

Prototype Aplikasi ONE Search Engine Dengan Fitur User Interface Dan Admin Control Panel Dan Pengujian Crawling Dalam Jangka Menengah

Aldian Asmara¹, Muhammad Eka Suryana², Med Irzal³

Program Studi Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Negeri Jakarta, DKI Jakarta, Indonesia

¹aldian.asmr@gmail.com, ²eka-suryana@unj.ac.id, ³medirzal@unj.ac.id

Abstract

Mesin pencari merupakan sebuah program komputer yang berfungsi untuk membantu pengguna dalam menemukan informasi dengan kata kunci tertentu. Pada penelitian [5] telah dirancang sebuah arsitektur search engine dengan mengintegrasikan web crawler, algoritma page ranking dan document ranking. Penelitian tersebut memiliki beberapa kekurangan yaitu tidak adanya admin console untuk manajemen dan visualisasi data hasil pengindeksan search engine yang telah dibuat. Penelitian ini memiliki tujuan untuk menyediakan suatu cara bagi pengguna untuk mengakses search engine yang telah dibuat beserta admin console untuk manajemen dan visualisasi data hasil pengindeksan search engine. Informasi pendukung untuk melakukan penelitian ini berasal dari studi literatur jurnal-jurnal terkait dan diskusi yang diadakan peneliti dengan stakeholder. Proses pengembangan yang digunakan dalam penelitian ini menggunakan metode Scrum dengan menggunakan teknologi Python dan Javascript. Hasil akhir dari penelitian ini adalah sebuah user interface search engine admin panel untuk manajemen dan visualisasi data hasil pengindeksan search engine.

Pendahuluan

Dengan bertambah banyaknya informasi yang berada di internet setiap harinya, tentu saja mencari informasi yang kita inginkan secara manual di Web sangatlah memakan waktu. Oleh karena itulah *search engine* atau mesin pencari hadir untuk menangani masalah tersebut. Mesin pencari atau search engine adalah program berbasis web yang dapat diakses di internet yang memiliki tujuan utama yaitu mencari yang informasi yang relevan dengan cepat terhadap *query* yang pengguna kirim. Mesin pencari atau *search engine* bekerja dengan cara mencocokkan *query* dari pengguna kepada index yang *search engine* atau mesin pencari telah buat

Mesin pencari atau yang biasa disebut *search engine* merupakan sebuah program komputer yang berguna untuk membantu pengguna dalam mencari situs web berdasarkan permintaan pencarian pengguna. Mesin pencari sebenarnya tidak berbeda dengan *website* pada umumnya, hanya saja perannya lebih terfokus pada pengumpulan dan pengorganisasian berbagai informasi di internet sesuai dengan kebutuhan penggunanya. Selain untuk memudahkan pencarian, mesin pencari juga berguna untuk meningkatkan pengunjung sebuah situs web.

Di masa ini, mesin pencari komersial seperti Google, Bing, Yandex banyak dipergunakan oleh *user* dalam kehidupannya sehari-hari. Meskipun demikian kelemahan dari *search engine* komersial ini terletak pada (1) Kerahasiaan *query* pencarian (2) *Mining* data

query user untuk keperluan *ad personalization* (3) Monopoli informasi sehingga yang muncul di ranking teratas berdasarkan algoritma yang ditentukan oleh perusahaan-perusahaan besar tersebut. Kemajuan teknologi dan solusi atas masalah masalah sebelumnya mengakibatkan terdapat upaya dari komunitas peneliti dan pengembang aplikasi untuk membuat *search engine* alternatif dalam hal ini terdapat dua alternatif yang cukup populer yang pertama adalah aplikasi Apache Solr berupa *tools* pencarian relevansi berbasis teks yang disediakan oleh Apache Foundation. Kelemahan dari Apache Solr adalah *tools* ini hanya dapat menggantikan peran pencarian text pada suatu korpus dokumen. Sebagai perbandingan, mesin pencari komersial seperti yang telah disebutkan diatas selain memiliki fitur pencarian dokumen berdasarkan relevansi namun juga memiliki fitur perankingan dokumen yang berfungsi untuk pengurutan halaman tanpa ketergantungan pada isi konten. Alternatif lainnya yang berupaya untuk mereplikasi peran mesin pencari komersial secara sempurna adalah YACY yang bersifat *open source* dan juga terdistribusi secara *peer-to-peer* seperti apache solr, yacy memiliki kelemahan yang sama tidak menyediakan layanan untuk perankingan dokumen.

Visualisasi data adalah metode utama untuk membantu data mendapatkan interpretasi data dan juga menemukan nilainya. Data disajikan secara visual untuk menyampaikan interpretasi dasar mengenai apa yang data katakan tanpa adanya kesulitan [2]. Saat ini, visualisasi data atau visualisasi informasi

menjadi topik yang menarik dan menjadi bidang penelitian yang luas. [4]

Dalam search engine, *user interface* atau tampilan merupakan hal yang penting mengingat *search engine* sering sekali digunakan dalam kehidupan sehari-hari bahkan menjadi bagian hidup dari seseorang. Menurut [3], pada umumnya, skenario dari penggunaan *user interface* atau tampilan dari *search engine* adalah sebagai berikut:

1. Pengguna memiliki kata yang ingin dicari dan mengirimkannya kepada mesin pencari atau *search engine*
2. *Search engine* merespon kata yang dikirimkan oleh user
3. *Search engine* akan mencari dokumen yang sesuai dengan query yang pengguna kirim dan menampilkannya ke tampilan *search engine*
4. Yang terakhir user menentukan apakah dokumen yang diterima user relevan atau tidak dengan yang diharapkan pengguna

Pada penelitian "Perancangan arsitektur *search engine* dengan mengintegrasikan *web crawler*, algoritma *page ranking*, dan *document ranking*" [5] telah dirancang arsitektur *search engine* berbasis *console*. Pada penelitian ini terdapat beberapa kekurangan yaitu tidak adanya *admin console* untuk manajemen dan visualisasi data hasil pengindeksan *search engine* yang telah dibuat.

Mempertimbangkan masih dibutuhkannya mesin pencarian yang bersifat *open source* terdistribusi secara *peer-to-peer* namun juga memiliki dukungan untuk melakukan perangkian dokumen. Penelitian ini akan merancang tampilan dari *search engine* dengan *admin console* untuk visualisasi dan manajemen hasil indeks dengan mengintegrasikan penelitian dari [5] yang berfokus pada perancangan arsitektur *search engine* dengan mengintegrasikan *web crawler*, algoritma *page ranking* dan *document ranking*.

Studi Literatur

Review Search Engine [5]

Pada penelitian [5], telah dirancang sebuah arsitektur *search engine* dengan mengintegrasikan *web crawler*, *page ranking* dan *document ranking*. Flowchart dari pengintegrasian tersebut adalah sebagai berikut

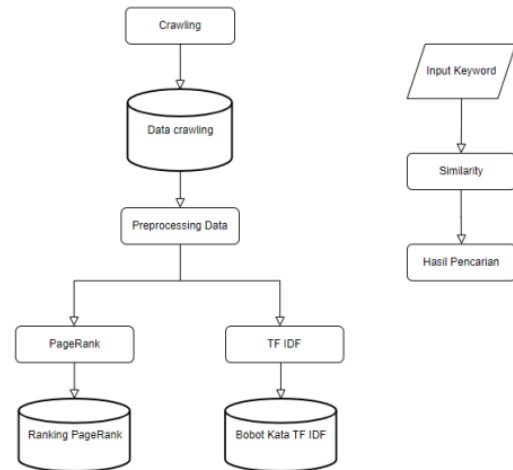


Figure 1: Flowchart search engine[5]

Untuk *web crawler* digunakan algoritma *crawling* bernama *breadth first search*. Pada penerapannya, algoritma *breadth first search* ini akan melakukan pencarian yang dimulai dari pemilihan *node* awal kemudian dilanjutkan dengan pencarian bertahap level demi level. Algoritma ini merupakan bentuk paling sederhana dari algoritma *crawling* [5].

Pada saat tahap *crawling*, digunakan pustaka *beautifulsoup* python. Pustaka *beautifulsoup* bertugas untuk mengolah setiap halaman yang algoritma *crawling* kunjungi untuk mendapatkan informasi yang akan disimpan dalam *database* yang ada.

Preprocessing data dilakukan agar data yang digunakan bebas noise, memiliki ukuran yang lebih kecil, dan lebih terorganisir sehingga dapat diproses lebih lanjut [5].

Pada tahap selanjutnya setelah melewati tahap *pre-processing data*, data dapat dilanjutkan ke dua tahap yaitu *pagerank* dan *tf-idf*. *TF-IDF* adalah metode pemberian bobot pada hubungan antara kata (*word*) dan dokumen, metode ini adalah metode yang paling umum digunakan untuk menghitung bobot setiap kata dalam pencarian informasi [5]. Pada tahapan *pagerank* digunakan algoritma *Pagerank* untuk menentukan peringkat halaman-halaman yang ada.

Agile

Pengembangan perangkat lunak menggunakan *agile* merupakan salah satu metode pengembangan perangkat lunak yang ada. Kata "*agile*" memiliki arti cepat, ringan dan bebas bergerak. Konsep pengembangan perangkat lunak menggunakan *agile* ditemukan oleh Kent Beck dan 16 koleganya dengan menyatakan *agile* adalah cara dalam membangun sebuah perangkat lunak dengan cara mengerjakannya dan membantu satu sama lain dalam mem-

bangunnya dalam satu waktu. Dalam pengembangan perangkat lunak menggunakan *agile*, interaksi dan personil adalah hal yang penting dibandingkan proses dan alat kerja, Perangkat lunak yang bekerja lebih penting dari dokumentasi yang lengkap, Kolaborasi antara klien lebih penting dari negosiasi kontrak dan responsif dalam perubahan adalah hal yang lebih penting dari mengikuti rencana yang telah dibuat. Spertim model lainnya, *agile* memiliki kelebihan dan tidak cocok untuk semua tipe proyek. *Agile* membuat model proses yang toleran terhadap perubahan kebutuhan sehingga perubahan dapat dilakukan dengan cepat [1].

Scrum

Scrum adalah metodologi pengembangan dan manajemen yang dikembangkan oleh Jeff Sutherland pada tahun 1993, dengan tujuan mengikuti prinsip *Agile*. Metode ini fokus pada strategi pengembangan perangkat lunak holistik, di mana tim pengembang bekerja sebagai satu unit untuk mencapai tujuan bersama.

Dalam pelaksanaannya, *Scrum* terdiri dari tiga peran utama: *Product Owner*, *Scrum Master*, dan *Team*. *Product Owner* bertanggung jawab menentukan spesifikasi perangkat lunak dan membuat *Product Backlog*. *Team*, sebagai entitas pengembangan, bekerja untuk menyelesaikan tugas-tugas dalam *Product Backlog*. *Scrum Master* memperkenalkan dan mengimplementasikan metode *Scrum* pada tim.

Proses pengerjaan proyek dengan *Scrum* dimulai dengan penggambaran sistem yang akan dibuat, kemudian *Product Owner* menggambarkan proses bisnis ke dalam *Product Backlog*. *Sprint*, sebagai bagian dari *Scrum*, adalah tujuan yang ingin dicapai dalam iterasi selanjutnya. Setiap *sprint* dimulai dengan *Sprint Planning Meeting*, diikuti oleh pertemuan harian *Daily Scrum Meeting* di mana setiap anggota tim berdiskusi tentang progres dan tugas selanjutnya. *Sprint* diakhiri dengan pertemuan untuk mendemonstrasikan hasil pekerjaan. *Scrum* dirancang untuk meningkatkan fleksibilitas dan responsivitas dalam pengembangan perangkat lunak [1].

Pengembangan

Pengembangan Dengan Scrum

Agar penelitian ini menjadi lebih terstruktur dan mudah, maka penelitian ini akan menggunakan metode *scrum* sebagai metode pengembangan sistemnya. Komponen-komponen *scrum* terdiri dari *product backlog*, *sprint backlog*, *daily scrum* dan *sprint* lalu dilanjutkan dengan pengujian sistem yang sudah

dibuat.

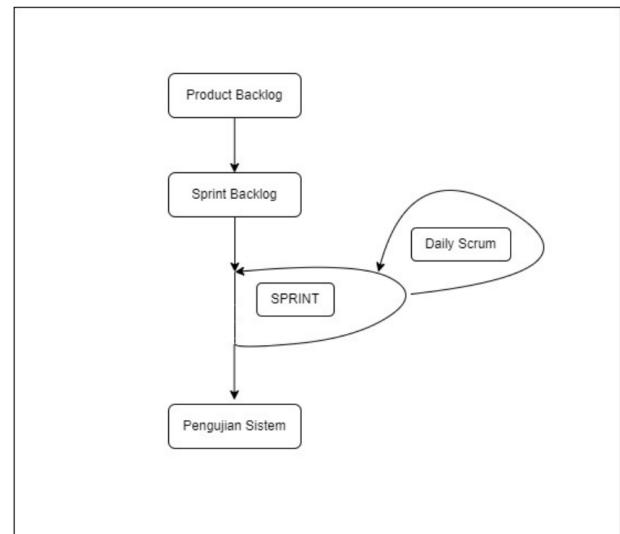


Figure 2: Tahapan penelitian dengan menggunakan metode scrum

Product Backlog *Product backlog* merupakan kumpulan tugas yang akan dilaksanakan. *Product backlog* seperti ditunjukkan oleh tabel 1 terdiri dari 3 komponen yaitu *story*, *sprint* dan *status*. *Story* ialah sebuah pekerjaan besar yang dapat dibagi bagi lebih kecil lagi menjadi tugas tugas kecil. *Sprint* menandakan sprint berapa *story* tersebut akan diselesaikan. *Status* memberitahu apakah sprint tersebut sudah terlaksanakan atau belum.

Table 1: *Product Backlog*

No.	Story	Sprint	Status
1	Fitur pencarian pengguna	1, 2, 8	Selesai
2	Fitur <i>page ranking</i>	4, 5	Selesai
3	Fitur <i>staff</i>	5, 8	Selesai
4	Fitur <i>document ranking</i>	5, 7	Selesai
5	Fitur <i>crawling</i>	3, 6	Selesai
6	Struktur proyek	2	Selesai
7	<i>multi-threaded service</i>	10	Selesai
8	<i>Service deployment</i>	11	Selesai
9	Background task dengan <i>Celery</i>	12	Selesai
10	Pengujian penggunaan memori	13	Selesai

Sprint Backlog *Sprint backlog* merupakan daftar tugas tugas kecil yang perlu dilaksanakan pada suatu *sprint*.

Sprint *Sprint* merupakan masa dimana pengerjaan tugas tugas yang telah direncanakan pada suatu

sprint dilakukan. Lama durasi setiap *sprint* ditentukan oleh *scrum master* yang telah disepakati bersama

Daily Scrum *Daily scrum* merupakan pertemuan dengan *scrum master* untuk membahas tugas apa yang telah dicapai hari kemarin dan tugas apa yang ingin dicapai hari ini yang dilaksanakan setiap hari.

Analisis Kebutuhan

Analisis kebutuhan dilakukan guna mengumpulkan informasi mengenai kebutuhan fitur aplikasi dan prioritas fitur aplikasi yang akan dibuat. Dari analisis kebutuhan yang dilakukan dihasilkan sebuah *usecase diagram* yang didefinisikan sebagai berikut.

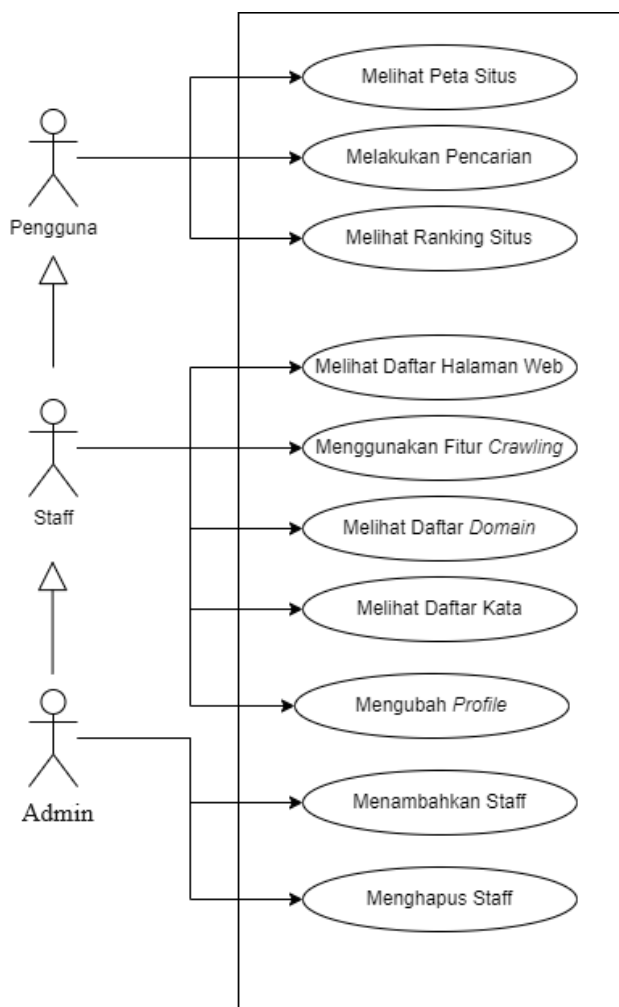


Figure 3: Use case diagram

Hasil dan Pembahasan

Implementasi User Interface

Pada implementasi *user interface*, digunakan bahasa pemrograman *javascript*. Adapun hasil implementasi *user interface* sebagai berikut

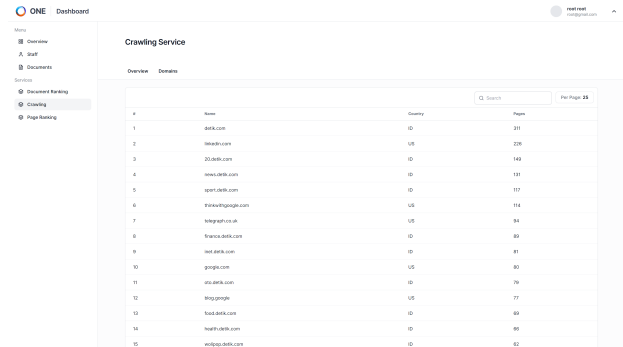


Figure 4: Halaman Daftar Domain

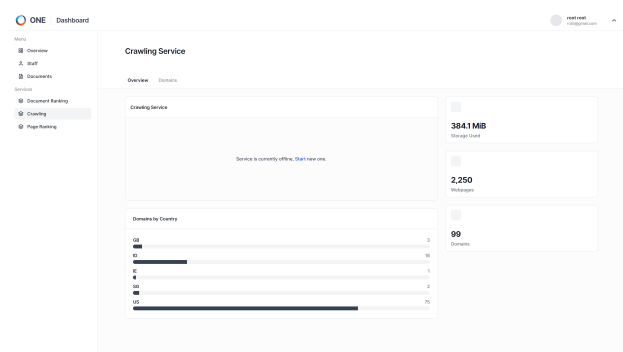


Figure 5: Halaman Service Crawling

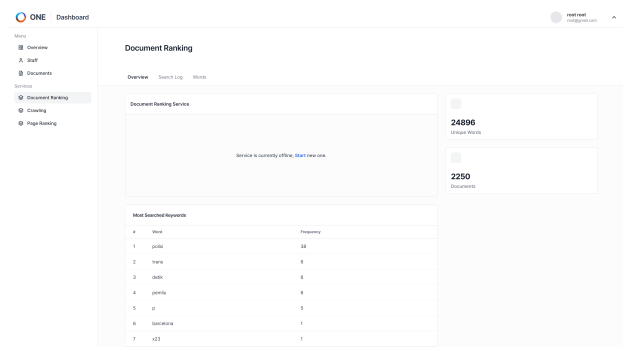


Figure 6: Halaman Document Ranking

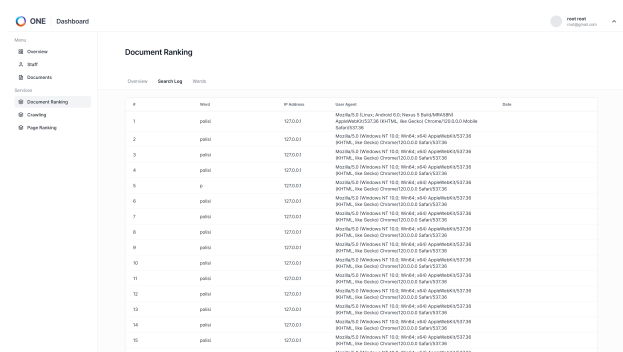


Figure 7: Halaman search log

Document Ranking

Overview Search Log Words

Search: Per Page: 10

ID	Word	Frequency
10000001	word	100
10000002	2023	857
10000003	word	804
10000004	word	790
10000005	id	766
10000006	id	776
10000007	id	721
10000008	id	721
10000009	id	683
10000010	id	672
10000011	id	662
10000012	id	648
10000013	id	647
10000014	advertisement	634
10000015	id	625
10000016	content	621
10000017	id	614

Figure 8: Halaman words

Add Staff

Personal Info

First Name:

Last Name:

Email:

Phone:

Address:

Role:

Save

Figure 12: Halaman tambah staff

Page Ranking

Overview

Page Ranking Service

Service is currently offline. Start new one.

384.1 MB Data Usage

337,251 Page Ranking

2,250 Total Index

1 Page Rank Vector Size

Top Websites

ID	Word
1	id
2	id
3	id
4	id
5	id
6	id
7	id
8	id
9	id
10	id

Figure 9: Halaman page ranking

Update Profile

Personal Info

First Name:

Last Name:

Email:

Phone:

Address:

Role:

Save

Figure 13: Halaman ubah profile

Search Results

Search:

Top Websites

ID	Word
1	id
2	id
3	id
4	id
5	id
6	id
7	id
8	id
9	id
10	id

Figure 10: Halaman pencarian

Staff

Search: Per Page: 10

ID	Staff	Role	Status
1	id	id	id
2	id	id	id
3	id	id	id
4	id	id	id
5	id	id	id

Figure 14: Halaman daftar staff

Search Results

Search:

Top Websites

ID	Word
1	id
2	id
3	id
4	id
5	id
6	id
7	id
8	id
9	id
10	id

Figure 11: Halaman hasil pencarian

ONE

Sign In

Please sign in with your staff account

Email:

Password:

Sign In

Figure 15: Halaman login staff

Desain Database

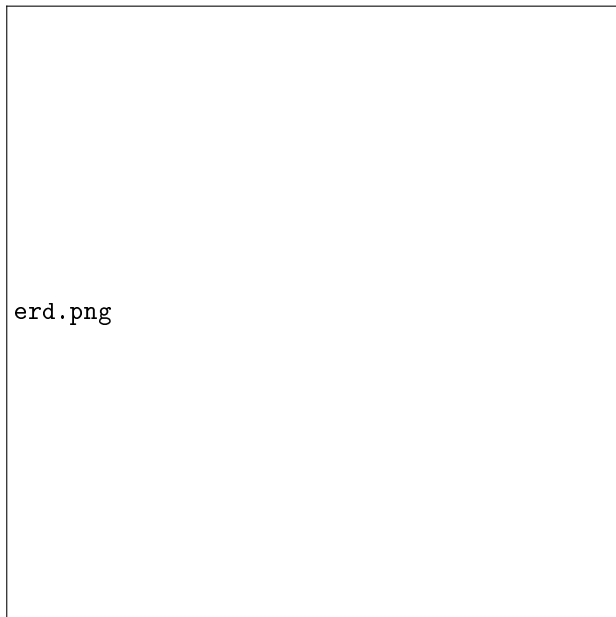


Figure 16: Halaman Desain database

Infrastruktur

Aplikasi Apache memiliki untuk meneruskan permintaan dari internet publik ke aplikasi web yang terdapat dalam Server yaitu aplikasi web berbasis Flask dengan bahasa pemrograman Python. Aplikasi web berbasis flask ini menangani seluruh permintaan yang datang dari pengguna dan mengirim respon yang sesuai dengan permintaan pengguna. Saat menjalankan tugasnya, aplikasi berbasis flask ini di-damping oleh beberapa aplikasi lainnya yaitu Celery dan MySQL. Aplikasi Celery digunakan oleh aplikasi web berbasis Flask untuk memindahkan sebagian beban kerja yang besar dari aplikasi web berbasis flask yang ada. Dengan adanya pemindahan beban dari aplikasi web berbasis flask ke aplikasi Celery ini, aplikasi web berbasis flask dapat melayani banyak permintaan pengguna yang masuk. Dalam menjalankan tugasnya, aplikasi Celery dibantu oleh aplikasi RabbitMQ yang bertugas sebagai *message broker*. Aplikasi MySQL merupakan aplikasi basis data yang digunakan dalam aplikasi web berbasis Flask. Aplikasi MySQL digunakan untuk menyimpan informasi dan mengambil data yang digunakan oleh aplikasi berbasis Flask.

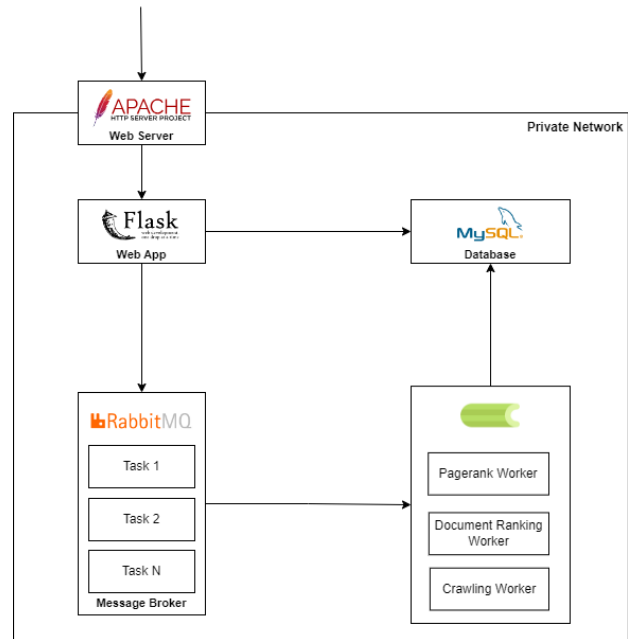


Figure 17: Infrastruktur search engine

Memory Profiling

Dilakukan pengujian penggunaan memori pada aplikasi *search engine*. Pengujian dilakukan dengan menggunakan pustaka *tracemalloc* dan *gc* dari bahasa pemrograman *python*. Pencatatan penggunaan memori aplikasi dilakukan setiap 30 menit dalam waktu lebih dari satu hari di lingkungan *server*. Pencatatan penggunaan memori dilakukan di dalam spesifikasi perangkat sebagai berikut:

1. 3 Gigabyte RAM
2. AMD CPU dengan 3 buah *thread*

Adapun grafik penggunaan memori aplikasi adalah sebagai berikut.

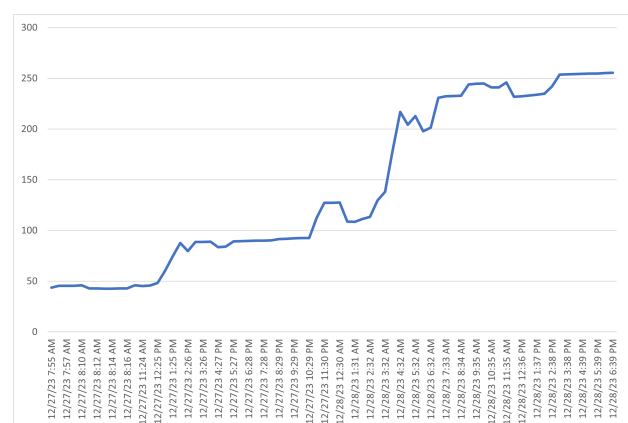


Figure 18: Grafik penggunaan memori aplikasi search engine

Percobaan lain dilakukan dengan waktu yang lebih singkat untuk mengidentifikasi letak kenaikan penggunaan memori oleh aplikasi. Dari percobaan yang

dilakukan, terdapat kenaikan penggunaan memori yang signifikan dari pustaka *urllib* yang berasal dari bahasa pemrograman *python*. Dari gambar 18, Dapat disimpulkan kenaikan penggunaan memori terbanyak dari aplikasi disebabkan oleh pustaka dari bahasa pemrograman *python* yang digunakan. Selain itu, untuk peringkat kedua dan ketiga dalam hal penggunaan memori terbanyak berasal dari implementasi algoritma *breadth first search* dari *search engine* pada penelitian [5].

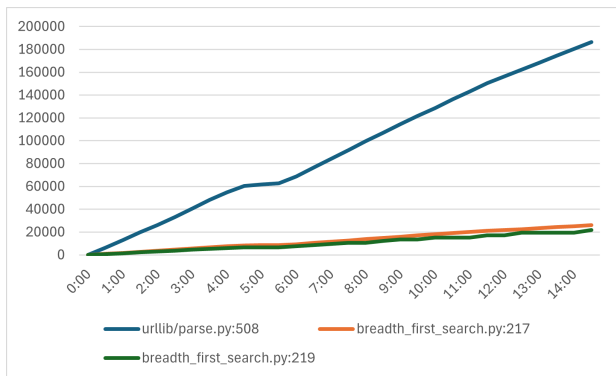


Figure 19: Grafik penggunaan memori terbanyak tiga baris kode aplikasi teratas

Unit Testing

Pengujian *unit testing* dilaksanakan dengan salah satu tim pengembang *backend* internal pada saat akhir *sprint*. Kesimpulan dari unit testing yang telah dilakukan adalah fitur yang telah dibuat dapat berjalan dengan baik.

Unit Testing			
Uji Fitur	Skenario Pengujian	Kesesuaian	
		Sesuai	Tidak Sesuai
Pencarian Pengguna	Pada tampilan halaman pencarian, ketika pengguna memasukkan kata kunci pencarian dan menekan tombol <i>enter</i> maka pengguna akan dialihkan ke halaman hasil pencarian	✓	

Hasil Pencarian	Saat menekan <i>tab</i> peta situs atau <i>sitemap</i> maka akan ditampilkan peta situs dari kata kunci yang sedang dicari	✓	
	Pengguna dapat memasukkan kata kunci pencarian dengan memasukkan <i>query</i> dengan kunci <i>search</i> untuk mendapatkan hasil pencarian	✓	
Peta Situs	Pengguna dapat memasukkan kata kunci ulang dengan memasukkan kata kunci pada <i>text field</i> di pojok kanan atas lalu menekan tombol <i>enter</i>	✓	
	Pengguna dapat mem- <i>filter</i> peta situs berdasarkan negara dengan memilih negara pada tombol <i>Select Countries</i> di pojok kanan atas	✓	

	Pengguna dapat memasukkan kata kunci pencarian dengan memasukkan kata kunci pencarian pada <i>query</i> URL dengan kunci <i>query</i> dan negara dengan kunci <i>countries</i>	✓	
Login	Ketika pengguna memasukkan informasi akun dengan benar ke dalam formulir yang ada maka pengguna akan dialihkan ke halaman <i>dashboard</i>	✓	
	Ketika pengguna memasukkan informasi akun dengan salah ke dalam formulir yang ada maka pengguna akan diberi pesan bahwa pengguna memasukkan informasi akun yang salah	✓	

Dashboard	Ketika pengguna menekan tombol <i>profile</i> maka pengguna akan disajikan <i>popup</i> yang berisi aksi <i>logout</i>	✓	
Page Ranking	Ketika pengguna menekan tombol <i>start</i> , maka <i>status page ranking</i> akan berubah menjadi <i>running</i> dan akan muncul tombol <i>stop</i>	✓	
	Ketika pengguna menekan tombol <i>stop</i> , maka <i>page ranking</i> akan berhenti dan tombol <i>start</i> akan muncul	✓	
Crawling	Ketika pengguna menekan tombol <i>start</i> , maka <i>status crawling</i> akan berubah menjadi <i>running</i> dan akan muncul tombol <i>stop</i>	✓	

	Ketika pengguna menekan tombol <i>stop</i> , maka <i>crawling</i> akan berhenti dan tombol <i>start</i> akan muncul	✓	
	Saat <i>tab domains</i> diklik, pengguna akan dialihkan ke sub halaman daftar domain	✓	
	Saat <i>tab webpages</i> diklik, pengguna akan dialihkan ke sub halaman daftar <i>webpages</i>	✓	
Daftar Situs	Ketika pengguna menekan salah satu situs pada daftar situs, pengguna akan dialihkan ke halaman detail situs	✓	
	Pengguna dapat menyaring daftar <i>situs</i> yang dimunculkan dengan mengaplikasikan <i>filter</i> yang tersedia yang terletak di atas tabel	✓	

Daftar <i>Domain</i>	Pengguna dapat menyaring daftar <i>domain</i> yang dimunculkan dengan mengaplikasikan <i>filter</i> yang tersedia yang terletak di atas tabel	✓	
Document Ranking	Ketika pengguna menekan tombol <i>start</i> , maka <i>status document ranking</i> akan berubah menjadi <i>running</i> dan akan muncul tombol <i>stop</i>	✓	
	Ketika pengguna menekan tombol <i>stop</i> , maka <i>document ranking</i> akan berhenti dan tombol <i>start</i> akan muncul	✓	
	Ketika pengguna menekan <i>tab words</i> , maka akan dialihkan ke sub halaman <i>words</i>	✓	
	Ketika pengguna menekan <i>tab search log</i> , maka akan dialihkan ke sub halaman <i>search log</i>	✓	

Daftar Staff	Ketika tombol <i>create new staff</i> ditekan, maka akan dialihkan ke halaman tambah <i>staff</i> baru	✓	
	Ketika tombol <i>more</i> yang dilambungkan dengan ikon tiga titik vertikal ditekan, maka akan muncul <i>popup</i> yang berisi aksi <i>delete staff</i> .	✓	
	Ketika <i>delete staff</i> ditekan, maka <i>staff</i> akan dihapus dari daftar <i>staff</i> setelah melakukan <i>refresh</i> ulang.	✓	
Tambah Staff	Ketika form berhasil diisi maka <i>staff</i> baru akan ditambahkan ke daftar <i>staff</i>	✓	

Kesimpulan

Berdasarkan hasil implementasi dan pengujian pada program ini, maka didapatkan kesimpulan sebagai berikut:

1. Perancangan aplikasi *admin console* untuk manajemen dan visualisasi data hasil pengindeksan *search engine* yang telah dibuat pada penelitian [5] yang dirancang menggunakan metode *scrum*.
2. Pada pengujian penggunaan memori aplikasi, diketahui adanya kenaikan penggunaan memori secara signifikan. Dalam pengujian terpisah, dilakukan pengujian untuk menemukan letak baris kode yang menggunakan memori terbanyak yang dapat dilihat pada gambar 18. Pada peringkat pertama, baris kode yang menggunakan memori terbanyak dan memiliki kenaikan

penggunaan memori yang signifikan berasal dari pustaka bawaan bahasa pemrograman python itu sendiri. Selain itu, baris kode peringkat kedua dan ketiga dalam hal penggunaan memori terbanyak berasal dari implementasi *search engine* pada penelitian [5].

References

- [1] P. Adi. Scrum method implementation in a software development project management. *International Journal of Advanced Computer Science and Applications*, 6, 09 2015.
- [2] S. Ajagbe, M. Oladipupo, and B. Emmanuel. Crime belt monitoring via data visualization: A case study of folium. 4:35–44, 06 2020.
- [3] O. Alonso and R. Baeza-Yates. A model and software architecture for search results visualization on the www. pages 8 – 16, 02 2000.
- [4] E. G. Caldarola, A. Picariello, A. Rinaldi, and M. Sacco. Exploration and visualization of big graphs - the dbpedia case study. 11 2016.
- [5] L. Khatulistiwa. Perancangan arsitektur search engine dengan mengintegrasikan web crawler, algoritma page ranking, dan document ranking. *Program Studi Ilmu Komputer Fakultas Matematika Dan Ilmu Pengetahuan Alam Universitas Negeri Jakarta* 2022, 2022.
- [6] A. S. M. Agus Muhyidin, Muhammad Afif Sulhan. Perancangan ui/ux aplikasi my cic layanan informasi akademik mahasiswa menggunakan aplikasi figma. 1 2020.