

Perancangan *User Interface Search Engine* Dengan *Admin Console* Untuk Manajemen Dan Visualisasi Data Hasil Pengindeksan *Search Engine*

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.

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.

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*.

Kajian Teori

Search Engine

Search Engine merupakan sebuah perangkat lunak yang dirancang untuk mencari informasi dalam internet. Dalam penggunaannya, pengguna memasukkan kata kunci yang ingin pengguna cari dalam *search engine* dan *search engine* akan menampilkan daftar dokumen, suara, gambar, video dan lain lain yang relevan dengan kata kunci yang pengguna masukkan.

Search engine menggunakan sebuah perangkat lunak bernama *crawler* yang bertugas untuk memindai dan meng-*index* halaman *web* yang berada di *internet*. *Crawler* ini mengikuti tautan dari satu halaman ke halaman lainnya mengumpulkan informasi mengenai konten dan struktur website. Data yang berhasil dikumpulkan tersebut lalu disimpan dalam sebuah database yang membentuk *search engine index*. Beberapa contoh *search engine* yang populer pada saat ini adalah Google, Yahoo, Bing, Baidu dan Yandex.

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 mengerjakan nya dan membantu satu sama lain dalam membangunnya 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].

Javascript

Javascript adalah bahasa pemrograman tingkat tinggi dan *interpreted* yang digunakan untuk menambahkan interaktivitas dan sifat dinamis dalam *website*. Dikembangkan oleh Brendan Eich pada tahun 1995, Javascript menjadi populer karena kemampuannya membuat elemen interaktif dan merespons aksi pengguna. Awalnya bernama "Mocha" dan "Livescript," kemudian diubah menjadi "Javascript" untuk keperluan pemasaran.

Javascript membedakan diri dari Java dan menjadi standar *de facto* untuk *client-side scripting* di *website*. Pengenalan fitur baru dan teknologi seperti AJAX pada tahun 2000 memperluas kemampuan Javascript, memungkinkannya melakukan komunikasi asynchronous dengan server tanpa perlu memuat ulang halaman *website*.

Selama bertahun-tahun, Javascript terus berkembang dengan adanya NodeJS yang memungkinkan eksekusi Javascript di luar *browser*. Hal ini membuka peluang penggunaan Javascript dalam *server-side scripting*, *command-line tools*, dan pembuatan aplikasi. Pengenalan berbagai framework dan library seperti Angular, React, dan VueJS juga memberikan

kemudahan bagi pengembang dalam membuat aplikasi *web* kompleks.

User Interface

User Interface adalah ilmu tentang tata letak grafis suatu web atau aplikasi [6]. Cakupan *User Interface* adalah tombol yang akan diklik oleh pengguna, teks, gambar, text entry fields, dan semua item yang berinteraksi dengan pengguna. Termasuk layout, animasi, transisi, dan semua interaksi kecil. UI mendesain semua elemen visual, bagaimana pengguna berinteraksi dengan halaman web dan apa yang ditampilkan di halaman web. Elemen visual yang ditangani oleh seorang desainer UI adalah skema warna, menentukan bentuk tombol, serta menentukan jenis font yang digunakan untuk teks. Desainer UI harus bisa membuat tampilan bagus yang akan meningkatkan kesetiaan pengguna.

Metodologi

Perancangan 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.

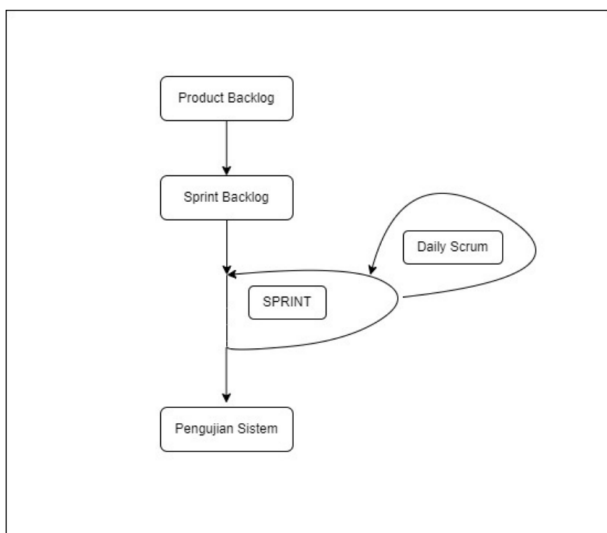


Figure 1: Tahapan penelitian dengan menggunakan metode *scrum*

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 projek	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.

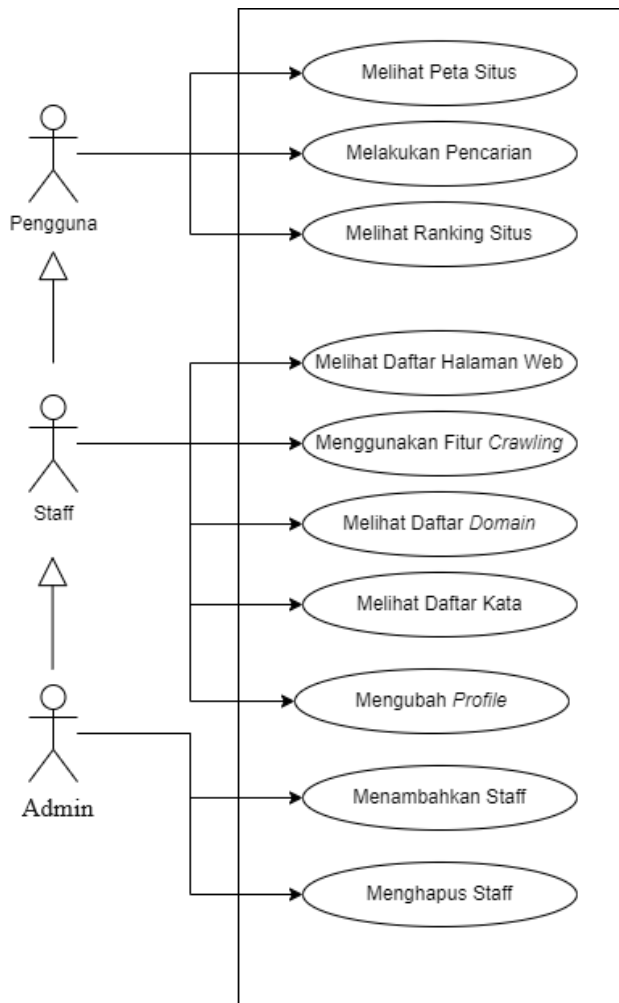


Figure 2: *Use case diagram*

Hasil dan Pembahasan

Implementasi *User Interface*

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

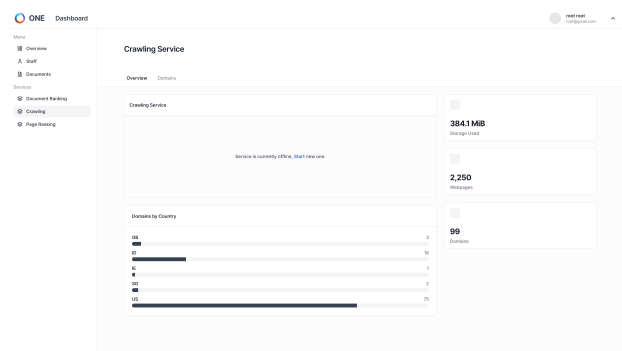


Figure 4: *Halaman Service Crawling*

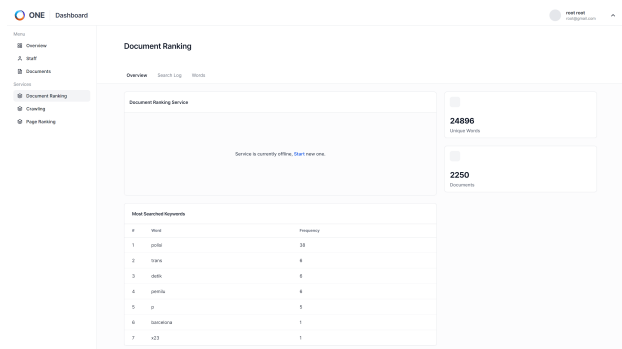


Figure 5: *Halaman Document Ranking*

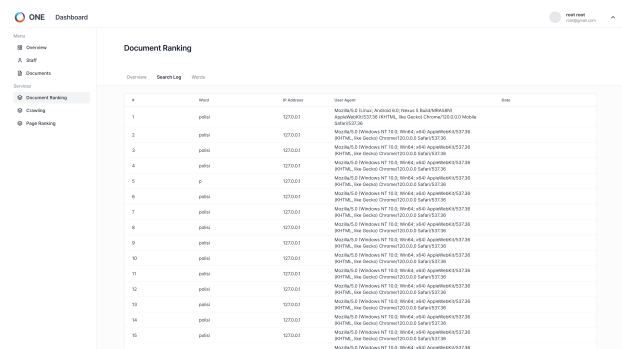


Figure 6: *Halaman search log*

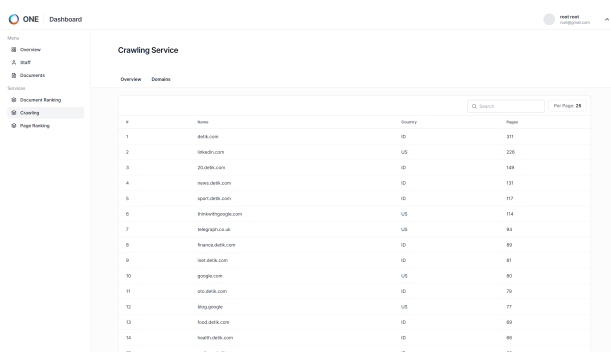


Figure 3: *Halaman Daftar Domain*

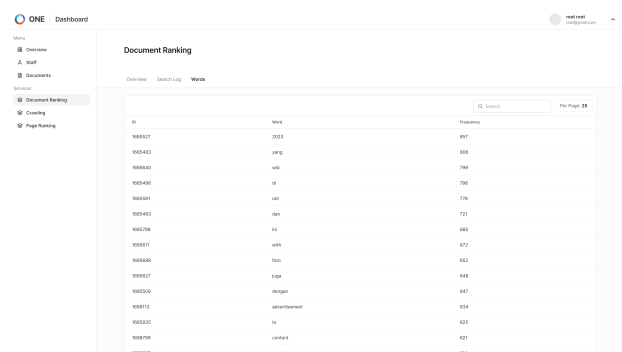


Figure 7: *Halaman words*

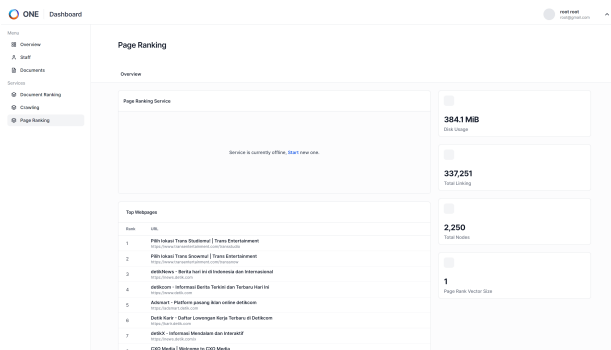


Figure 8: Halaman page ranking

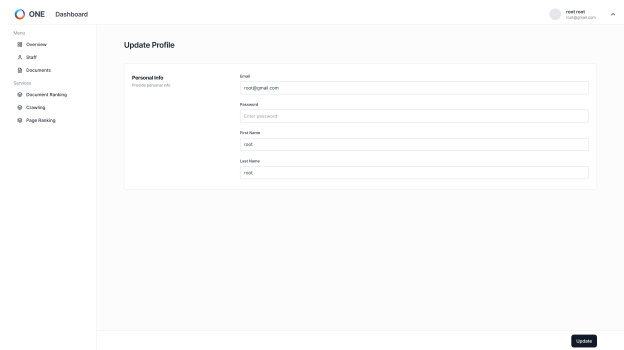


Figure 12: Halaman ubah profile

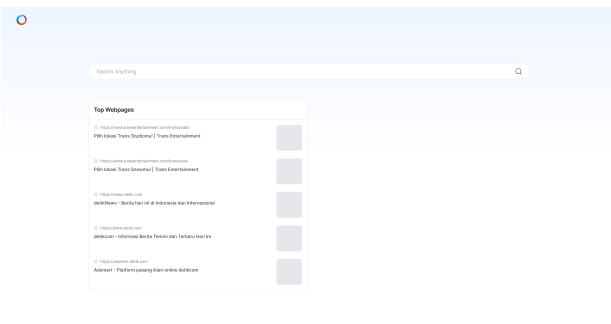


Figure 9: Halaman pencarian

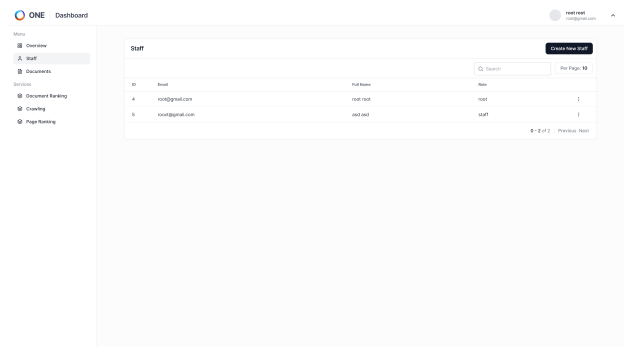


Figure 13: Halaman daftar staff

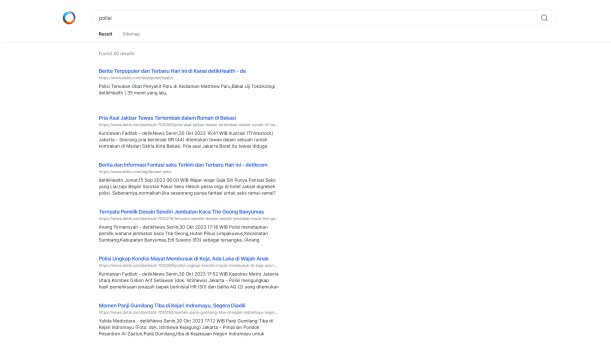


Figure 10: Halaman hasil pencarian

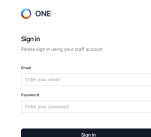


Figure 14: Halaman login staff

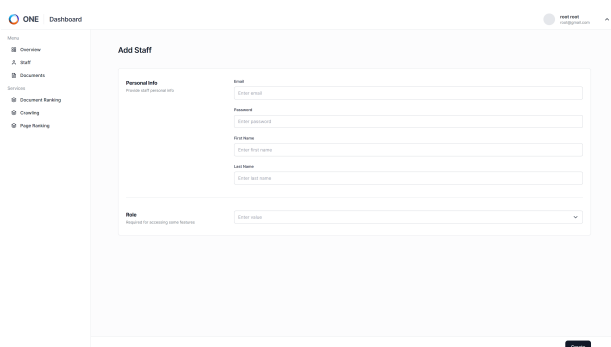


Figure 11: Halaman tambah staff

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 didampingi 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.

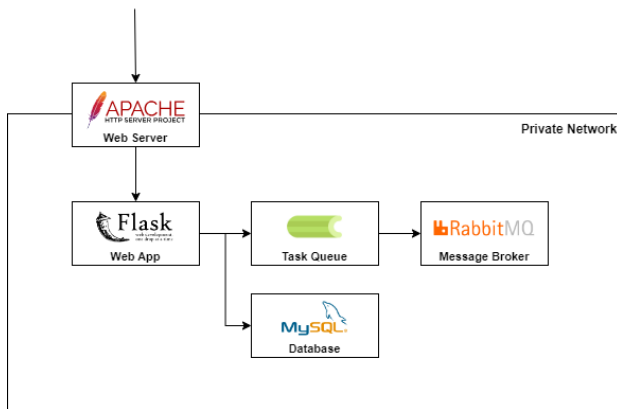


Figure 15: Infrastruktur search engine

Memory Profiling

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

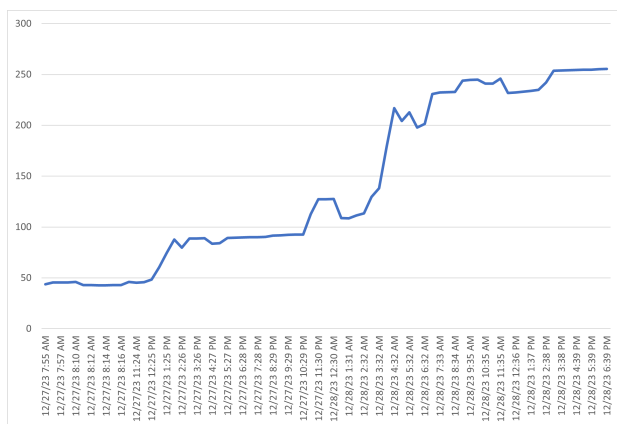


Figure 16: Grafik penggunaan memori aplikasi search engine

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 17, 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].

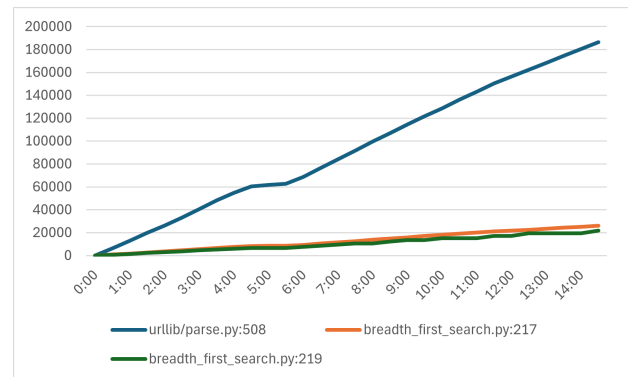


Figure 17: Grafik penggunaan memori terbanyak tiga baris kode aplikasi teratas

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 17. 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.