

IMPLEMENTASI TEKNIK WEB SCRAPING DAN FITUR DATA EKSTERNAL PADA SISTEM INFORMASI DOSEN PENELITIAN DAN PENGABDIAN DOSEN FAKULTAS TEKNIK UNIVERSITAS DIPONEGORO

Dionis Balan Pratama^{*)}, Aghus Sofwan dan Yosua Alvin Adi Soetrisno

Departemen Teknik Elektro, Universitas Diponegoro, Semarang, Indonesia
Jl. Prof. Sudharto, SH., Kampus UNDIP Tembalang, Semarang 50275, Indonesia

^{*)}E-mail: li234gpt@gmail.com

Abstrak

Dalam rangka mengoptimalkan hasil pelaksanaan penelitian dan pengabdian di lingkungan fakultas Teknik Universitas Diponegoro, dosen dan mahasiswa membutuhkan peran publikasi ilmiah. Publikasi ilmiah dapat menjadi sebuah panduan yang dapat digunakan sebagai referensi untuk penelitian dan pengabdian serupa atau di wilayah yang sama. Oleh karena itu, pada tugas akhir ini dibuat sebuah fitur data eksternal yang memanfaatkan teknik web scraping untuk diimplementasikan pada Sistem informasi penelitian dan pengabdian dosen Fakultas Teknik Universitas Diponegoro, yang akan memfasilitasi kebutuhan publikasi ilmiah yang terkait dengan penelitian atau pengabdian dosen. Pada pengujian validasi data hasil scraping, informasi-informasi yang berhasil di scrap telah sesuai dengan hasil yang diharapkan. Pada pengujian performa aplikasi web scraping, akan dibandingkan kinerja lama eksekusi dan penggunaan memori antara library scraping yang dipakai saat ini, yaitu BeautifulSoup, dengan library scraping lainnya, yaitu Scrapy dan Selenium. Hasil pengujian ini menyatakan bahwa BeautifulSoup menjadi library scraping tercepat sekaligus terhemat, dengan Scrapy diurutan kedua dan Selenium diurutan ketiga. Pada pengujian fitur data eksternal dengan menggunakan metode black box testing, hasil yang didapat telah sesuai dengan hasil yang di harapkan.

Kata kunci : Web scraping, Fitur Data Eksternal , BeautifulSoup

Abstract

In order to optimize the results of research and service implementation within the Faculty of Engineering, Diponegoro University, lecturers and students also need the role of scientific publications. Scientific publications can be a guide that can be used as a reference for similar research and service or in the same area. Therefore, in this final project, an external data feature is made that utilizes web scraping techniques to be implemented in the research information system and lecturer service at the Faculty of Engineering, Diponegoro University, which will facilitate the need for scientific publications related to research or lecturer service. In testing the validation of scraping data, the information that was successfully scraped was in accordance with the expected results. In testing the performance of the scraping web application, the performance of execution time and memory usage will be compared between the current scraping library in use, namely BeautifulSoup, and other scraping libraries, namely Scrapy and Selenium. The results of this test state that BeautifulSoup is the fastest and most economical scraping library, with Scrapy in second place and Selenium in third. In testing the external data features using the black box testing method, the results obtained are in accordance with the expected results.

Keywords: Web scraping, External Data Feature, BeautifulSoup

1. Pendahuluan

Pasal 9 ayat 1 undang-undang No. 12 Tahun 2012 tentang Pendidikan Tinggi, menyebutkan pelaksanaan Tri Dharma Perguruan Tinggi yaitu pendidikan, penelitian, dan pengabdian kepada masyarakat, menjadi tanggung jawab dan kewajiban sivitas akademika [1]. Dalam rangka mendukung kesuksesan pelaksanaan Tri Dharma Perguruan Tinggi, Fakultas Teknik Universitas

Diponegoro membuat sebuah sistem informasi penelitian dan pengabdian. Sistem informasi tidak hanya digunakan untuk menampilkan informasi, namun dapat digunakan untuk berinteraksi dengan data sehingga memberikan informasi untuk keperluan pengelolaan, pengambilan keputusan dan pengintegrasian data [2].

Untuk mengoptimalkan hasil pelaksanaan penelitian dan pengabdian di lingkungan fakultas Teknik Universitas

Diponegoro, dosen dan mahasiswa juga membutuhkan peran publikasi ilmiah. Publikasi ilmiah dapat menjadi sebuah panduan yang dapat digunakan sebagai referensi untuk penelitian dan pengabdian serupa atau di wilayah yang sama. Oleh karena ini, sebuah fitur dapat implementasikan pada Sistem informasi penelitian dan pengabdian dosen Fakultas Teknik Universitas Diponegoro, yang akan memfasilitasi kebutuhan publikasi ilmiah yang terkait dengan penelitian atau pengabdian dosen. Fitur publikasi ini juga dapat menjadi solusi akan publikasi ilmiah yang tersebar di banyak situs database publikasi ilmiah dengan mengkolleksikannya kedalam satu *serverdatabase* tanpa ada salinan data. Fitur ini akan memperkaya fitur yang terdapat pada sistem informasi penelitian dan pengabdian dosen Fakultas Teknik Universitas Diponegoro.

Dengan melihat fungsi pada fitur ini, terdapat teknik pengumpulan data yang dapat dimanfaatkan untuk mengoptimalkan fitur ini, yaitu *web scraping*. *Web scraping* adalah teknik pengumpulan data secara *online* yang lebih efisien dan efektif [3-5]. Dengan teknik ini, konten utama dari suatu halaman situs dapat diekstrak, dikoleksi dan selanjutnya dapat diproses oleh proses pengindeksan. Sebelumnya teknik ini dikenal kedalam beberapa istilah, diantaranya *screen scraping*, *data mining*, *web harvesting* ataupun teknik lain yang sejenis. Pembuatan aplikasi web scraping dapat dikombinasikan dengan berbagai metode, dalam laporan ini metode yang digunakan adalah metode *HTML parsing*.

Tugas akhir ini akan membahas implementasi fitur data eksternal dan teknik *web scraping* pada sistem informasi penelitian dan pengabdian dosen Fakultas Teknik Universitas Diponegoro.

2. Metode

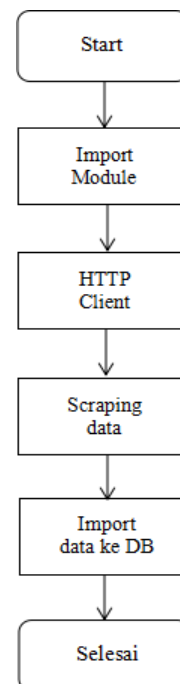
2.1. Deskripsi Sistem

Pada bagian ini, akan dijelaskan bagaimana cara kerja aplikasi ini dalam mengambil informasi publikasi ilmiah secara otomatis, dari beberapa situs database publikasi ilmiah. Kemudian proses pengolahan data informasi sampai disimpan dalam *database* mysql berbasis phpMyAdmin [6]. Selain itu, terdapat fitur publikasi ilmiah dilengkapi mesin pencarian berbasis *framework* codeigniter sebagai *interface* sehingga *user* dapat mengakses informasi publikasi ilmiah. Desain sistem dapat dilihat pada gambar 1.



Gambar 1. Desain Sistem [7]

Pengambilan data dilakukan secara otomatis menggunakan teknik *web scraping* data dapat dilihat alur *flowchart* pada Gambar 2. Proses *scraping* data diawali dengan *Urllib* yang akan mengakses URL (*Uniform Resource Locator*) situs *database* publikasi ilmiah target. Kemudian dilakukan proses utama dalam sistem ini yaitu *scraping* (pengambilan data). Data akan diambil dalam bentuk *string* yang akan disimpan sementara pada tipe data *list* dan disimpan ke dalam *database*. Kemudian dibuat sebuah fitur publikasi ilmiah sebagai sarana komunikasi antara *user* dengan *database*. fitur ini akan menampilkan daftar data hasil *scraping* dari *database*.



Gambar 2. Flowchart proses *scraping* [8]

Web scraping menggunakan *library* BeautifulSoup dengan *html5lib* sebagai *library parser*. BeautifulSoup dapat dijalankan menggunakan beberapa bahasa pemrograman, salah satunya adalah Python yang akan digunakan dalam pembangunan aplikasi *web scraping* ini [8]. Sedangkan pada fitur publikasi menggunakan *codeigniter* yang berbasis bahasa pemrograman php sebagai kerangka kerjanya [9-11].

Untuk pengkoneksian antara Python dengan phpMyAdmin digunakan *library* mysql [7]. Mysql merupakan *library* python yang digunakan untuk menghubungkan antara *database* MySQL dengan program python. Sedangkan *framework* Codeigniter sudah menyediakan fitur khusus untuk pengkoneksian dengan phpMyAdmin, sehingga tidak memerlukan *library* eksternal

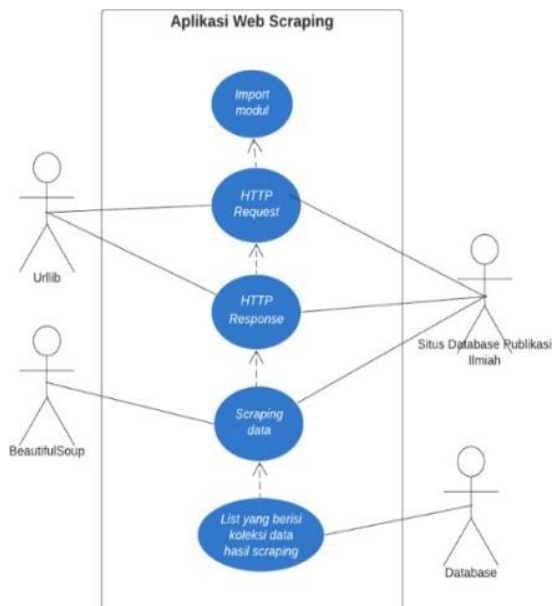
2.2. Desain Aplikasi

Desain aplikasi merupakan suatu proses perencanaan sebuah sistem yang berjalan dan merupakan tahap lanjutan dari analisa sistem, dimana pada perancangan sistem digambarkan rancangan sistem yang akan dibangun sebelum dilakukan pengkodean kedalam suatu bahasa pemrograman [12].

2.2.1. Diagram Use Case

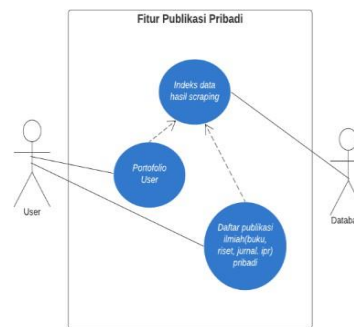
Diagram *use case* menggambarkan fungsi-fungsi yang ada pada fitur. Diagram *use case* lebih berfokus pada fungsi-fungsi fitur dari sudut pandang pihak luar, yang dalam hal ini adalah *user* aplikasi. Penulis membagi fitur menjadi 2 sub-fitur yaitu fitur publikasi pribadi dan fitur publikasi umum.

1. Diagram Use Case pada Aplikasi Web Scraping



Gambar 3. Diagram use case aplikasi web scraping [2]

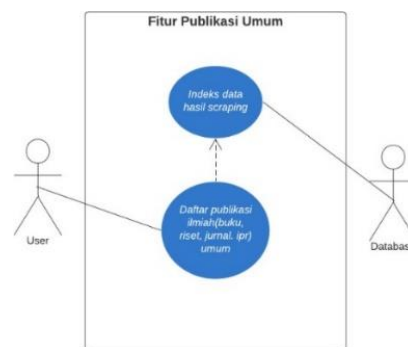
Gambar 3 menggambarkan fungsi-fungsi pada aplikasi *web scraping* dan interaksinya dengan para aktor [13]. Terdapat *import modul* yang berfungsi untuk *multifile*, *HTTP request/response* untuk menghubungkan dengan URL target, kemudian *scraping* yang berfungsi untuk mengambil data-data target, dan terakhir mengumpulkan data-data yg telah diambil kedalam suatu penyimpanan sementara, yaitu *list*, untuk kemudian di simpan di dalam *database MySQL*. Diagram Use Case pada Fitur Publikasi Pribadi



Gambar 4. Diagram use case fitur publikasi pribadi [2]

Gambar 4 menggambarkan fitur-fitur pada menu publikasi pribadi yang dapat diakses oleh *user*. *user* yang berperan sebagai dosen dapat melihat portofolio dan mengakses daftar publikasi ilmiah (buku, jurnal, ipr, riset) pribadi yang bersumber dari Sinta.

2. Diagram Use Case pada Fitur Publikasi Umum



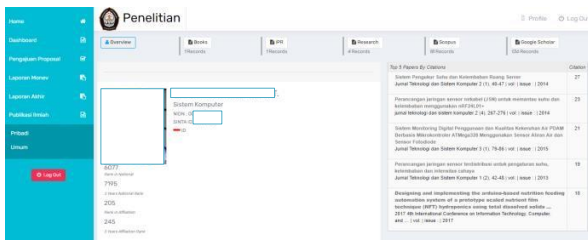
Gambar 5. Diagram use case fitur publikasi umum [2]

Gambar 5 menggambarkan fitur-fitur pada menu publikasi umum yang dapat diakses oleh *user*. *User* yang berperan sebagai dosen dapat mengakses daftar jurnal umum yang bersumber dari Gscholar, Scopus, Sinta, Emerald, ScienceDirect, Garuda, dan mengakses daftar publikasi ilmiah selain jurnal (buku, ipr, riset) umum yang bersumber dari Sinta.

3. Hasil dan Pembahasan

3.1. Tampilan Fitur Publikasi Pribadi

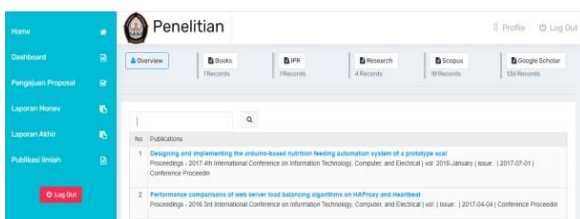
Ketika *user* masuk ke halaman pencarian publikasi pribadi dengan mengklik sub-menu publikasi pribadi, tampilan yang pertama kali muncul yaitu tampilan *overview user* sebagai halaman pertama dari aplikasi. Gambar 6 berikut merupakan tampilan *overview user* sesuai dengan rancangan yang telah dibuat.



Gambar 6. Tampilan overview user [9]

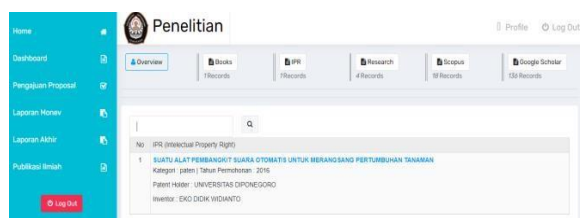
Secara default aplikasi akan langsung menampilkan halaman *user overview* setelah *user* menklik sub-menu publikasi pribadi. Halaman ini berisi portofolio singkat *user* seperti nama lengkap, afiliasi jurusan, nidn, sinta id. Terdapat juga beberapa prestasi individu *user* seperti ranking dalam nasional. *Ranking* dalam afiliasi universitas diponegoro, dan 5 *paper user* teratas berdasarkan jumlah disitasi.

Pada sub-menu publikasi pribadi ini terdapat menu *Books* menampilkan daftar buku yang ditulis *user* dengan *field* judul, ISBN, penulis, penerbit, tempat terbit dan tanggal terbit. Gambar 7 berikut merupakan tampilan menu *Books* pribadi sesuai rancangan yang telah dibuat.



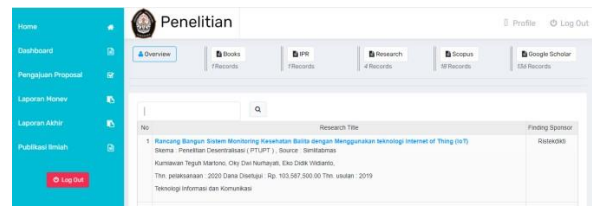
Gambar 7. Tampilan menu Books pribadi [9]

Menu *IPR* menampilkan daftar *IPR (Intellectual Property Rights)* yang ditulis *user* dengan *field* judul, ID, tahun permohonan, kategori, *patent holder*, dan inventor. Gambar 8 berikut merupakan tampilan menu *IPR* pribadi sesuai rancangan yang telah dibuat.



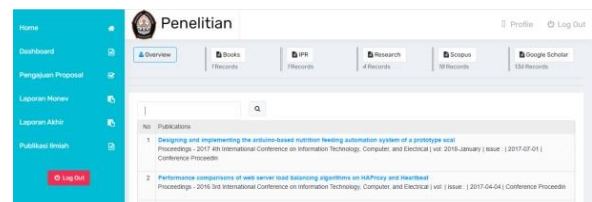
Gambar 8. Tampilan menu IPR pribadi [9]

Menu *Research* menampilkan daftar riset yang ditulis *user* dengan *field* judul, skema, penulis, tahun usulan, tahun pelaksanaan, dana disetujui, sumber, dan keahlian/keilmuan. Gambar 9 berikut adalah tampilan menu *research* pribadi sesuai rancangan yang dibuat.



Gambar 9. Tampilan menu Research pribadi [9]

Menu *scopus documents* menampilkan daftar jurnal terindeks *scopus* yang ditulis *user* dengan *field* judul, link detail, indeks, dan sitasi. Gambar 10 berikut merupakan tampilan menu *scopus* pribadi sesuai rancangan yang telah dibuat.



Gambar 10. Tampilan menu Scopus pribadi [9]

Menu *GScholar* menampilkan daftar buku yang ditulis *user* dengan *field* judul, penulis, link detail, indeks, dan sitasi. Gambar 11 berikut merupakan tampilan menu *gs documents* pribadi sesuai rancangan yang telah dibuat.



Gambar 11. Tampilan menu Gscholar pribadi [9]

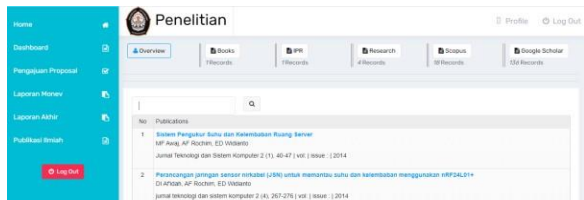
3.2. Tampilan Fitur Publikasi Umum

Pada saat *user* masuk ke halaman publikasi umum dengan mengklik sub-menu publikasi umum, tampilan pertama yang muncul adalah tampilan *dashboard* dengan enam pilihan sumber pencarian, yaitu *Scopus*, *ScienceDirect*, *Google Scholar*, *Emerald Insight*, *Sinta* dan *Garuda*. Gambar 12 berikut merupakan tampilan *dashboard* publikasi umum sesuai rancangan yang telah dibuat.



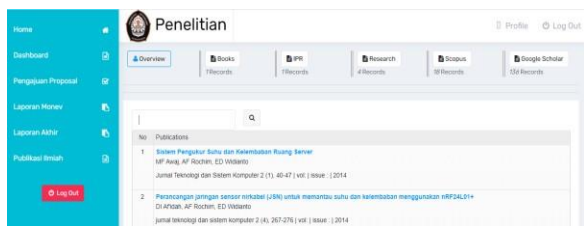
Gambar 12. Tampilan dashboard publikasi umum [9]

Halaman pencarian publikasi ilmiah bersumber Scopus, menampilkan kumpulan jurnal ter-indeks Scopus dengan *field* judul, link detail, indeks, dan sitasi. Gambar 13 berikut merupakan tampilan daftar publikasi umum bersumber Scopus sesuai rancangan yang telah dibuat.



Gambar 13. Tampilan menu Scopus umum [9]

Halaman pencarian publikasi ilmiah bersumber Google Scholar, menampilkan kumpulan jurnal ter-indeks Google Scholar dengan *field* judul, penulis, link detail, indeks, dan sitasi. Gambar 14 berikut merupakan tampilan daftar publikasi umum bersumber Google Scholar sesuai rancangan yang telah dibuat.



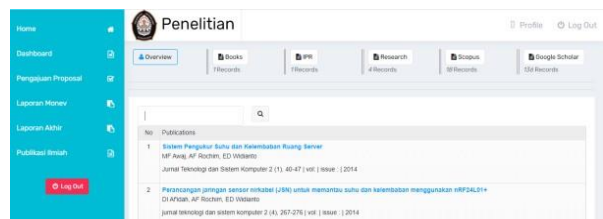
Gambar 14. Tampilan menu Gscholar umum [9]

Halaman pencarian publikasi ilmiah dengan bersumber Emerald menampilkan kumpulan jurnal ter-indeks Emerald dengan *field* judul, penulis, link detail, dan tanggal terbit. Gambar 15 berikut merupakan tampilan daftar publikasi umum bersumber Emerald sesuai rancangan yang telah dibuat.



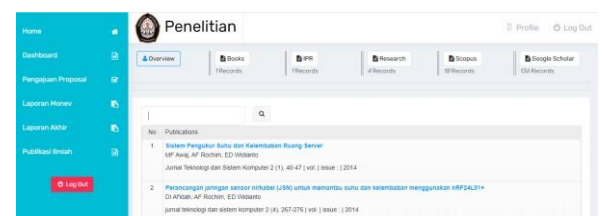
Gambar 15. Tampilan menu Emerald umum [9]

Halaman pencarian publikasi ilmiah dengan bersumber Garuda menampilkan kumpulan jurnal ter-indeks Garuda dengan *field* judul, penulis, link detail, dan tanggal terbit. Gambar 16 berikut merupakan tampilan daftar publikasi umum bersumber Garuda sesuai rancangan yang telah dibuat.



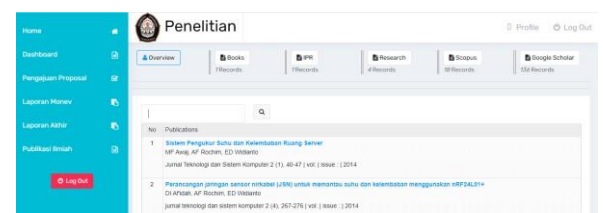
Gambar 16. Tampilan menu Garuda umum [9]

Pada pencarian publikasi ilmiah dengan bersumber Sinta, terdapat fitur *select dropdown* yang menampilkan tiga opsi, yaitu IPR, Research, dan Book. Opsi IPR menampilkan halaman pencarian IPR bersumber Sinta dengan *field* judul, ID, tahun permohonan, kategori, *patent holder*, dan *inventor*. Gambar 17 berikut merupakan tampilan daftar IPR umum sesuai rancangan yang telah dibuat.



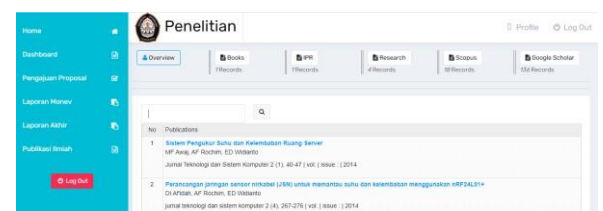
Gambar 17. Tampilan menu Gscholar pribadi [9]

Opsi *Research* menampilkan halaman pencarian riset bersumber Sinta dengan *field* judul, skema, penulis, tahun usulan, tahun pelaksanaan, dana disetujui, sumber, dan keahlian/keilmuan. Gambar 18 berikut merupakan tampilan daftar research umum sesuai rancangan yang telah dibuat.



Gambar 18. Tampilan menu Research umum [9]

Opsi *Book* menampilkan halaman pencarian buku bersumber Sinta dengan *field* judul, ISBN, penulis, penerbit, tempat terbit dan tanggal terbit. Gambar 19 berikut merupakan tampilan daftar buku umum sesuai rancangan yang telah dibuat.



Gambar 19. Tampilan menu Book umum [9]

3.3. Pengujian Performa pada Aplikasi Web Scraping

Pengujian dilakukan dengan membandingkan kinerja lama eksekusi dan penggunaan memori antara 3 library scraping, yaitu Scrapy, BeautifulSoup, dan Selenium [14]. Pengujian dilakukan disetiap laman berdasarkan 3 kondisi, yakni saat pengaksesan URL target, saat *scraping* dan koleksi data, dan saat pengiriman data ke *database server*.

1. Pengujian Lama Eksekusi

Pada pengujian lama eksekusi, akan dibandingkan waktu yang dibutuhkan oleh Scrapy, BeautifulSoup, dan Selenium dalam menyelesaikan eksekusi di setiap kondisi pengujian [15]. Pengujian ini menggunakan *library time.it* dalam mengukur lama eksekusinya.

Tabel 1. Hasil pengujian lama eksekusi Scrapy, BeautifulSoup dan Selenium [6][13]

Jenis Data	Kondisi Pengujian	Lama Eksekusi (s)		
		Scrapy	BeautifulSoup	Selenium
Jurnal Google Scholar	a	1,61	0,59	4,2
	b	0,68	1,04	1,15
	c	0,73	0,45	0,34
Jurnal Scopus	a	1,52	0,46	4,6
	b	0,48	0,86	1,52
	c	0,56	0,85	0,73
JurnalEmerald Insight	a	1,24	0,37	3,91
	b	0,57	0,83	1,60
	c	0,52	0,46	0,6
Jurnal Garuda	a	1,48	0,79	3,62
	b	0,56	0,95	1,73
	c	0,24	0,30	0,62
Jurnal ScienceDirect	a	1,83	0,67	3,60
	b	0,67	1,10	2,26
	c	0,97	0,81	0,68
IPR	a	1,37	0,53	4,54
	b	0,15	0,68	1,32
	c	0,27	0,2	0,58
Research	a	1,76	0,62	3,41
	b	0,38	0,79	1,85
	c	0,76	0,49	0,52
Book	a	1,27	0,59	4,48
	b	0,17	0,59	1,68
	c	0,37	0,38	0,52

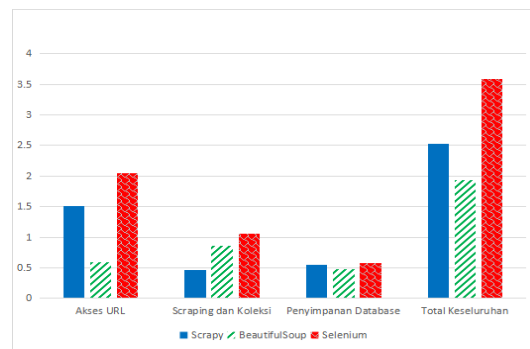
Keterangan :

a : Pengaksesan URL target

b : *Scraping* dan koleksi data

c : Penyimpanan ke *database*

Dari data yang ada pada tabel 1, penulis membuat grafik perbandingan selisih rata-rata lama eksekusi. Grafik didapat dengan menjumlahkan lama eksekusi saat mengambil seluruh jenis data yang dibutuhkan dan membaginya dengan jumlah jenis data yang diambil. Grafik perbandingan selisih rata-rata lama eksekusi dapat dilihat pada gambar 20.



Gambar 20. Grafik perbandingan (S) Scrapy, BeautifulSoup, dan Selenium [6][13]

Gambar 20 menampilkan grafik perbandingan selisih rata-rata lama eksekusi antara Scrapy, BeautifulSoup dan Selenium. Dapat dilihat pada saat kondisi akses URL, BeautifulSoup adalah yang tercepat dengan Scrapy di urutan kedua dan Selenium di urutan ketiga. BeautifulSoup lebih cepat dari Scrapy dengan selisih rata-rata 1 sekon dan Scrapy lebih cepat dari Selenium dengan selisih rata-rata 0,5 sekon. Tetapi pada kondisi *scraping*, Scrapy menjadi yang tercepat dengan unggul selisih rata-rata 0,4 sekon dari BeautifulSoup, di urutan kedua BeautifulSoup yang unggul selisih rata-rata 0,3 sekon dari Selenium. Sedangkan pada kondisi penyimpanan data ke *database*, perbedaan lama eksekusi yang ditemukan sangatlah kecil. Jika di total keseluruhan waktu dari 3 kondisi, BeautifulSoup menjadi yang tercepat dengan total waktu 1,93 sekon, diikuti Scrapy dengan total waktu 2,52 sekon, dan terakhir Selenium dengan total waktu 3,59 sekon.

2. Pengujian Penggunaan Memori

Pada pengujian penggunaan memori, akan dibandingkan jumlah memori yang dihabiskan oleh Scrapy, BeautifulSoup, dan Selenium dalam menyelesaikan eksekusi di setiap kondisi pengujian. Pengujian ini menggunakan *library time.it* dalam mengukur jumlah penggunaan memori.

Dari data yang ada pada tabel 2, penulis membuat grafik perbandingan selisih rata-rata penggunaan memori. Grafik didapat dengan menjumlahkan penggunaan memori saat mengambil seluruh jenis data yang dibutuhkan dan membaginya dengan jumlah jenis data yang diambil. Grafik perbandingan selisih rata-rata penggunaan memori dapat dilihat pada gambar 21. Gambar 21 Grafik perbandingan selisih (MB) Scrapy, BeautifulSoup, dan Selenium.

Tabel 2. Hasil pengujian penggunaan memori Scrapy, BeautifulSoup dan Selenium [6][13]

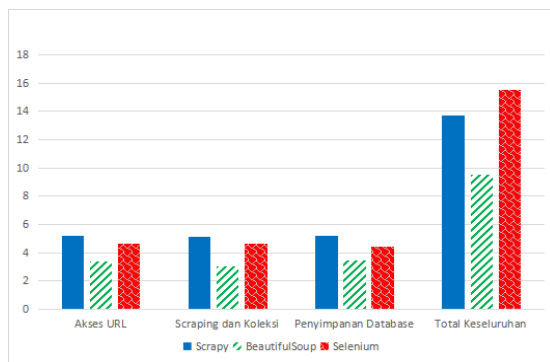
Jenis Data	Kondisi Pengujian	Penggunaan Memori (MB)		
		Scrapy	BeautifulSoup	Selenium
Jurnal Google Scholar	a	5,93	3,68	5,17
	b	5,36	2,74	4,12
	c	5,43	3,69	5,3
Jurnal Scopus	a	4,87	3,27	4,82
	b	4,92	2,23	4,61
	c	4,91	3,21	4,23
Jurnal Emerald Insight	a	5,45	3,87	4,43
	b	5,16	2,83	5,09
	c	5,53	3,80	4,95
Jurnal Garuda	a	4,78	2,99	4,13
	b	4,85	2,54	4,29
	c	5,65	3,96	5,03
Jurnal ScienceDirect	a	5,34	3,68	4,92
	b	5,36	3,73	4,78
	c	5,45	3,78	4,93
IPR	a	5,38	3,51	4,88
	b	5,34	3,46	5,17
	c	4,43	2,57	3,84
Research	a	4,78	2,99	4,13
	b	4,85	2,54	4,29
	c	5,42	3,17	4,72
Book	a	1,27	0,59	4,48
	b	0,17	0,59	1,68
	c	0,37	0,38	0,52

Keterangan :

a : Pengaksesan URL target

b : Scraping dan koleksi data

c : Penyimpanan ke database



Gambar 21. Grafik perbandingan (MB) Scrapy, BeautifulSoup, dan Selenium [6][13]

Gambar 21 menampilkan grafik perbandingan selisih rata-rata penggunaan memori antara Scrapy, BeautifulSoup dan Selenium. Dapat dilihat pada semua kondisi, BeautifulSoup menggunakan memori paling sedikit diikuti Selenium di urutan kedua dan Scrapy di urutan ketiga. Dalam kondisi akses URL, BeautifulSoup lebih hemat dari Selenium dengan selisih rata-rata 1,2 MB dan Selenium lebih hemat dari Scrapy dengan selisih rata-rata 0,5 MB. Dalam kondisi *scraping*, BeautifulSoup lebih hemat dari Selenium dengan selisih rata-rata 1,5 MB dan selenium lebih hemat dari Scrapy dengan selisih rata-rata 0,5 MB. Dalam kondisi menyimpan data ke *database*,

BeautifulSoup lebih hemat dari Selenium dengan selisih rata-rata 1 MB dan selenium lebih hemat dari Scrapy dengan selisih rata-rata 0,8 MB. Jika di total keseluruhan penggunaan memori dari 3 kondisi, BeautifulSoup memiliki total 9,4 MB, diikuti Scrapy dengan total waktu 13,7 MB, dan terakhir Selenium dengan total waktu 15,5 MB.

4. Kesimpulan

Tugas Akhir ini tentang implementasi teknik web scraping dan fitur data eksternal pada sistem informasi penelitian dan pengabdian dosen Fakultas Teknik Universitas Diponegoro. Fitur data eksternal memiliki beberapa fungsi, dimana fungsi utamanya akan menyediakan kebutuhan publikasi ilmiah pada proses kegiatan penelitian dan pengabdian. Fitur ini memanfaatkan teknik *web scraping* untuk mengumpulkan publikasi ilmiah terkait kegiatan penelitian dan pengabdian dari beberapa situs *database* publikasi ilmiah. Pada pengujian validasi data hasil *scraping*, informasi-informasi yang berhasil di *scrap* telah sesuai dengan hasil yang diharapkan. Pada pengujian performa aplikasi *web scraping*, akan dibandingkan kinerja lama eksekusi dan penggunaan memori antara *library scraping* yang dipakai saat ini, yaitu BeautifulSoup, dengan *library scraping* lainnya, yaitu Scrapy dan Selenium. Hasil pengujian ini menyatakan bahwa BeautifulSoup menjadi *library scraping* tercepat dengan total waktu 1,93 sekon, Scrapy di urutan kedua total waktu 2,52 sekon, dan Selenium di urutan ketiga total waktu 3,59 sekon. Tetapi Scrapy menjadi yang terhemat dalam penggunaan memori dengan total penggunaan memori 9,4 MB, Scrapy di urutan kedua dengan total penggunaan memori 13,7 MB, dan Selenium di urutan ketiga dengan total penggunaan memori 15,5 MB. Pada pengujian fitur data eksternal dengan menggunakan metode *black box testing*, hasil yang didapat telah sesuai dengan hasil yang di harapkan.

Referensi

- [1]. Undang-Undang No. 12 Tahun 2012, Pasal 9 ayat 1, tentang Pendidikan Tinggi.
- [2]. Stair, Ralph M. Principles of Information Systems. 2020.
- [3]. R. Abrantes and J. Figueiredo, "Information systems and change in project based organizations", *Procedia Computer Science*, vol. 181, pp. 367-376, 2021. Available: 10.1016/j.procs.2021.01.184.
- [4]. Q. Liu, G. Feng, X. Zhao and W. Wang, "Minimizing the data quality problem of information systems: A process-based method", *Decision Support Systems*, vol. 137, p. 113381, 2020. Available: 10.1016/j.dss.2020.113381.
- [5]. C. Pampulim Caldeira, *Information Processing & Management*, vol. 43, no. 3, pp. 824-825, 2007. Available: 10.1016/j.ipm.2006.10.004.

- [6]. A. N. Syahrudin and T. Kurniawan, "Perbandingan Penggunaan NOSQL MONGODB dan MYSQL pada Basis Data Forum Komunikasi," *J. Menej. Inf.*, vol. 6, no. 1, pp. 134–142, 2016.
- [7]. A. Penelitian, "Perancangan dan Pembuatan Crawler Website Partsouq Menggunakan Beautiful Soup 4," no. April, pp. 1–29, 2019.
- [8]. A. N. Syahrudin and T. Kurniawan, "Input Dan Output Pada Bahasa," *J. Dasar Pemrograman Python STMIK*, no. January, pp. 1–7, 2018.
- [9]. R. Andika, "Penerapan CI (Codeigniter) dalam Pengembangan Sistem Informasi Manajemen Surat dan Pengarsipan (Studi kasus: PT Semen Padang)," Fakultas Sains dan Teknologi, Universitas Islam Negeri Syarif Hidayatullah Jakarta, 2011.
- [10]. Yuhefizar, "Cara Mudah dan Murah Membangun dan Mengelola Website. Yogyakarta: Graha Ilmu.," *Berk. Ilmu Perpust. dan Inf.*, 2012.
- [11]. B. Sidik, "Framework Codeigniter," Bandung Inform., 2012.
- [12]. V. Mitra, H. Sujaini, and A. Negara, "Rancang Bangun Aplikasi Web Scraping untuk Korpus Paralel Indonesia - Inggris dengan Metode HTML DOM," *J. Sist. dan Teknol. Inf.*, vol. 5, no. 1, pp. 1–6, 2017.
- [13]. T. Rizaldi and H. Arief, "Perband PemrogramanPython STMIK, no. January, pp. 1–7, 2018.ingan Metode Web Scrapping menggunakan CSS Selector dan XPATH Selector," *J. ILKOM Universitas Udayana*, vol. 10, no. 2, 2017.
- [14]. E. Uzun, T. Yerlikaya, and O. Kirat, "Comparision of Python Libraries Used for Web Data Extraction," vol. 24, 2018.
- [15]. M. Levi, H. N. Palit, S. Rostianingsih, and J. Siwalankerto, "Perbandingan Performa Tools Web Scraping pada Website dengan Data Statis dan Dinamis," pp. 1–7, 2020.