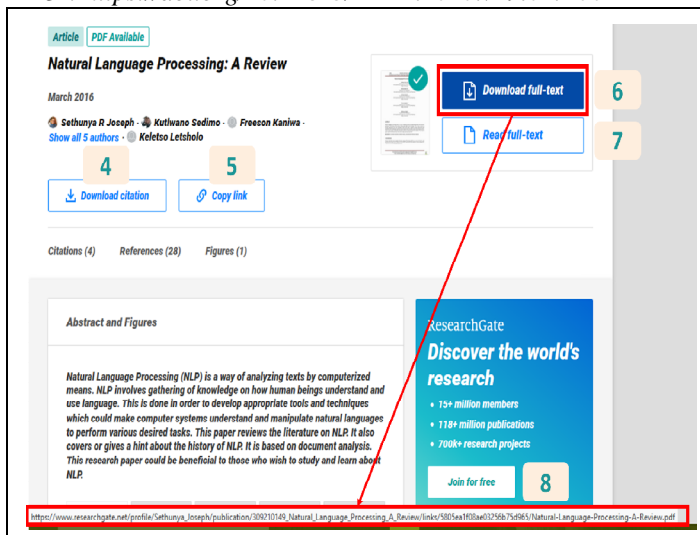
Ilustrasi dari penerapan Metode *Depth First Search* pada penelitian ini dapat dilihat pada Gambar 4.



Gambar 4: Level 1 dan 2 *Depth First Search* pada Pencarian Google

Ilustrasi pada Gambar 4 adalah permodelan dari Metode *Depth First Search* pada Gambar 1. Kata kunci yang dimasukkan oleh pengguna diibaratkan sebagai node atau level paling atas, dan tautan dari hasil *request* terhadap Google adalah Node 1, 2, 3, 4 dan seterusnya.

Selanjutnya jika pada Node 1, tautan yang didapatkan sudah berekstensi pdf, maka pencarian akan selesai disana dan akan dilanjutkan ke Node 2. Namun karena pada Node 1, tautan tersebut tidak berekstensikan pdf, maka pencarian akan dilanjutkan ke level yang ada di bawah node 1.



Gambar 5: Level 3 *Depth First Search* pada Pencarian Google

Pada penelitian ini, metode *Depth First Search* akan diterapkan hanya sampai level ke 3, agar hasil yang didapat masih relevan dan tidak menyimpang dari topik atau kata kunci yang dimasukkan.

Pada level 3 yang berada di bawah Node 1, pencarian terhadap tautan akan dilakukan pada Node 4, 5, 6, 7, dan 8. Tautan dengan ekstensi pdf yang ada di node-node tersebut akan diunduh secara otomatis dan disimpan di komputer. Jika sudah selesai (walaupun tautan pdf ditemukan ataupun tidak ditemukan pada level ini), pencarian akan dilanjutkan ke Node 2, dan begitu seterusnya sampai *page* pada halaman pencarian habis.

B. Implementasi Sistem

Aplikasi ini dibangun menggunakan aplikasi Python dengan memanfaatkan *library* pendukung, yaitu :

- *Requests* merupakan *library* yang digunakan untuk mengirimkan *request* HTTP kepada *server*. *Library* ini digunakan untuk mengirimkan tautan dari tempat publikasi artikel ilmiah yang bersifat *open source*
- *BeautifulSoup* merupakan *library* yang berfungsi untuk mem-*parsing* isi halaman HTML. *Library* ini digunakan untuk mem-*parsing* konten tautan dalam tag `<a>`
- *Wget* merupakan modul pada Python yang dapat digunakan untuk mengunduh sebuah *file* ke komputer tanpa harus membukanya.

IV. PEMBAHASAN DAN ANALISIS SISTEM

Penelitian “Penerapan *Web Scraping* Sebagai Media Pencarian dan Menyimpan Artikel Ilmiah Secara Otomatis Berdasarkan *Keyword*” diharapkan dapat memberikan pengguna daftar artikel ilmiah yang dicari dengan hasil yang relevan.



Gambar 6: Implementasi Sistem

Gambar 6 menampilkan hasil implementasi sistem pada aplikasi Python. Pengguna diharuskan memasukkan kata kunci terlebih dahulu, kemudian barulah sistem mencari tautan artikel yang berhubungan dengan kata kunci tersebut. Secara otomatis tautan akan di unduh dan disimpan di komputer lokal milik pengguna.

TABEL 6
 HASIL UJI COBA

Kata Kunci	Total	True	False	Relevan
Computer Science	19	18	1	18
Computer Security	22	19	3	19
Fingerprint	14	12	2	12
Forward Chaining	12	12	0	12
Game Education	25	14	11	13
MD5	10	10	0	10
Naive Bayes	15	14	1	14
Sistem Pakar	17	17	0	17
TOTAL	134	116	18	115

Hasil uji coba dibagi menjadi tiga kategori, yaitu *True*, *False*, dan *Relevan*. *True* adalah kondisi dimana *file* dapat diunduh dan disimpan di komputer. *False* adalah kondisi dimana *file* dapat diunduh dan disimpan di komputer namun *file* tersebut tidak dapat dibuka karena *file* tersebut *error* atau *corrupt* atau tidak terunduh dengan baik. *Relevan* merupakan kondisi dimana *file* yang diunduh dan disimpan sesuai dengan kata kunci yang dicari oleh pengguna. Penyebab artikel yang diunduh tidak relevan adalah karena *keyword* yang dimasukkan tercantum pada halaman artikel tersebut, namun bukan sebagai *main topic*, melainkan hanya sebagai sitasi, *footer* halaman, ataupun karena *keyword* tersebut tercantum sebagai opsi untuk menuju ke halaman yang lainnya.

Berdasarkan hasil uji coba, sistem “Penerapan *Web Scraping* Sebagai Media Pencarian dan Menyimpan Artikel Ilmiah Secara Otomatis Berdasarkan *Keyword*” yang



dibangun menggunakan aplikasi Python dapat mengunduh dan menyimpan *file* secara otomatis dengan nilai akurasi *precision* mencapai 0.87

$$precision = \frac{(Relevant\ document\ retrieved)}{(Retrieved\ document)} \quad (3)$$

$$precision = \frac{116}{134}$$

$$precision = 0.87$$

Tingkat relevansi atau *recall* dari aplikasi ini yaitu sebesar 0.99, hal ini dikarenakan proses pencarian hanya dilakukan sampai level ketiga, sehingga hasil yang didapatkan tidak akan jauh dari kata kunci yang diberikan oleh pengguna.

$$recall = \frac{(Relevant\ document\ retrieved)}{(Relevant\ document)} \quad (4)$$

$$recall = \frac{115}{116}$$

$$recall = 0.99$$

Dari perhitungan Persamaan 3 dan 4, dapat dilihat bahwa tingkat akurasi dari aplikasi *Web Scrapping* ini dapat bekerja dengan baik dengan nilai *precision* sebesar 0.87 (Persamaan 3) dan nilai *recall* sebesar 0.99 (Persamaan 4).

Dengan tingkat *precision* dan *recall* tersebut, maka dapat disimpulkan bahwa sistem "Penerapan *Web Scrapping* Sebagai Media Pencarian dan Menyimpan Artikel Ilmiah Secara Otomatis Berdasarkan *Keyword*" dapat mengenali *keyword* yang diberikan oleh pengguna dan mencari artikel yang relevan dengan *keyword* tersebut, dan secara otomatis langsung menyimpan artikel-artikel tersebut ke dalam penyimpanan lokal komputer.

V. KESIMPULAN

Berdasarkan penelitian dari "Penerapan *Web Scrapping* Sebagai Media Pencarian dan Menyimpan Artikel Ilmiah Secara Otomatis Berdasarkan *Keyword*" dapat disimpulkan bahwa *Web Scrapping* dengan menggunakan algoritma pencarian *Depth First Search* sampai di tingkat atau level ke-3 dapat menghasilkan hasil pencarian artikel ilmiah yang relevan dengan nilai *precision* yang mencapai 0.87 dan nilai *recall* yang mencapai 0.99. Kelemahan pada sistem ini yaitu belum dapat mengoptimalkan metode pengunduhan artikel, sehingga terdapat beberapa dokumen hasil pengunduhan yang tidak dapat dibuka dikarenakan dokumen tersebut belum terunduh dengan baik. Diharapkan kedepannya, penelitian ini dapat dikembangkan menggunakan bahasa pemrograman PHP agar sistem ini memiliki interface yang dapat digunakan secara luas. Penelitian ini juga dapat dikembangkan di platform sosial media berbasis *web* seperti Twitter dan Facebook.

REFERENSI

- [1] N. L. Ratmiasih, M. Sudarma and N. Gunantara, "Penerapan Text Mining dalam Spam Filtering untuk Aplikasi Chat," *Majalah Ilmiah Teknologi Elektro*, vol. 16, no. 3, 2017.
- [2] K. E. Hubbard and S. D. Dunbar, "Perceptions of Scientific Research Literature and Strategies for Reading Papers Depend on Academic Career Stage," *PLoS ONE*, vol. 12, no. 2, 2017.
- [3] M. Khabsa and C. L. Giles, "The Number of Scholarly Documents on the Public Web," *PLoS ONE*, vol. 9, no. 5, 2014.
- [4] R. E. P. Nasution, "Rintangan Dalam Menemukan Referensi Skripsi," *White Coat Hunter*, 2017. [Online]. Available: <https://whitecoathunter.com/menemukan-referensi-skripsi/>. [Accessed 24 4 2020].
- [5] D. S. Chawla, "Need a paper? Get a plug-in," *Nature*, 2017. [Online]. Available: <https://www.nature.com/articles/d41586-017-05922-9>. [Accessed 24 4 2020].
- [6] A. Rahmatulloh and R. Gunawan, "Web Scraping with HTML DOM Method for Data Collection of Scientific Articles from Google Scholar," *Indonesian Journal of Information Systems (IJIS)*, vol. 2, 2020.
- [7] D. D. Ayani, H. S. Pratiwi and H. Muhandi, "Implementasi Web Scraping untuk Pengambilan Data pada Situs Marketplace," *Jurnal Sistem dan Teknologi Informasi*, vol. 7, no. 4, 2019.
- [8] A. Josi, L. A. Abdillah and Suryayusra, "Penerapan Teknik Web Scraping pada Mesin Pencari Artikel Ilmiah," *Jurnal Sistem Informasi*, vol. 5, no. 2, 2014.
- [9] S. Lailiyah, A. Yusnita and T. A. Panotogomo, "Penerapan Algoritma Depth First Search Pada Sistem Pencarian Dokumen," in *SNITT-Politeknik Negeri Balikpapan*, Balikpapan, 2017.
- [10] P. S. Endah Ratna Arumi, "Exploiting Web Scraping for Education News Analysis Using Depth-First Search Algorithm," *JOIN (Jurnal Online Informatika)*, vol. 5, no. 1, 2020.
- [11] A. E. Wibowo and K. M. Lhaksmana, "Perbandingan Performansi Terhadap Algoritma Breadth First Search (BFS) & Depth First Search (DFS) Pada Web Crawler," *e-Proceeding of Engineering*, vol. 6, no. 2, 2019.
- [12] B. B. M. N. Y. v. Yayımları, "How to Write and Publish a Scientific Article," *Journal of Urological Surgery*, vol. 5, no. 2, 2018.
- [13] T. U. o. Queensland, "Library," [Online]. Available: <https://web.library.uq.edu.au/files/913/What%20is%20a%20Journal.pdf>. [Accessed 24 4 2020].
- [14] S. R. N. Reis and A. I. Reis, "How to Write Your First Scientific Paper," in *2013 3rd Interdisciplinary Engineering Design Education Conference*, Santa Clara, 2013.
- [15] B. Zhao, "Web Scraping," in *Springer International Publishing AG*, USA, 2017.
- [16] K. A. B. Permana, M. Sudarma and W. G. Ariastina, "Sentiment Rating Analysis on Videos on Youtube Social Media Using STRUCT-SVM," *Majalah Ilmiah Teknologi Elektro*, vol. 18, no. 1, 2019.
- [17] M. A. S. Kencana, L. linawati and I. M. O. Widyantara, "Analisis Pemanfaatan Internet Di Pemerintah Kota Denpasar," *Majalah Ilmiah Teknologi Elektro*, vol. 15, no. 2, 2016.
- [18] G. R, "Comparative Analysis of Various Uninformed Searching Algorithms in AI," *International Journal of Computer Science and Mobile Computing*, vol. 8, no. 6, 2019.
- [19] Rismayani and Ardiansyah, "Aplikasi Berbasis Mobile untuk Pencarian Rute Angkutan Umum Kota Makassar Menggunakan Algoritma Depth First Search," *Jurnal Pekommas*, vol. 18, no. 3, 2015.
- [20] S. Suryadi, "Perancangan Aplikasi Pencarian File dengan Menggunakan Metode Best First Search," *Informatika : Jurnal Ilmiah AMIK Labuhan Batu*, no. 2, p. 2, 2014.
- [21] D. J. Priskilla and K. Arulanandam, "An Node Search of DFS with Spanning Tree in Undirected Graphs," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 9, no. 3, 2020.
- [22] M. A. Rohim, "Skripsi : Implementasi Ekstraksi Web (Web Scraping) pada Situs Berita Menggunakan Metode Ekspresi Reguler," *Universitas Jember*, 2019.
- [23] N. P. Lestari, "Uji recall and precision sistem temu kembali informasi opac perpustakaan its surabaya," *Journal Unair*, vol. 5, no. 3, 2016.