IMPLEMENTASI *ELASTICSEARCH* UNTUK PENCARIAN DAN MENENTUKAN SKOR *SIMILARITY* PADA PROPOSAL SKRIPSI DI FAKULTAS TEKNOLOGI INFORMASI UNIVERSITAS BALE BANDUNG

Maesuri Fauziah[#1], Yudi Herdiana, S.T, M.T [*2], Nurul Imamah, S.T., M.T [#3]

[#]*Fakultas Teknologi Informasi, Universitas Bale Bandung*
Jl. Raden AA Wiranatakusumah No.7, Baleendah, Kec. Baleendah, Bandung, Jawa Barat 40375

[1]maesurifauziah3@gmail.com

[2]yudiherdiana@unibba.ac.id

[3]nurulimamah@unibba.ac.id

Abstract - ***The development of Information Technology nowdays is so rapid that make us thirst for the speed, efficiency and accuracy. In a college or university, the data of papers or theses from previous years students can be useful for students who need scientific references in composing paper or thesis. Also, before start to compose the paper or thesis, students must submit a proposal related to the title that has been taken. Therefore, the submitted proposal must be the result of the author's thought. Before starting the work it is necessary to determine the similarity of the thesis proposal to the previous paper or thesis document must be known. With Elasticsearch, it can help users in searching paper or thesis documents, because Elasticsearch is a full text search engine which open source and as a data analysis tool that can also be scaled (scalable). Moreover, Elasticsearch also was developed with the Java programming language powered by Apache Lucene which is also a search engine database that has a querylow level. Elasticsearch has an easier query to use because it uses the RESTful web service and is also one of the databases that enter the NoSQL world.***

Keywords - ***Elasticsearch, web service, Thesis, Full Text, RESTful, Java, NoSQL, Database, Proposal, Similarity.***

Abstrak – **Perkembangan Teknologi Informasi saat ini begitu pesat sehingga menyebabkan kita haus akan kecepatan, efisiensi dan akurasi yang tepat. Pada suatu perguruan tinggi data-data skripsi dari mahasiswa tahun-tahun sebelumnya dapat bermanfaat bagi mahasiswa yang membutuhkan referensi ilmiah dalam menyusun skripsi dan juga sebelum memulai penyusunan skripsi harus mengajukan proposal terkait judul yang akan diambil maka dari itu sebelumnya proposal yang diajukan haruslah hasil pemikiran penulis. Sebelum memulai pengerjaan perlu adanya penentuan kemiripan (*similarity) proposal* skripsi terhadap dokumen skripsi harus diketahui. Dengan adanya *Elasticsearch* dapat membantu pengguna dalam pencarian dokumen skripsi, karena *Elasticsearch* merupakan mesin pencari *full text* yang *open source* dan sebagai alat analisis data juga dapat di skalakan (*scalable*). Selain itu *Elasticsearch* juga *Elasticsearch* di kembangkan dengan bahasa pemrograman Java yang ditenagai oleh Apache Lucene yang juga merupakan *search engine database* yang memiliki *querylow level*. *Elasticsearch* memiliki *query* yang lebih mudah untuk digunakan karena menggunakan *web service* RESTful dan juga merupakan salah satu *database* yang masuk ke dunia NoSQL..**

Kata kunci – ***Elasticsearch, web service,* Skripsi*, Full Text, RESTful, Java, NoSQL, Database,* Proposal*, Similarity.***

I. PENDAHULUAN

Skripsi adalah istilah yang digunakan di Indonesia untuk mengilustrasikan suatu karya tulis ilmiah berupa paparan tulisan hasil penelitian sarjana S1 yang membahas suatu permasalahan/fenomena dalam bidang ilmu tertentu dengan menggunakan kaidah-kaidah yang berlaku. Skripsi merupakan persyaratan untuk mendapatkan status sarjana (S1) di setiap Perguruan Tinggi Negeri (PTN) maupun Perguruan Tinggi Swasta (PTS) yang ada di Indonesia.

Fakultas Teknologi Informasi adalah salah satu Fakultas di Universitas Bale Bandung yang sebelum mahasiswa memulai penyusunan skripsi tentunya mahasiswa diharapkan untuk mencari referensi-referensi ilmiah terkait topik atau judul, yang nantinya akan dijadikan sebagai bahan acuan pembuatan judul skripsi.

Pencarian skripsi sebagai referensi ilmiah secara langsung datang ke perpustakaan lalu menuju ke rak penyimpanan skripsi yang tersimpan dalam bentuk cetak atau hardcopy, dalam pencarian menelaah satu-persatu judul skripsi, untuk menemukan judul yang sesuai memerlukan waktu yang lama. Mahasiswa juga mencari tahu apakah penelitiannya sudah pernah dilakukan atau tidak. Proses awal saat memulai penyusunan skripsi sering ditemukan mahasiswa yang melakukan

pengambilan pustaka tanpa menyertakan sumbernya yang mana akan dianggap sebagai plagiat atau pelanggaran hak cipta dari suatu penelitian atau skripsi yang telah dibuat. Tentunya mahasiswa harus mengetahui bahwa topik yang diambil haruslah murni hasil pemikiran dari mahasiswa yang bersangkutan dan tidak merupakan sebuah plagiat dari penelitian orang lain.

Berdasarkan pemaparan di atas maka perlu diterapkan similarity dengan menggunakan elasticsearch sebagai alternatif dengan harapan diterapkannya elasticsearch ini jauh lebih baik dibandingkan dengan menggunakan konsep dengan database pada umumnya. Similarity atau kemiripan dapat dilakukan dengan pencarian fulltext dengan menggunakan elasticsearch. Keluaran dari elasticsearch berupa dokumen yang sudah diberikan skor yang didapat untuk menemukan nilai kemiripan dokumen dan diurutkan berdasarkan relevansi dokumen dengan kata kunci yang diinputkan pengguna.

Sebuah aplikasi yang menerapkan elasticsearch untuk mempermudah membangun aplikasi pencarian topik atau judul skripsi dan pendeteksi similarity dokumen proposal skripsi . Oleh karena itu judul yang diambil adalah "Implementasi Elasticsearch Untuk Pencarian Dan Menentukan Similarity Pada Proposal Skripsi Di Fakultas Teknologi Informasi".

II. LANDASAN TEORI DAN METODE

A. *Landasan Teori*

1. *Elasticsearch*

Elasticsearch adalah mesin pencari *fulltext* yang *open source* dan alat analisis data yang dikembangkan dengan bahasa pemrograman Java, berbasiskan *Apache Lucene*, dan dapat diskalakan. (Hüseyin Akdoğan, 2015). *Elasticsearch* memanupilasi data dengan menggunakan *REST API* yang dapat digunakan untuk berbagai tugas, yaitu mengelola indeks, mengubah parameter instance, memeriksa node dan status cluster, indeks data, cari data, atau ambil dokumen melalui GET API yang berkonsentrasi pada penggunaan bagian CRUD (*create-retve-update-delete*) dari API, yang memungkinkan untuk menggunakan Elasticsearch dengan cara yang mirip dengan menggunakan database *NoSQL* (Rafał Kuć dan Marek Rogoziński, 2013).

2. Web Service

Web service adalah aplikasi sekumpulan data (*database*), perangkat lunak (*software*) atau bagian dari perangkat lunak yang dapat diakses secara *remote* oleh berbagai piranti dengan sebuah perantara tertentu. Secara umum, *web service* dapat diidentifikasikan dengan menggunakan URL

3. Search Engine

Search Engine adalah program komputer yang dirancang untuk melakukan pencarian atas berkas-berkas yang tersimpan dalam layanan www, ftp, publikasi milis, ataupun *news group* dalam sebuah ataupun sejumlah komputer dalam suatu jaringan.

4. Apache Lucene

Lucene adalah pencarian teks berfitur lengkap. Ini berarti, cukup sederhana: suatu program mencari serangkaian dokumen teks untuk satu atau lebih istilah yang telah ditentukan pengguna.

5. NoSQL

NoSQL adalah istilah yang dikenal dalam teknologi komputasi untuk merujuk kepada kelas yang luas dari sistem manajemen basis data yang di identifikasikan dengan tidak mematuhi aturan pada model sistem manajemen basis data relasional yang banyak digunakan. *NoSQL* dibuat dengan tujuan khusus untuk model data spesifik dan memiliki skema fleksibel untuk membuat aplikasi modern.

6. REST (*REpresentational State Transfer*)

REST (*REpresentational State Transfer*) merupakan standar arsitektur komunikasi berbasis web yang sering diterapkan dalam pengembangan layanan berbasis web. Umumnya menggunakan HTTP (*Hypertext Transfer Protocol*) sebagai protocol untuk komunikasi data. REST pertama kali diperkenalkan oleh Roy Fielding pada tahun 2000. Pada umumnya formatnya menggunakan JSON dan XML.

7. Java

Java adalah bahasa pemrograman dan platform komputasi pertama kali dirilis oleh *Sun Microsystems* pada tahun 1995. Pengguna aplikasi biasanya menggunakan Java RuntimeEnvironment (JRE) diinstal pada mesin mereka sendiri untuk menjalankan aplikasi Java, atau dalam *browser web* untuk applet Java. Untuk pembuatan dan pengembangan aplikasi berbasis Java diperlukan *Java Development Kit* (JDK), dimana saat ini pemilik lisensi dari JDK adalah *Oracle Corporation* yang telah secara resmi mengakuisisi *Sun Microsystem* pada awal tahun 2010.

8. Java Script Object Notation (JSON)

Java Script Object Notation (JSON) adalah sebuah format data yang memungkinkan aplikasi untuk saling berkomunikasi di dalam sebuah jaringan, yang melalui RESTful API. JSON menyediakan semua bahasa pemrograman modern dengan bantuan untuk menghasilkan dan menerima data (Marss, 2017).

9. Similarity

Fungsi Kemiripan atau *similarity* adalah menghitung kesamaan dan ketidak samaan antara dua objek yang diobservasi. Objek yang di maksud disini adalah komunitas yang saling berbeda. Ludwig & Reynolds (1988) menyatakan bahwa kemiripan suatu komunitas dengan komunitas lain dapat dinyatakan dengan *similarity coefficients. Similarity coefficients* memiliki nilai yang

bervariasi antara 0 (jika komunitas benar-benar berbeda) hingga 1 (jika kedua komunitas identik).

10. Apace Maven

Apache Maven adalah manajemen proyek perangkat lunak dan alat pemahaman. Berdasarkan konsep model objek proyek (POM), *Maven* dapat mengelola pembangunan, pelaporan, dan dokumentasi proyek dari informasi utama.

11. Kibana

Kibana adalah *plugin* visualisasi data *open source* untuk *Elasticsearch*. Ini memberikan kemampuan visualisasi di atas konten yang diindeks pada *cluster Elasticsearch*. Pengguna dapat membuat bar, garis, dan sebaran plot, atau diagram lingkaran dan peta di atas volume data yang besar.

12. Postman

Postman adalah sebuah aplikasi HTTP client yang merupakan plugin dari browser Chrome. Fungsi Postman adalah untuk pengecekan web service. Postman dapat menampilkan hasil dari HTTP request yang kompleks sekalipun dengan cepat. (Alifa dan Alief, 2015).

13. Model Driven Development (MDD)

Teknik pengembangan berbasis model *Model Driven Development (MDD)* menekankan gambar model untuk membantu memvisualisasikan dan menganalisis masalah, mendefinisikan kebutuhan bisnis, dan merancang sistem informasi. Analisis dan desain sistem terstruktur - berpusat pada proses Teknik informasi (IE) - berpusat pada data Analisis dan desain berorientasi obyek (OOAD) - terpusat pada objek (integrasi data dan masalah proses) *Rute model driven development*.

14. Flowmap

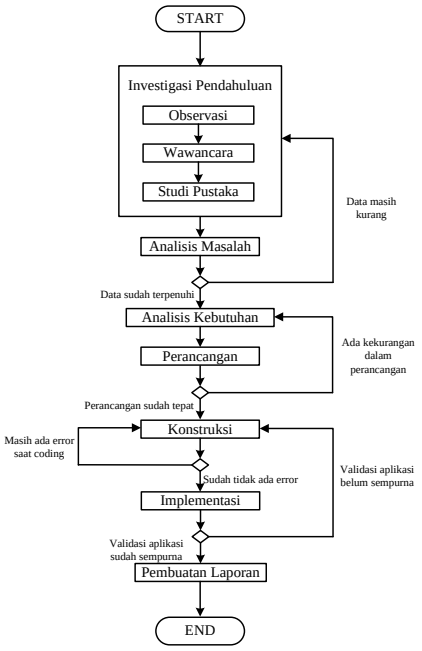
Flowmap atau bagan alir adalah bagan yang menunjukan aliran di dalam program atau prosedur sistem secara logika. Flowmap ini berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu. Pembuatan flowmap ini harus dapat memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi.

B. Metode

Metodologi penelitian dalam penelitian ini melalu berbagai tahapan yaitu metode pengumpulan data dan metode pengembangan sistem. Metode pengumpulan data meliputi yang meliputi observasi, wawancara, dan studi pustaka, sedangkan dalam metode pengembanngan sistem menggunakan *Model Driven Developmen.* Berdasarkan penelitian ini berikut adalah penjelasan dari kerangka berfikir dan juga perancangan aplikasi.

1. Kerangka Berfikir

Berikut ini adalah langkah-langkah yang di lakukan untuk mencapai tujuan dari penelitian, ditunjukan dengan gambar dibawan ini:



Gambar. 1 Kerangka Berfikir

III. PEKERJAAN DAN DISKUSI HASIL

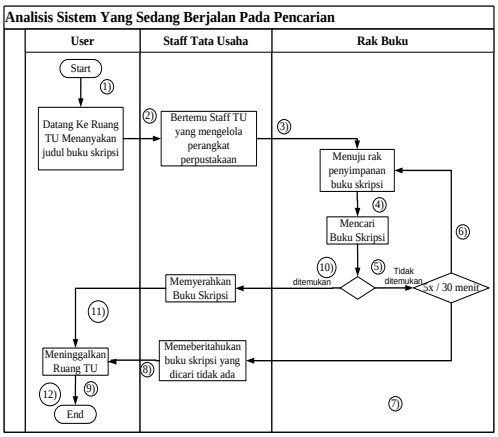
A. Proses Pekerjaan

1. Analisis Sistem

Analisis system ini bertujuan untuk membandingkan sistem yang sedang berjalan saat ini dan sistem usulan yang akan dilakukan dalam penelitian ini.

a. Analisis Sistem Yang Sedang Berjalan

Sistem yang sedang berjalan saat ini yaitu mencari dokumen dengan ara manual yaitu dengan datang langsung ke ruang Tata Usaha dan nantinya bertemu dengan Staff Tata Usaha yang mengelola perangkat perpustakaan Lalu mencari langsung di rak penyimpanan buku skripsi.

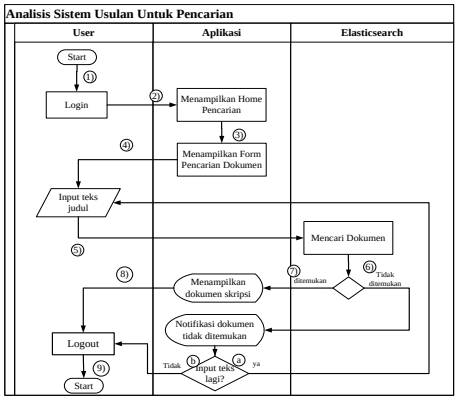


Gambar. 2 Analisis Sistem Yang Sedang Berjalan

b. Analisis Sistem Usulan

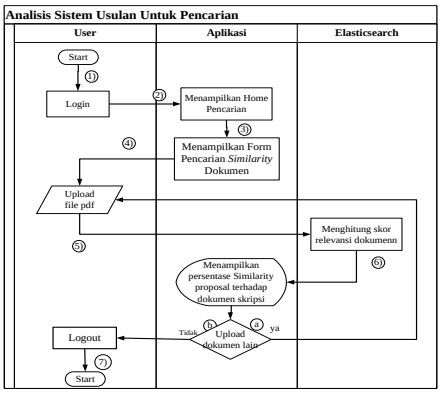
Analisis Usulan dilakukan untuk memberikan usulan dalam analisis sistem ini agar didapatkannya sistem yang dapat membantu terkait permasalahan yang ada. Pada Analisis Sistem Usulan ini tentunya mempunyai kelebihan yaitu dengan memakai aplikasi yang dapat memudahkam pengguna dalam melakukan pencarian dan menentukan persentase *similarity*.

1) Analisis Sistem Usulan Untuk Pencarian



Gambar. 3 Analisis Sistem Usulan Untuk Pencarian

2) Analisis Sistem Usulan Untuk *Similarity*



Gambar. 4 Analisis Sistem Usulan Untuk *Similarity*

2. Hasil Analisis

Hasil analisis dilakukan dengan membuat tabel dari beberapa kemungkinan *similarity* dari proposal skripsi terhadap skripsi yang sudah ada. Berikut ini adalah tabel kemungkinan yang telah dibuat:

Tabel 1 Hasil Analisis

Autor	Title	Publiser	Abstrak	Full Text	Keterangan
Y	Y	Y	Y	Y	*Similarity* Pasti
Y	Y	Y	Y	TP	*Similarity* Pasti
T	Y	T	T	T	*Similarity* Pasti
Y	Y	Y	T	T	*Kemungkinan*
T	T	T	Y	T	*Kemungkinan*
T	T	T	T	Y	*Tidak Mungkin*

Ket:

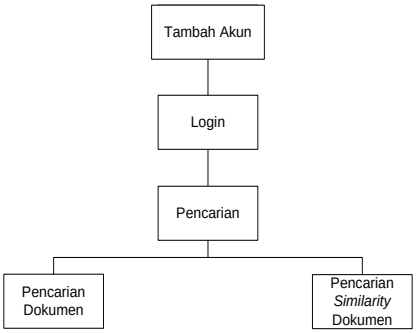
Y = Sama; T = Tidak sama; TP = Tidak Perlu

3. Perancangan

Proses perancangan aplikasi ini menggunakan Unified Modeling Language (UML) untuk pemodelannya, dan untuk memberikan gambaran pada penelitian ini yang meliputi Struktur Menu, Use Case Diagram, Activity Diagram Sequence Diagram.

a. Struktur Menu

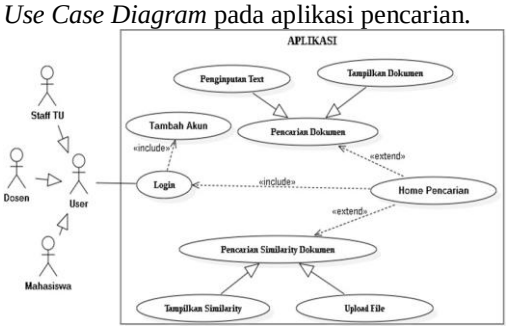
Struktur menu disusun secara bertingkat dan dibuat untuk memudahkan pengoperasian dalam aplikasi. Berikut adalah struktur menu dari Analisis Sistem Usulan.



Gambar. 2 Struktur Menu

b. Use Case

Use Case Diagram bertujuan untuk memberikan gambaran interaksi aktor dengan sistem aplikasi pencarian. Berikut ini adalah *Use Case Diagram* pada aplikasi pencarian.

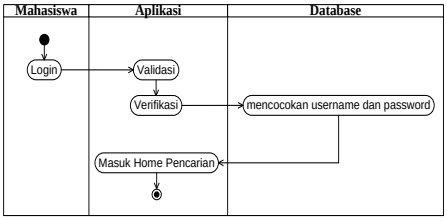


Gambar. 3 *Use Case*

c. Activity Diagram

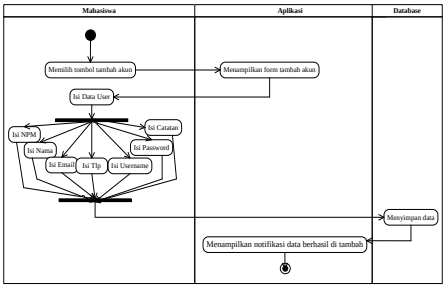
Activity Diagram merupakan cara memodelkan aktifitas yang ada dalam suatu *use case* yang meliputi *Activity Diagram* Login, *Activity Diagram* Tambah Akun, *Activity Diagram* Home Pencarian, *Activity Diagram* Pencarian Dokumen dan *Activity Diagram* Pencarian Similarity Dokumen.

1) *Activity Diagram Login*



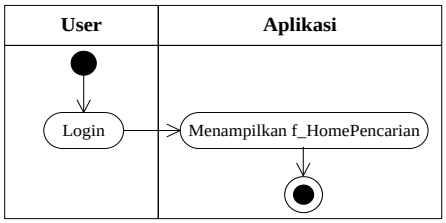
Gambar. 4 *Activity Diagram Login*

2) *Activity Diagram* Tambah Akun



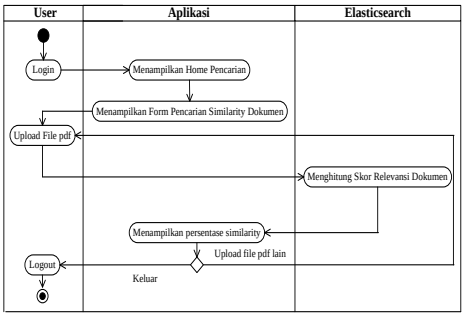
Gambar. 5 *Activity Diagram* Tambah Akun

3) *Activity Diagram* Home Pencarian



Gambar. 6 *Activity Diagram* Home Pencarian

4) *Activity Diagram* Pencarian Similarity Dokumen

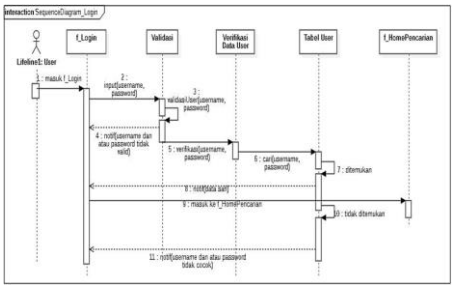


Gambar. 7 *Activity Diagram* Pencarian Similarity Dokumen

d. Sequence Diagram

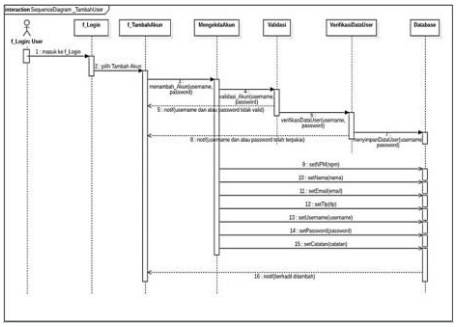
Sequence Diagram merupakan diagram yang menggambarkan interaksi antar objek di sekitar aplikasi yang meliputi *Sequence Diagram Login, Sequence Diagram* Tambah Akun Baru, *Sequence Diagram* Pencarian Dokumen dan *Sequence Diagram* Pencarian *Similarity* Dokumen

1) *Sequence Diagram Login*



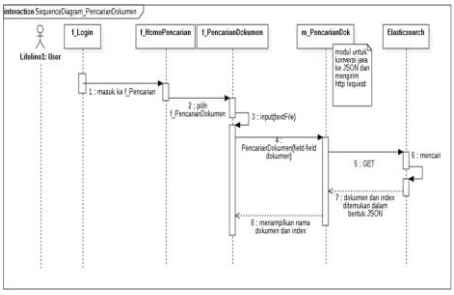
Gambar 8 *Sequence Diagram* Login

2) *Sequence Diagram* Tambah Akun



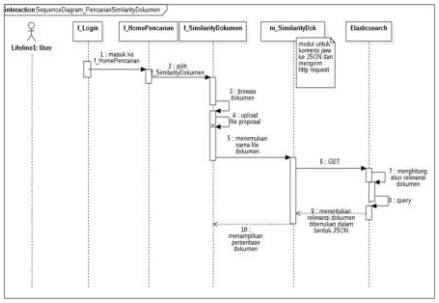
Gambar 9 *Sequence Diagram* Tambah Akun

3) *Sequence Diagram* Pencarian Dokumen



Gambar 10 *Sequence Diagram* Pencarian Dokumen

4) *Sequence Diagram* Pencarian *Similarity* Dokumen



Gambar 11 *Sequence Diagram* Pencarian *Similarity* Dokumen

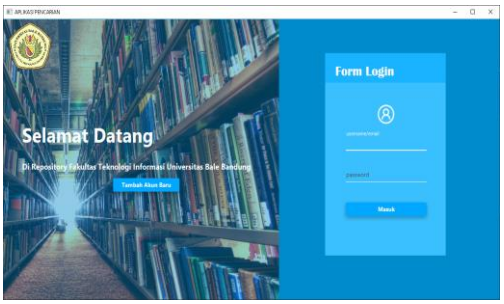
B. Hasil Pekerjaan

1. Implementasi Antar Muka

Implementasi antar muka merupakan perancangan yang telah dilakukan dan diketahui

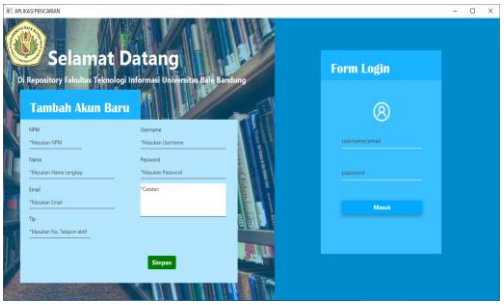
bahwa struktur menu bertujuan untuk memudahkan pengoperasian dalam aplikasi yang terdiri dari Tambah Akun, Login, Home Pencarian, Pencarian Dokumen, Pencarian Similarity Dokumen. Berikut ini adalah Tampilan Struktur Menu:

a. Login



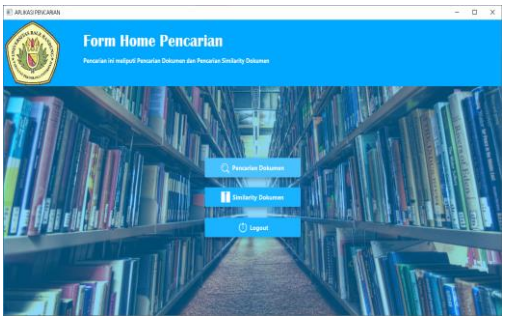
Gambar 12 Tampilan *Login*

b. Tambah Akun



Gambar 13 Tampilan Tambah Akun

c. Home Pencarian

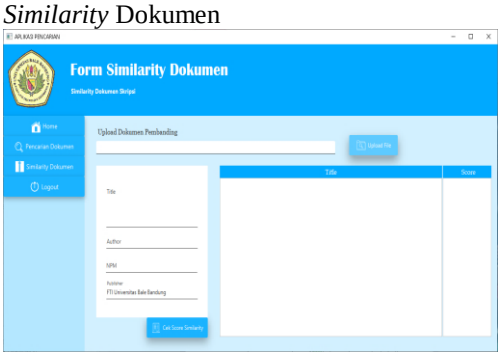


Gambar 14 Tampilan Home Pencarian

d. Pencarian Dokumen

Gambar 15 Tampilan Pencarian Dokumen

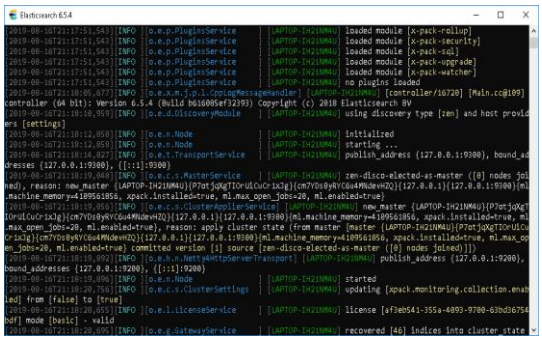
e. *Similarity* Dokumen



Gambar 16 Tampilan *Similarity* Dokumen

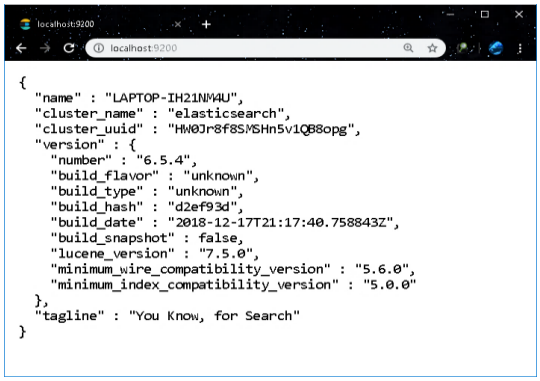
2. Implementasi *Elasticsearch*

Pertama *Elasticsearch* dijalankan untuk mengetahui keaktifan *Elasticsearch.* Berikut ini adalah tampilannya:



Gambar 16 Tampilan *Command Line Elasticsearch* Aktif

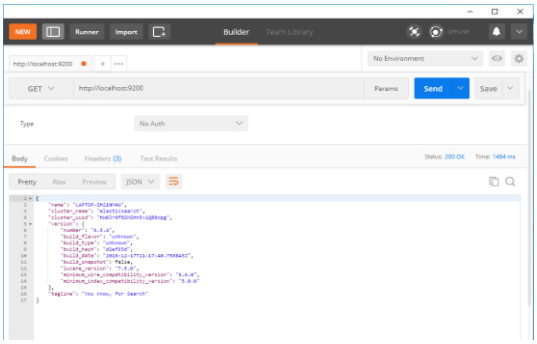
Gambar diatas menujukan bahwa *Elasticsearch* sudah aktif atau sudah siap untuk digunakan. Ditunjukan dengan munculnya kata "*started*" Berikut ini pengecekan untuk memastikan *Elasticsearch* sudah aktif:



Gambar 16 Tampilan Aktif *Elasticsearch* di *Browser*

Berdasarkan gambar diatas untuk memastikan bahwa *Elasticsearch* sudah aktif yaitu dengan cara mengetikan localhost:9200 pada *addres bar* pada browser lalu aka tampil JSON seperti yang ditunjukan pada gambar.

Berikut ini pengecekan untuk memastikan *Elasticsearch* sudah aktif melalui HTTP *Client Postman*:



Gambar 15 Tampilan Aktif *Elasticsearch* di *Postman*

Berdasarkan gambar diatas untuk memastikan bahwa *Elasticsearch* sudah aktif yaitu dengan cara melakukan HTTP *Request* GET localhost:9200 pada HTTP *Client* Postman lalu akan tampil JSON seperti yang ditunjukan pada gambar.

3. Kasus dan Hasil Pengujian

a. Kasus

Sebelum melakukan pengujian ada beberapa kasus, berikut ini adalah beberapa kasus:

- Untuk mengambil data dari server elasticsearch yaitu melakuakn http request ke elasticsearch, dengan uri yang sesuai dengan query untuk mencari berdasarkan filter yang di pilih.
- Ditemukannya format response dari http request adalah berbrntuk JSON sehingga harus dikonversi ke java object.
- Mengubah / parse JSON menjadi java object dengan library Jackson. Menemukan cara parsing JSON ke java object
- Dengan memakai library Jackson pada saat konversi lebih kompleks sehingga masih terjadi kesalahan/ error karna belum ditemukannya format yang tepat untuk membuat sintaks penulisan, dengan menggunakan library JSON Simple lebih sederhana
- Menampilkan data java object ke listview di form pencarian
- Memecahkan masalah pengkonversi JSON menjadi java object ke listview
- Menganalisis data hasil report hasil dari perintah http://localhost:9200/_cat/indices?v yang menghasilkan.
- Pengaturan clustering

b. Hasil Pengujian

Pengujian ini dilakukan secara *black box* dengan hanya memperhatikan masukan ke dalam sistem dan keluaran dari masukan tersebut, berikut ini pemaparan dari setiap nomor pengujian yang terdapat pada rencana pengujian:

Tabel 2 Pengujian Aplikasi

Pengujian Aplikasi				
No	**Item Uji**	**Skenario Uji**	**Hasil yang diharapkan**	**Hasil Pengujian**
1.	Login User	Mengisi *username* dan *password* dan menekan *button* Masuk	Jika username dan password benar maka akan muncul tampilan Form Home Pencarian	+Berhasil
		Menampilkan notifikasi saat mengisi *username* dan *password* yang salah.	Ketika memasukan username dan atau password yang salah akan muncul alert peringatan	+Berhasil
2.	Tambah Akun	Mengisi Npm, Nama, Email, Tlp, *Username*, *Password* dan Catatan dan menekan tombol Simpan	Ketika klik tombol simpan akan muncul alert pemberitahuan yang menyatakan data sudah tersimpan.	+Berhasil
		Melakukan Login di Form Tambah Akun	Ketika sudah menyimpan data di form tambah akun , dapat melakukan login di form yang sama dan telah di sediakan untuk melakukan login	+Berhasil
3.	Halaman Utama Pencarian	Menekan Tombol Pencarian Dokumen	Ketika menekan tombol pencarian dokumen maka akan di arahkan ke Form Pencarian Dokumen	+Berhasil
		Menekan Tombol Pencarian Similarity Dokumen	Ketika menekan tombol similarity dokumen maka akan di arahkan ke Form Similarity Dokumen.	+Berhasil
		Menekan Tombol Logout	Ketika menekan tombol logout akan keluar dari operasi pencarian dan akan di arahkan ke halaman awal saat mebuka aplikasi yaitu Form Login	+Berhasil
4.	Form Pencarian Dokumen	Mencari bersadarkan NPM	Akan menampilkan judul berdasarkan NPM yang di cari	+Berhasil
		Mencari bersadarkan Year	Akan menampilkan judul berdasarkan tahun (year) yang di cari	+Berhasil
		Mencari bersadarkan Author	Akan menampilkan judul berdasarkan Author yang di cari	+Berhasil
		Mencari bersadarkan Title	Akan menampilkan judul berdasarkan Title yang di cari	+Berhasil
		Mencari bersadarkan Keywords	Akan menampilkan judul berdasarkan Keywords yang di cari	+Berhasil

		Mencari bersadarkan IDDoc	Akan menampilkan judul berdasarkan IDDoc yang di cari	+Berhasil
		Mencari bersadarkan Publisher	Akan menampilkan judul berdasarkan Publisher yang di cari	+Berhasil
		Mencari bersadarkan Status	Akan menampilkan judul berdasarkan Status yang di cari	+Berhasil
		Mencari bersadarkan Mounth	Akan menampilkan judul berdasarkan Mounth yang di cari	+Berhasil
		Mencari bersadarkan Tags	Akan menampilkan judul berdasarkan Tags yang di cari	+Berhasil
		Klik Item hasil pencarian	Ketika klik salah satu item hasil pencarian maka akan muncul detail dari item yang di klik	+Berhasil
5.	Form *Similarity* Dokumen	Mengupload File Pdf	Ketika mengupload file, pdf akan di konversi menjadi teks	+Berhasil
		Menekan tombol Cek skor *Similarity*	Ketika menekan tombol cek skor similarity maka akan muncul title dan juga score	+Berhasil

IV. KESIMPULAN

Berdasarkan hasil analisis, pengumpulan data, dan perancangan maka hasil yang tercapai adalah menyelesaikan laporan dan Aplikasi Pencarian Dan Menentukan Similarity Pada Proposal Skripsi Di Fakultas Teknologi Informasi Universitas Bale Bandung, penulis dapat menyimpulkan bahwa:

1. Aplikasi ini dapat melakukan pencarian berdasarkan filter yang dipilih, dan berbeda dari yang sebelumnya masih melakukan pencarian secara konvensional.
2. Aplikasi ini dapat menampilkan skor berdasarkan relevansi setiap dokumen yang dibandingkan dari perbandingan proposal skripsi terhadap skripsi yang tersimpan dalam repository.
3. Aplikasi ini mengunakan query simple elasticsearch yang dipanggil dengan URL. Sehingga aplikasi ini terbatas saat melakukan pencarian dengan fulltext ..

REFERENSI

[1] A.A. Gede Yudhi Paramartha, G. K. (2016). IMPLEMENTASI WEB SERVICE PADA SISTEM PENGINDEKSAN DAN PENCARIAN DOKUMEN TUGAS AKHIR, SKRIPSI, DAN PRAKTIK KERJA LAPANGAN. *Jurnal Sains dan Teknologi, Vol. 5, No. 2*, 775-784.

[2] Akdoğan, H. (2015). *Elasticsearch Indexing.* Birmingham: Packt Publishing Ltd.

[3] Alambiyah, W. (2015, Maret). *Wahidin Alambiyah*. Dipetik Mei 17, 2019, dari Pengertian Web Service dan Web Server: //wahidin-alambiyah-19.blogspot.com/2015/03/pengertian-web-service-dan-web-server.html

[4] Allamaraju, S. (2010). RESTful Web Services Cookbook. Sebastopol: O'Reilly Media, Inc.

[5] Ardian Prima Atmajaa, S. V. (2018). Pemanfaatan Elasticsearch untuk Temu Kembali Informasi Tugas Akhir. *Jurnal Nasional Teknologi dan Sistem Informasi, VOL. 04 NO. 03* , 160-167.

[6] Clinton Gormley, Z. T. (2015). *Elasticsearch The Definitive Guide.* Sebastosol: O'Reilly Media, Inc.

[7] Gupta, Y. (2015). *Kibana Essentials.* Brimingham: Packt Publishing Ltd.

[8] Marrs, T. (2017). *JSON at Work.* Sebastopol: O'Reilly Media, Inc.

[9] Melita, R., Amrizal, V., Suseno, H. B., & Dirjam, T. (2018). Penerapan Metode Term Frequency Inverse Document Frequency (Tf-Idf) Dan Cosine Similarity Pada Sistem Temu Kembali Informasi Untuk MengetahuiSyarah Hadits Berbasis Web (Studi Kasus: Syarah Umdatil Ahkam). *Jurnal Teknik Informatika, Vol 11 No. 2*.

[10] Michael Mccandless, E. H. (2010). *Lucene in Action.* Stamford,: Manning Publications Co.

[11] Rafał Kuć, M. R. (2014). *Elasticsearch Server.* Birmingham: Packt Publishing.

[12] Rohman, A. (2017, Februari 9). *Documentation & Testing API dengan Postman part 1*. Dipetik Juni 16, 2019, dari Medium.com: https://medium.com/skyshidigital/documentation-testing-api-dengan-postman-part-1-5d33e430dca7 Ardani Rohman

[13] Utomo, D. B. (2018, September 19). Similarity. Dipetik Juni 2019, 11 , dari Prezi: https://prezi.com/q6swh_cn9xc8/similarity/

[14] W. Bruce Croft, D. M. (2015). Search Engi nes Information Retrieval in Practice. Pearson Education, Inc.

[15] Wijaya, T. (2009, 09). Model-Driven Development. Dipetik Juni 11, 2019, dari Information System Lecture Notes: http://trisnadi169.blogspot.com/2009/09/model-driven-development.html

[16] Bloom ,SA 1981. Indeks kesamaan dalam studi komunitas: potensi perangkap. Ekologi Laut-seri kemajuan 5: 125-128