

**IMPLEMENTASI ELASTICSEARCH UNTUK
PENGELOLAAN REPOSITORY KARYA ILMIAH
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG**

SKRIPSI

Karya Tulis sebagai syarat untuk memperoleh
Gelar Sarjana Komputer dari Fakultas Teknologi Informasi
Universitas Bale Bandung

Disusun oleh:

JERI RISMAWAN
NIM. C1A150003



PROGRAM STRATA 1
PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG
BANDUNG

2019

LEMBAR PERSETUJUAN PEMBIMBING

**IMPLEMENTASI ELASTICSEARCH UNTUK
PENGELOLAAN REPOSITORY KARYA ILMIAH
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG**

Disusun oleh:

JERI RISMAWAN
NIM. C1A150003

Telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar
SARJANA KOMPUTER

Pada

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG**

Baleendah, Agustus 2019

Disetujui oleh:

Pembimbing 1

Pembimbing 2

Yudi Herdiana, S.T.,M.T
NIK. 04104808008

Nurul Imamah, S.T.,M.T
NIK. 04104808121

LEMBAR PERSETUJUAN PENGUJI

**IMPLEMENTASI ELASTICSEARCH UNTUK
PENGELOLAAN REPOSITORY KARYA ILMIAH
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG**

Disusun oleh:

JERI RISMAWAN
NIM. C1A150003

Telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar
SARJANA KOMPUTER

Pa da

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG**

Baleendah, Agustus 2019

Disetujui oleh:

Penguji 1

Penguji 2

Denny Rusdianto, S.T.,M.KOM
NIK. 04104808121

Yaya Suharya, S.Kom.,M.T
NIK. 01043170007

LEMBAR PENGESAHAN PROGRAM STUDI

**IMPLEMENTASI ELASTICSEARCH UNTUK
PENGELOLAAN REPOSITORY KARYA ILMIAH
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG**

Disusun oleh:

JERI RISMAWAN
NIM. C1A150003

Skripsi ini telah diterima dan disetujui untuk memenuhi persyaratan mencapai
gelar

SARJANA KOMPUTER

Pada

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG**

Baleendah, Agustus 2019

Mengetahui,
Dekan,

Mengesahkan,
Ketua Program Studi

Yudi Herdiana, S.T.,M.T
NIK. 04104808008

Yaya Suharya, S.Kom.,M.T
NIK. 01043170007

LEMBAR PERNYATAAN KEASLIAN

Saya yang bertanda tangan dibawah ini:

Nama : JERI RISMAWAN

NIM : C1A150003

Judul Skripsi : **IMPLEMENTASI ELASTICSEARCH UNTUK PENGELOLAAN REPOSITORY KARYA ILMIAH FAKULTAS TEKNOLOGI INFORMASI UNIVERSITAS BALE BANDUNG**

Menyatakan dengan sebenarnya bahwa penulisan skripsi ini berdasarkan hasil penelitian, pemikiran dan pemaparan asli dari saya sendiri, baik untuk naskah laporan maupun kegiatan *programming* yang tercantum sebagai bagian dari skripsi ini. Jika terdapat karya orang lain, saya mencantumkan sumber yang jelas.

Pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan ketidak benaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini dan sanksi lain sesuai dengan peraturan yang berlaku di FAKULTAS TEKNOLOGI INFORMASI UNIVERSITAS BALE BANDUNG.

Demikian surat pernyataan ini saya buat dalam keadaan sadar tanpa paksaan dari pihak manapun.

Baleendah, Agustus 2019
Yang membuat pernyataan,

JERI RISMAWAN
NIM. C1A150003

ABSTRACT

Repository of scientific work in universities is a valuable asset that must be preserved. When talking about repositories they cannot be separated from digital repositories. The digitization process is the process of transferring collection media from printed formats to digital formats. The digital collection is then managed and collected in a container (data center server) so that it can be used or accessed by users quickly and accurately. Case study taken in this final project is to build ELASTICSEARCH IMPLEMENTATION IN SCIENTIFIC WORKS REPOSITORY OF INFORMATION TECHNOLOGY FACULTY OF BALE BANDUNG UNIVERSITY.

In line with technological developments, the compiler found an interesting case study to be studied at the Information Technology Faculty, namely, the Thesis data processing process at the Information Technology Faculty is not yet available. digital files, because of storage that has not been indexed and the unavailability of cloud storage that functions as a digital file storage medium (Repository). For this reason, the writer is interested in developing a management information system, especially in building the Scientific Work Repository at the Information Technology Faculty of Bale University, Bandung.

The method used in this writing in the form of observation, interviews with writing sources, and literature study. In this paper the authors design a system using Elasticsearch as a database engine which is a database that has the concept of NoSQL with the hope that the above problems can be reduced, besides that it can improve management activities and increase work motivation, especially in the administration, so that the required data related to thesis data can be managed quickly.

Keywords: *Repository, Scientific work, Elasticsearch, NoSQL.*

ABSTRAK

Repository karya ilmiah di perguruan tinggi merupakan aset berharga yang harus dilestarikan. Apabila berbicara tentang repository maka tidak akan bisa dipisahkan dengan digital repository. Proses digitalisasi merupakan proses alih media koleksi dari format tercetak menjadi format digital. Koleksi digital tersebut kemudian dikelola dan dikumpulkan dalam suatu wadah (pusat data server) agar dapat dimanfaatkan atau diakses oleh pemustaka secara cepat dan tepat. Studi kasus yang diambil dalam proyek akhir ini yaitu, membangun IMPLEMENTASI ELASTICSEARCH PADA REPOSITORY KARYA ILMIAH FAKULTAS TEKNOLOGI INFORMASI UNIVERSITAS BALE BANDUNG.

Sejalan dengan perkembangan teknologi, penyusun mendapati satu studi kasus yang menarik untuk dikaji di Fakultas Teknologi Informasi yaitu, Proses pengolahan data Skripsi di Fakultas Teknologi informasi belum tersedianya User Interface sebagai media komunikasi berkas digital Sehingga timbul permasalahan seperti Sulitnya temu kenali dokumen baik dokumen berkas fisik maupun berkas digital, karena penyimpanan yang belum terindeks dan tidak tersedianya cloud storage yang berfungsi sebagai media penyimpanan berkas digital (Repository). Untuk itu penulis tertarik untuk mengembangkan sistem informasi manajemen khususnya dalam membangun Repository Karya Ilmiah Di Fakultas Teknologi Informasi Universitas Bale Bandung.

Metode yang digunakan dalam penulisan ini berupa observasi, wawancara dengan sumber penulisan, dan studi pustaka. Dalam penulisan ini penulis merancang suatu sistem dengan menggunakan *Elasticsearch* sebagai database engine yang merupakan sebuah database yang memiliki konsep NoSQL dengan harapan permasalahan di atas dapat dikurangi, selain itu dapat meningkatkan aktifitas manajemen serta menambah motivasi kerja khususnya di bagian tata usaha, sehingga data yang diperlukan yang berhubungan dengan data skripsi dapat dikelola dengan cepat.

Kata kunci: Repository, Karya ilmiah, Elasticsearch, NoSQL.

KATA PENGANTAR

Puji dan syukur hanyalah milik Allah Subhanahuwata'ala yang mana berkat limpahan Rahmat dan Karunia-nya penulis dapat menyelesaikan Skripsi yang berjudul “IMPLEMENTASI ELASTICSEARCH UNTUK PENGELOLAAN REPOSITORY KARYA ILMIAH FAKULTAS TEKNOLOGI INFORMASI UNIVERSITAS BALE BANDUNG” dengan baik dan tepat pada waktunya.

Tujuan penulisan ini disusun guna untuk melengkapi salah satu syarat dalam menyelesaikan jenjang Strata satu jurusan Teknik Informatika. Dalam kesempatan ini penulis juga ingin menyampaikan ucapan terima kasih kepada :

1. Kedua Orang Tua penulis yaitu Bapak Parman dan Ibu Yayah Rokayah yang sangat penulis cintai.
2. Bapak Yudi Herdiana, S.T.,M.T. Selaku Dekan di Fakultas Teknologi Informasi sekaligus Dosen Pembimbing satu.
3. Bapak Yaya Suharya, S.Kom.,M.T. selaku ketua Program Studi Teknik Informatika.
4. Ibu Nurul Imamah, S.T.,M.T. Selaku Dosen Pembimbing dua.
5. Rekan-rekan Mahasiswa Angkatan 2015 yang telah membantu dan memberikan dukungan kepada penulis.

Penulis menyadari sepenuhnya bahwa laporan ini masih jauh dari kesempurnaan, oleh karena itu kritik dan saran dari semua pihak yang bersifat membangun selalu penulis harapkan demi kesempurnaan laporan ini.

Baleendah, Agustus 2019

Penulis

DAFTAR ISI

<i>ABSTRACT</i>	vi
ABSTRAK	vii
KATA PENGANTAR.....	viii
DAFTAR ISI	ix
DAFTAR TABEL	xiii
DAFTAR GAMBAR	xiv
DAFTAR LAMPIRAN	xvi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	3
1.5 Metodologi	3
1.6 Sistematika	4
BAB II TINJAUAN PUSTAKA.....	6
2.1 Landasan Teori	6
2.2 Dasar Teori	10
2.2.1 Repository	10
2.2.2 Elasticsearch	12
2.2.3 Pengertian, Kelebihan Dan Kekurangan REST API.....	14
2.2.4 NoSQL (Not Only SQL).....	17
2.2.5 Search Engine.....	20
2.2.6 Basis Data	23
2.2.7 Database MySQL	26

2.2.8	JAVA	27
2.2.9	Java Script Object Notation (JSON)	30
2.2.10	Netbeans IDE	30
2.2.11	<i>POSTMAN</i>	31
2.2.12	KIBANA	31
2.2.13	Pengertian dan Fungsi XAMPP.	32
2.2.14	Model Driven Development (MDD).....	33
2.2.15	Clickchart Free Diagram And Flowchart Marker	34
2.2.16	Balsamic mhockap.	35
2.2.17	UML.....	36
2.2.18	Javafx Scene Builder.....	41
BAB III METODOLOGI PENELITIAN.....		42
3.1	Kerangka pikir	42
3.2	Deskripsi Kerangka Pikir	43
3.2.1	Investigasi Awal.....	43
3.2.2	Analisis Masalah	45
3.2.3	Analisis Kebutuhan	46
3.2.4	Desain.....	47
3.2.5	Konstruksi	48
3.2.6	Implementasi	49
3.2.7	Pembuatan Laporan.....	50
BAB IV ANALISIS DAN PERANCANGAN		52
4.1	Analisis	52
4.1.1	Instrumen Penelitian.....	52
4.1.2	Analisis Sistem	54
4.1.3	Analisis Kebutuhan.....	58

4.1.4 Hasil Analisis	59
4.2 Perancangan.....	60
4.2.1 <i>Use Case Diagram</i>	60
4.2.2 Class Diagram.....	61
4.2.3 Activity Diagram	62
4.2.4 Sequence Diagram	67
4.2.5 Perancangan User Interface	74
BAB V IMPLEMENTASI DAN PENGUJIAN	80
5.1 Implementasi	80
5.2 Pengujian	85
5.2.1 Pengujian <i>Server Elasticsearch</i>	85
5.2.2 Pengujian Login	86
5.2.3 Pengujian Tambah data baru	87
5.2.4 Pengujian Edit data dokumen.....	88
5.2.5 Pengujian Hapus data dokumen.	88
5.2.6 Pengujian Lengkapi indeks dokumen.	89
5.2.7 Pengujian Pencarian judul karya ilmiah berdasarkan Tahun Terbit	89
5.2.8 Pengujian Tambah akun baru.....	90
5.2.9 Pengujian Reset akun	90
BAB VI KESIMPULAN DAN SARAN	91
5.1 Kesimpulan.....	91
5.2 Saran	91
DAFTAR PUSTAKA	92
LAMPIRAN	94
Lampiran 1 Form Wawancara	94
Lampiran 2 Listing Program	96

Lampiran 3 Instalasi Elasticsearch 6.5.4	106
Lampiran 4 Kartu Bimbingan.....	109
Lampiran 5 RIWAYAT HIDUP PENULIS	110

DAFTAR TABEL

Tabel 2.1 Sejarah perkembangan SQL.....	24
Tabel 2.2 Kegunaan Dari Masing-masing Edisi	28
Tabel 4.1 Tabel Hasil Analisis	60
Tabel 5.1 Pengujian Aktif Elasticsearch di Browser.....	85
Tabel 5.2 Tampilan Aktif Elasticsearch di Postman	86
Tabel 5.3 menambahkan indeks kepada serfer elasticsearch	86
Tabel 5.4 Pengujian login staff perpustakaan	86
Tabel 5.5 Pengujian Login mahasiswa.....	87
Tabel 5.6 Ujicoba Tambah data baru	87
Tabel 5.7 Uji coba Edit data dokumen.	88
Tabel 5.8 Uji coba Hapus data dokumen	88
Tabel 5.9 Uji Coba lengkapi indeks dokumen	89
Tabel 5.10 Pengujian Pencarian judul karya ilmiah.....	89
Tabel 5.11 Uji coba tambah akun baru	90
Tabel 5.12 Pengujian Reset akun	90

DAFTAR GAMBAR

Gambar 2.1 Ilustrasi Basis Data	23
Gambar 2.2 Alur Hidup Basis Data	25
Gambar 2.3 Platformn Pada JENI	28
Gambar 3.1 Kerangka pikir	42
Gambar 3.2 Blok Pengujian	50
Gambar 4.1 Analisis Sistem Yang Sedang Berjalan	54
Gambar 4.2 Analisis Sistem yang akan dibangun	56
Gambar 4.3 Struktur Menu	60
Gambar 4.4 Use Case Diagram	60
Gambar 4.5 Class diagram	61
Gambar 4.6 Activity Diagram Tambah akun baru	62
Gambar 4.7 Activity Diagram Login	63
Gambar 4.8 Activity Diagram Reset Akun	63
Gambar 4.9 Activity Diagram tambah dokumen	64
Gambar 4.10 Activity Diagram Edit data dokumen	65
Gambar 4.11 Activity Diagram Hapus Dokumen	65
Gambar 4.12 Activity diagram cari judul karya ilmiah	66
Gambar 4.13 Activity diagram lengkapi indeks dokumen	67
Gambar 4.14 Sequence Diagram Login	68
Gambar 4.15 Sequence Diagram Tambah akun baru	68
Gambar 4.16 Sequence Diagram Reset akun	69
Gambar 4.17 Sequence Diagram Tambah Dokumen	70
Gambar 4.18 Sequence Diagram Hapus Dokumen	71
Gambar 4.19 Sequence Diagram edit data dokumen	72
Gambar 4.20 Sequence Pencarian judul skripsi	73
Gambar 4.21 Sequence Diagram Lengkapi Indeks Dokumen	73
Gambar 4.22 Mockup Login	74
Gambar 4.23 Mockup Login Mahasiswa	75
Gambar 4.24 Mockup Tambah akun baru	75
Gambar 4.25 Mockup Menu utama admin	76

Gambar 4.26 Mockup Menu utama mahasiswa.	76
Gambar 4.27 Mockup Hapus dokumen.....	77
Gambar 4.28 Mockup Edit data dokumen	77
Gambar 4.29 Mockup Reset akun	78
Gambar 4.30 Mockup Lengkapi indeks dokumen	78
Gambar 5.1 Strutur Menu Form Login	80
Gambar 5.2 Struktur menu Form Login Mahasiswa.....	81
Gambar 5.3 Struktur Menu Tambah Akun baru	81
Gambar 5.4 Struktur Menu Form menu utama admin	82
Gambar 5.5 Struktur Menu Form Menu Utama Mahasiswa.....	82
Gambar 5.6 Struktur Menu Form Hapus Dokumen	83
Gambar 5.7 Struktur Menu Form Edit data dokumen.....	84
Gambar 5.8 Struktur Menu Form Reset akun	84
Gambar 5.9 Struktur Menu Form Lengkapi Indeks Dokumen	85

DAFTAR LAMPIRAN

Lampiran 1 Form Wawancara.....	94
Lampiran 2 Listing Program	96
Lampiran 3 Instalasi Elasticsearch 6.5.4.....	106
Lampiran 4 Kartu Bimbingan	109
Lampiran 5 RIWAYAT HIDUP PENULIS.....	110

BAB I

PENDAHULUAN

1.1 Latar Belakang

Repository karya ilmiah di perguruan tinggi merupakan aset berharga yang harus dilestarikan. Apabila berbicara tentang repository maka tidak akan bisa dipisahkan dengan digital repository. Proses digitalisasi merupakan proses alih media koleksi dari format tercetak menjadi format digital. Koleksi digital tersebut kemudian dikelola dan dikumpulkan dalam suatu wadah (pusat data server) agar dapat dimanfaatkan atau diakses oleh pemustaka secara cepat dan tepat. Secara sederhana arti dari repository adalah tempat penyimpanan. Dalam konteks kepastakawanan, repository adalah suatu tempat dimana dokumen, informasi atau data disimpan, dipelihara dan digunakan.

Sering kita jumpai koleksi repository karya ilmiah di perguruan tinggi hanya disimpan di rak buku tanpa ada perawatan sehingga rusak tanpa ada manfaatnya bagi orang lain ataupun masyarakat luas. Padahal untuk menyusun repository tersebut dibutuhkan tenaga, waktu, dan biaya yang cukup banyak. Kondisi yang umum saat ini banyak repository institusi yang dihasilkan masih berupa bentuk cetak (hard copy/paper based). Bentuk ini sangat rawan mengalami kerusakan jika tidak disediakan tempat dan perawatan khusus.

Sejalan dengan perkembangan teknologi, penyusun mendapati satu studi kasus yang menarik untuk dikaji. FTI adalah salah Satu Fakultas yang terdapat di UNIBBA, Sekretariat Fakultas Teknologi Informasi yang berlokasi di Jl.R.A.A Wiranatakusuma No. 7 Baleendah Pada bagian Tata Usaha ada beberapa pokok permasalahan yang ada pada sistem pengolahan data yang ada sekarang, yang diantaranya adalah:

1. Sulitnya temu kenali dokumen karya ilmiah baik dokumen berkas fisik maupun berkas digital, karena penyimpanan yang belum terindeks.
2. Tidak tersedianya cloud storage yang berfungsi sebagai media penyimpanan berkas digital (Repository).

3. Didalam menghasilkan seluruh laporan yang akurat dan tepat relatif lama serta kurang lengkapnya laporan yang dihasilkan karena, proses pengelolaan dokumen skripsi Hanya menggunakan aplikasi Ms-Excel.

Dalam pengelolaan dokumen elektronik memerlukan teknik khusus yang memiliki perbedaan dengan pengelolaan dokumen tercetak. Proses penngelolaan dokumen elektronik melewati beberapa tahapan yang dapat kita rangkumkan dalam proses digitalisasi, penyimpanan dan pengaksesan/temu kenali dokumen. Pengelolaan dokumen elektronik yang baik dan terstruktur adalah bekal penting dalam pembangunan sistem perpustakaan digital (*Digital library*).

Berdasarkan uraian di atas, penulis mengambil kesimpulan bahwa diperlukan adanya penyimpanan berkas digital (Repository) yang di simpan dalam format elsticsearch sehingga memiliki performa pengaksesan data lebih cepat dan dapat melakukan pemindaian seluruh data di dokumen sehingga dapat dengan mudah dalam proses temu kenali dokumen dan mengakomodasi permasalahan diatas.

1.2 Rumusan Masalah

1. Bagaimana membuat aplikasi yang dapat melakukan pengarsipan dokumen yang lebih cepat, tepat, dan akurat?
2. Bagaimana mempermudah pihak tata usaha, dan dosen dalam temu kenali dokumen baik dokumen berkas fisik maupun berkas digital?
3. Bagaimana membuat aplikasi yang dapat mempermudah dalam proses penyimpanan arsip?
4. Bagimana cara melakukan sebuah pengindeksan dokumen secara lebih cepat dan efisien?

1.3 Batasan Masalah

Batasan-batasan dari aplikasi ini adalah :

1. Aplikasi ini menggunakan *Elasticsearch* 6.5.4
2. Aplikasi ini menggunakan Java sebagai bahasa pemrograman.

3. Aplikasi ini hanya sebatas penginputan data , edit data, penghapusan data, serta pencarian judul skripsi berdasarkan tahun terbit.
4. Aplikasi ini di lengkapi MySQL Sebagai database.
5. Aplikais ini hanya bisa menjalankan Client dalam satu waktu.
6. Aplikasi ini hanya sebatas pengarsipan dokumen skripsi.
7. Arsip yang di inputkan berupa file PDF dengan ukuran maksimum file 1 MB.

1.4 Tujuan

Adapun tujuan dari penelitian ini adalah untuk :

1. Membangun Repository di Fakultas Teknologi Informasi yang memiliki performa pengaksesan data lebih cepat.
2. mempermudah pihak tata usaha, dan dosen dalam temu kenali dokumen baik dokumen berkas fisik maupun berkas digital
3. Mengefisienkan proses Pengarsipan dokumen di Fakultas Teknologi Informasi Universitas Bale Bandung.
4. Mempermudah proses pengindeksan dokumen.

1.5 Metodologi

Metode pengumpulan data pada metodologi penelitian yaitu meliputi :

- 1 Observasi.

Observasi adalah suatu cara pengumpulan data dengan pengamatan langsung dan pencatatan secara sistematis terhadap objek yang akan di teliti. Observasi dilakukan oleh penulis dengan cara pengamatan dan pencatatan mengenai sistem pengarsipan dokumen Skripsi di bagian Tata Usaha Fakultas Teknologi Informasi UNIBBA.

- 2 Wawancara.

Penelitian di lakukan dengan proses tanya jawab kepada pihak tata usaha yang mengelola Perpustakaan dan dosen di Fakultas Teknologi Informasi yang mengetahui permasalahan yang akan di bahas.

- 3 Studi pustaka.
 - a. Mencari laporan hasil-hasil penelitian baru atau merupakan kelanjutan penelitian selanjutnya di perpustakaan fakultas teknologi informasi.
 - b. Mengumpulkan jurnal yang memuat artikel-artikel yang relevan dengan masalah.

1.6 Sistematika

Sistematika penulisan yang akan di bahas dalam penulisan laporan tugas akhir ini terdiri dari 6 (Enam) bab yaitu sebagai berikut :

BAB I PENDAHULUAN

Pada bab ini dijelaskan beberapa hal yang mendasari atau yang melatarbelakangi penulisan tugas akhir ini. Di dalam bab I ini dideskripsikan tentang latar belakang, Rumusan Masalah, Batasan Masalah, Tujuan, Metodologi, dan sistematika penulisan yang digunakan.

BAB II LANDASAN TEORI

Uraian dan penjelasan teori yang berhubungan dengan judul yaitu tentang Aplikasi Implementasi Elasticsearch Untuk Pengelolaan Repository Karya Ilmiah fakultas teknologi informasi UNIBBA dan beberapa hal mengenai permasalahan konsep dasar yang dijadikan sebagai acuan di dalam penyusunan tugas akhir ini. Lengkapnya dalam bab ini dijelaskan beberapa teori pendukung yang diperlukan dalam penyusunan tugas akhir.

BAB III METODOLOGI

Bab ini membahas tentang waktu dan tempat penelitian, alat dan bahan, metode pengumpulan data dan metode perancangan.

BAB IV ANALISA DAN PERANCANGAN SISTEM

Pada bab ini berisi penjelasan mengenai analisis yang di gunakan dan juga pemodelan sistem yang di gunakan.

BAB V IMPLEMENTASI DAN PENGUJIAN

Bab ini berisi pengujian serta hasil pengamatan secara keseluruhan pada aplikasi Implementasi Elasticsearch Untuk Pengelolaan Repository Karya Ilmiah fakultas teknologi informasi UNIBBA.

BAB VI KESIMPULAN DAN SARAN

Bab ini akan diuraikan kesimpulan dan hasil analisis yang dapat memberikan saran-saran kepada Fakultas Teknologi Informasi Universitas Bale Bandung.

BAB II

TINJAUAN PUSTAKA

2.1 Landasan Teori

Landasan Teori berisi tentang referensi dari jurnal atau penelitian terlebih dahulu sehingga penulis dapat memperkaya teori yang digunakan dalam mengkaji penelitian yang dilakukan. Berikut penelitian terdahulu berupa beberapa jurnal terkait daengan penelitian yang dilakukan penulis.

1. Pengelolaan Institutional Repository Di Perpustakaan Utsman Bin Affan Universitas Muslim Indonesial (NURHASANAH,2017, jurnal ilmu perpustakaan)

Perpustakaan Perguruan Tinggi merupakan institusi pengelola koleksi perpustakaan yang dilakukan secara profesional dengan menerapkan sistem yang telah ditetapkan untuk memenuhi kebutuhan pemustaka. Perpustakaan sering dikatakan sebagai “jantung” pada setiap institusi, oleh karenanya keberadaan perpustakaan di Perguruan Tinggi menduduki posisi yang sangat strategis dalam pengembangan ilmu pengetahuan dan teknologi.

Perpustakaan Perguruan Tinggi memiliki tujuan dan fungsi sebagai institusi yang memilih, menghimpun, mengolah, merawat serta menyebarkan informasi kepada *acivitas* akademika untuk menunjang terlaksananya Tri dharma Perguruan.

Penelitian ini membahas tentang pengelolaan Institutional Repository di perpustakaan Utsman Bin Affan UMI Makassar. Usaha menghimpun, mengelola, melestarikan dan menyebarluaskan karya-karya intelektual sebuah Perguruan Tinggi dikenal dengan istilah Institutional Repository (Simpanan Kelembagaan). Dengan pokok permasalahan: “Bagaimana proses pengelolaan institutional repository dan bagaimana kendala yang dihadapi dalam pengelolaan institutional repository di perpustakaan Utsman Bin Affan”.

Tujuan penelitian ini adalah untuk mengetahui bagaimana proses pengelolaan koleksi repository serta kendala yang dihadapi dalam pengelolaan repository di perpustakaan Utsman Bin Affan UMI Makassar.

Jenis penelitian ini adalah penelitian lapangan (field research) dengan menggunakan pendekatan deskriptif yaitu penelitian yang mengumpulkan informasi mengenai status suatu gejala sesuai dengan apa adanya ketika penelitian ini dilakukan. Sedangkan teknik analisa data yang digunakan adalah kualitatif yang bersifat induktif, yakni suatu analisis berdasarkan data yang diperoleh, selanjutnya dikembangkan pola hubungan tertentu menjadi hipotesis, kemudian memilih mana yang penting dan menyusun data yang relevan agar menjadi informasi yang dapat disimpulkan dan memiliki makna.

Hasil penelitian menunjukkan bahwa hal yang perlu dipersiapkan dalam pengelolaan repository ada 3, pertama Infrastruktur yang terdiri dari persiapan hardware dan software, kedua persiapan sumber daya manusia yang akan menjalankan program/sistem pengelolaan repository, ketiga adalah prosedur penerimaan koleksi yang meliputi syarat pengumpulan karya ilmiah sebelum memasuki tahap pengolahan. Proses pengelolaan koleksi repository di perpustakaan Utsman Bin Affan meliputi 2 proses: pertama, proses awal digitalisasi yaitu dari tahapan pembongkaran, scanning, dan editing. Kedua, proses pengolahan koleksi dari tahap editing ke proses uploading atau proses penginputan ke dalam repository hingga akhirnya karya tersebut dapat di akses pada digilib UMI.

2. Sistem Informasi Pengarsipan Dokumen Berbasis Web (Agustina Simangunsong, 2018, jurnal Mantik Penusa, Volume 2 No. 1)

Arsip merupakan salah satu sumber informasi yang memiliki fungsi penting untuk menunjang proses kegiatan administrasi dan manajemen sebuah instansi. Semua kegiatan yang dilakukan oleh instansi tersebut, baik itu berupa proposal, surat-menyurat maupun

dokumen-dokumen lain akan menjadi arsip. Informasi yang terekam tersebut merupakan bukti dan dokumentasi atau memori bagi instansi yang bersangkutan.

Dokumen arsip akan terus bertambah seiring berjalannya waktu serta semakin kompleksnya kegiatan dan fungsi instansi. Apabila arsip yang dimiliki kurang baik pengelolaannya dapat mengakibatkan sulitnya menemukan informasi yang telah disimpan dan akhirnya dapat menghambat tahapan proses pekerjaan selanjutnya. Oleh karena itu sebaiknya arsip dikelola menggunakan sistem pengelolaan arsip yang baik dan benar.

Pada penelitian ini dibahas bagaimana merancang, menerapkan sistem serta pemeliharaan dan keamanan dokumen sehingga terhindar dari kerusakan. Sistem dibuat dengan menggunakan aplikasi php mysql dengan tujuan pengarsipan dokumen yang berbasis web ini dilakukan untuk membantu dan mencegah proses yang sangat lama dalam penyimpanan maupun pencarian suatu dokumen yang dibutuhkan di waktu yang cepat, tepat dan terperinci. Oleh karena itu sistem ini sangat membantu pihak SDM di Perumnas Regional – I Medan dalam melakukan pekerjaan dengan efektif dan efisien.

Metode pengumpulan data dalam penelitian ini yaitu studi kepustakaan, wawancara, observasi, dan dokumen. Berdasarkan pembahasan pengarsipan dokumen berbasis web pada Perumnas Regional-I Medan peneliti berharap sistem yang dibuat membantu dan memudahkan bagian SDM dalam proses penyimpanan, pencarian dan dokumen yang tak mudah hilang atau tercecer.

Berdasarkan pembahasan tentang sistem informasi pengarsipan dokumen berbasis web pada perumnas- regional I Medan, maka diambil kesimpulan sebagai berikut :

- a. Sistem Pengarsipan yang dibangun di sub bagian SDM telah berbasis web.

- b. Penataan dokumen tersusun rapi dan menggunakan waktu yang lebih efektif.
 - c. Dengan adanya sistem ini, maka dapat memudahkan proses pekerjaan di dalam mencari dokumen yang dibutuhkan suatu waktu dengan cepat dan terperinci.
 - d. Dokumen terpelihara dan aman.
3. Perbandingan Kemampuan Database NoSQL dan SQL dalam Kasus ERP Retail (Faizal Anugrah Bhaswara, 2017, Jurnal Teknik ITS Vol. 6, No. 2).

Pada setiap aplikasi yang dibangun oleh developer tidak akan lepas dari peran database. Database atau basis data sendiri adalah kumpulan data yang disimpan secara sistematis di dalam komputer yang dapat diolah maupun direkayasa menggunakan perangkat lunak untuk menghasilkan informasi.

Isi dari makalah ini adalah membandingkan antara database relational (SQL) dengan database non-relational (NoSQL) dalam hal performance, flexibility, dan scalability. Setelah terbukti database mana yang lebih unggul dari keduanya maka akan diterapkan untuk aplikasi ERP Retail yang berorientasikan multitenancy. Dengan begitu diharapkan ERP Retail yang dibangun akan memiliki performa bagus, fleksibel dalam penyimpanan data, juga mendukung penyimpanan data yang terus berkembang seiring berjalannya waktu.

Pengujian dibagi menjadi dua: uji performa dan uji fleksibilitas. Pengujian performa dilakukan dengan memasukkan dataset untuk pengujian uji coba insert dan pengelolaan data dalam database dengan query join/aggregation, select, update, dan delete. Untuk pengujian fleksibilitas dilakukan dengan memasukkan beberapa data berbeda yang membentuk skema ke dalam database

Dalam uji coba yang telah dilakukan, database NoSQL terbukti memiliki kecepatan penyimpanan data yang lebih unggul dalam hal CRUD daripada SQL. Juga memiliki struktur penyimpanan data yang

fleksibel karena model data berupa BSON (Binary JSON). Dan memiliki kemampuan untuk menjadi scalable dengan metode sharding. Jadi dalam hal ini database NoSQL akan lebih baik untuk diterapkan pada ERP.

2.2 Dasar Teori

2.2.1 Repository

1. Pengertian Repository

Apabila berbicara tentang repository maka tidak akan bisa dipisahkan dengan digital repository. Proses digitalisasi merupakan proses alih media koleksi dari format tercetak menjadi format digital. Koleksi digital tersebut kemudian dikelola dan dikumpulkan dalam suatu sistem pengelolaan agar dapat dimanfaatkan atau diakses oleh pemustaka secara cepat dan tepat.

Di lingkungan perguruan tinggi digital repository ini dapat digunakan untuk mengumpulkan, melestarikan, dan menyebarluaskan hasil penelitian dan karya intelektual institusi dalam 1 format digital secara online ataupun offline. Konten repository institusi yang terkumpul ini juga dapat dimanfaatkan kembali untuk menunjang kegiatan akademik dan penelitian. Ada beberapa tujuan mengapa mesti ada IR, sebagaimana dicatat Wikipedia (2012) yaitu:

- a. Menyediakan akses terbuka kepada hasil penelitian kelembagaan
- b. Menciptakan visibilitas global untuk penelitian ilmiah suatu lembaga.
- c. Untuk mengumpulkan content dalam satu lokasi.
- d. Untuk menyimpan dan melestarikan aset kelembagaan digital lainnya termasuk di dalamnya grey literature, tesis, maupun laporan teknis.

2. Tahapan Awal Persiapan Pengelolaan Repository Institusi

Untuk mengelola repository institusi maka perlu dipersiapkan beberapa hal agar dalam tahap awal sampai tahap akhir pengelolaan repository dapat berjalan lancar dan untuk kedepan dapat dilakukan pengembangan-pengembangan sesuai dengan situasi dan kondisi perkembangan teknologi informasi yang berkembang dengan cepat. Ada beberapa hal yang perlu dipersiapkan dalam pengelolaan repository institusi diantaranya:

- a. Prosedur operasional (SOP) yang jelas Dalam pengelolaan repository diperlukan peraturan standar dan kebijakan yang jelas dari pimpinan atau stakeholder institusi, misalnya peraturan serah simpan repository institusi.
- b. Sarana dan Prasarana Untuk dapat mengelola repository maka perlu didukung sarana dan prasarana yang memadai seperti komputer dan server dengan spesifikasi yang ideal dalam hal pengelolaan koleksi digital. Untuk media penyimpanan (*hardisk*) direkomendasikan menggunakan kapasitas yang besar sehingga dapat menampung banyak file digital dan dapat digunakan dalam kurun waktu jangka panjang serta disediakan hardisk backup. Server juga membutuhkan perangkat tambahan UPS (*uninterruptible power supply*) yang merupakan perangkat penyimpan daya sementara apabila terjadi listrik mati.
- c. Alat bantu alih media Dalam kegiatan proses alih media koleksi dari bentuk *hardcopy* ke dalam bentuk digital (*softcopy*) maka pustakawan membutuhkan perangkat alat bantu berupa hardware dan software. Perangkat minimal yang dibutuhkan adalah hardware yang dapat berupa scanner, digunakan sebagai alat pemindai berformat teks tercetak ke dalam format digital. Spesifikasi yang direkomendasikan adalah yang bertipe ADF (*Automatic Document Feeder*) sehingga lebih cepat dalam proses scanning. Selain hardware juga dibutuhkan software atau aplikasi

pengolah file PDF, foto, audio video atau sejenisnya sesuai dengan format koleksi.

- d. Koneksi jaringan internet Sebuah komputer server repository harus terhubung dengan koneksi internet 24 jam yang stabil agar mudah diakses oleh pihak lain. Koneksi harus dilengkapi dengan firewall untuk keamanan data agar tidak mudah diganggu atau dirusak dari pihak yang berniat jahat. Kebutuhan *bandwidth* koneksi harus disesuaikan dengan jumlah pengakses setiap harinya. Hal ini dikarenakan bentuk *file* digital rata-rata memiliki ukuran cukup besar sehingga dapat menghabiskan bandwidth jika jumlah pengunjungnya banyak.
- e. Perangkat lunak repository Perangkat lunak repository adalah aplikasi yang akan dijadikan sebagai sarana untuk mengautomatisasikan proses pengelolaan repository institusi. Pemilihan perangkat lunak yang tepat dapat mempermudah pustakawan untuk mengelola repository agar mudah disajikan dan ditelusur oleh pemustaka serta mempunyai tingkat keamanan yang baik dan dapat dimodifikasi/dikembangkan sendiri.

(Agustiawan, S.S., M.IP, 2016, Buku Pedoman Pengelolaan Repository Institusi Di Upt Perpustakaan Isi Yogyakarta, 1)

2.2.2 *Elasticsearch*

Elasticsearch merupakan sebuah tool yang digunakan untuk query kata-kata. *Elasticsearch* dapat melakukan beberapa tugas yang lainnya, tetapi itu digunakan untuk mencari teks, mengembalikan hasil teks yang dicari menjadi query atau analisis statistik dari kata-kata. *Elasticsearch* merupakan standalone database server, dituliskan dengan bahasa Java, yang mengambil dan menyimpan data dalam format canggih yang dioptimalkan untuk pencarian berdasarkan teks. Dengan menggunakan *elasticsearch* bekerja menjadi lebih mudah diimplementasikan ke dalam HTTP/JSON. *Elasticsearch* mudah terukur, mendukung *clustering* dan menjadi pilihan yang out of the box.

ElasticSearch dapat digunakan baik sebagai mesin pencari dan sebagaimenyimpan data. Sebuah deskripsi singkat dari logika *ElasticSearch* membantu pengguna untuk meningkatkan kinerja dan kualitas, dan memutuskan kapan dan bagaimana berinvestasi di bidang infrastruktur untuk meningkatkan skalabilitas dan ketersediaan. beberapa rincian tentang ulangan data dan proses komunikasi dasar simpul juga dijelaskan. Berikut konsep dan fitur-fitur dasar tentang *ElasticSearch* :

1. *Index*

ElasticSearch menyimpan data ke dalam satu index atau lebih. Secara analogi dari dunia SQL, index merupakan sesuatu yang menyerupai dengan database.

2. *Document*

Yang dimaksud dengan dokumen yang ada pada *ElasticSearch* adalah yang terdiri dari beberapa *field* dan setiap field memiliki nama dan memiliki satu nilai atau lebih.

3. *Mapping*

Sebelum sebuah dokumen disimpan, dokumen terlebih dahulu dianalisa. Pengguna dapat mengkonfigurasi bagaimana teks input dibagi ke dalam token, dimana token tersebut harus disaring atau melalui proses tambahan terlebih dahulu, seperti menghapus tag HTML yang tidak diperlukan.

4. *Type*

Setiap dokumen memiliki tipe yang telah didefinisikan. Hal ini memungkinkan bahwa pengguna dapat menyimpan dokumen dengan berbagai macam tipe ke dalam satu index dan juga memiliki mapping yang berbeda-beda untuk setiap tipe dokumen yang berbeda pula. Tipe ini akan secara otomatis dikenali ketika proses indexing.

5. *Node*

Node merupakan *instance* dari *ElasticSearch*. Satu cluster mengandung beberapa node *ElasticSearch* yang sama-sama berbagi data dan beban kerja. Satu master node *ElasticSearch* diperlukan di dalam

cluster untuk manajemen node yang lain seperti penambahan dan penghapusan node. Perlu diingat bahwa komputer klien bisa mengakses node *ElasticSearch* manapun dalam cluster.

6. *Cluster*

Cluster merupakan sekumpulan *node* yang bekerja sama untuk mengendalikan data yang lebih besar daripada hanya dengan menggunakan satu *node*

7. *Shard*

ElasticSearch membagi data ke dalam beberapa index fisik yang ada. Index fisik itu disebut dengan *Shard*.

8. *Replica*

Replika membantu *node* untuk mengurangi beban data yang meningkat dan ketika node tunggal tidak mampu menanganinya.

Dari percobaan yang telah dilakukan, didapatkan rata-rata waktu *search request* yaitu 1.304 detik dan waktu *index request* yaitu 1.093 detik. Untuk rata-rata waktu pencarian kata pada dokumen sangat singkat. Dari percobaan yang dilakukan didapatkan rata-rata waktu pencarian dokumen yaitu 0.0485 detik. Dengan menggunakan metode *ElasticSearch* dapat digunakan untuk proses dijadikan salah satu solusi untuk menangani big data yang terdiri dari data tidak terstruktur yang besar yaitu dengan menambahkan klasterisasi dalam indexing informasi.

(Rista Novitasari, 2015, Temu Kembali Informasi Menggunakan Elasticsearch Pada Unstructured Datatext Multidimensi, 27).

2.2.3 Pengertian, Kelebihan Dan Kekurangan REST API.

REST (REpresentational State Transfer) yang merupakan standar arsitektur berbasis web yang menggunakan protokol HTTP untuk berkomunikasi data. REST diperkenalkan oleh Roy Fielding pada tahun 2000. Lalu API itu artinya Application Protocol Interface dimana adalah kumpulan dari subroutine definitions, protocol dan juga tools untuk

berkomunikasi data antar aplikasi software. Dengan kata lain REST adalah salah satu dari desain arsitektur di dalam API.

Cara kerja RESTful API adalah REST client akan mengakses data/resource ke REST server dimana masing-masing resource atau data tersebut dibedakan oleh sebuah global ID atau URIs (Universal Resource Identifiers). Jadi data yang diberikan oleh REST server itu bisa berupa format text, JSON atau XML. Yang paling populer dipakai saat ini adalah format JSON.

Adapun metode HTTP yang secara umum dipakai dalam REST api adalah:

- a. GET, biasanya digunakan untuk membaca resource dari REST server
- b. POST, biasanya digunakan untuk membuat resource baru di REST server
- c. PUT, biasanya digunakan untuk memperbaharui resource di REST server
- d. DELETE, biasanya digunakan untuk menghapus resource dari REST server
- e. OPTIONS, digunakan untuk mendapatkan operasi yang disupport pada *resource* dari REST server

REST adalah salah satu implementasi dari web service sebagai sebuah standar yang digunakan untuk pertukaran data antar aplikasi atau sistem. Biasanya aplikasi atau sistem ini menggunakan bahasa pemrograman yang berbeda sehingga untuk bisa berkomunikasi satu sama lainnya bisa menggunakan web service ini.

1. CARA KERJA REST API.

Pertama harus ada sebuah REST server yang akan menyediakan resource/data. Sebuah REST client akan membuat HTTP request ke server melalui sebuah global ID atau URIs dan server akan merespon dengan mengirimkan balik sebuah HTTP response sesuai yang diminta client.

2. komponen dari HTTP request:
 - a. HTTP method seperti GET, POST, PUT, DELETE dll sesuai dengan tugasnya masing-masing
 - b. URI untuk mengetahui lokasi data di server
 - c. HTTP Version, seperti HTTP v1.1
 - d. Request Header, berisi metadata seperti Authorization, tipe client dan lain
 - e. Request Body, data yang diberikan client ke server seperti URI params.
3. komponen dari HTTP response:
 - a. Response Code, status server terhadap request yang diminta seperti 200, 401, 404 dan lainnya.
 - b. HTTP Version
 - c. Response Header yang berisi meta data seperti content type, cache tag dan yang lainnya.
 - d. Response Body, data/resource yang diberikan oleh server baik itu berupa text, json ataupun xml.
4. keuntungan dan kelemahan REST.
 - a. Kelebihan REST :
 - 1) Bisa digunakan oleh banyak bahasa pemrograman dan banyak platform
 - 2) Lebih simple dibandingkan dengan SOAP
 - 3) Mudah dipelajari
 - 4) ini seperti web, menggunakan protokol HTTP, dan HTTP ada dimana-mana.
 - b. Kekurangan:
 - 1) Waktu akses yang biasanya lebih lama dibandingkan dengan native library
 - 2) Lebih rentan dengan serangan keamanan karena harus melewati protokol HTTP.

(Agus Yusida, January 15 2018, Pengertian, Kelebihan Dan Kekurangan Rest Api, <https://ngide.net/posts/apa-itu-rest-api>)

2.2.4 *NoSQL* (Not Only SQL)

RDBMS merupakan suatu sistem khusus yang mengatur organisasi, penyimpanan, akses, keamanan dan integritas data. RDBMS merupakan sistem pengelolaan basis data yang paling banyak digunakan saat ini, meskipun terdapat berbagai macam sistem yang lain. RDBMS menyimpan informasi dalam suatu kumpulan tabel. Suatu desain RDBMS dikatakan bagus jika terdapat normalisasi basis data didalamnya. Beberapa aplikasi seperti weblog dan perbankan menggunakan Structure Query Language (SQL) dan RDBMS menawarkan solusi yang baik dalam penggunaan basis data. RDBMS menerapkan Atomicity, Consistency, Isolation, and Durability (ACID) dalam penyimpanan dan pengelolaan data [1]. Atomicity mengharuskan suatu modifikasi basis data mengikuti aturan “all or nothing”.

Suatu transaksi dikatakan atomic indikasinya adalah jika satu transaksi gagal maka menyebabkan kegagalan keseluruhan transaksi. Consistency mengatur validitas data yang akan dituliskan dalam basis data. Jika suatu transaksi berhasil dijalankan maka akan dilakukan pengambilan basis data dari suatu kondisi yang konsisten dengan aturanaturannya menuju kondisi lain yang juga konsisten terhadap aturan-aturan yang berlaku. Isolation menyatakan bahwa transaksi yang berlipat ganda dalam satu waktu tidak berpengaruh terhadap pengeksekusian transaksi yang lain. Durability memastikan bahwa transaksi yang dilakukan tidak akan hilang dengan melakukan backup basis data dan log transaksi yang memberikan fasilitas restorasi dari transaksi yang dilakukan walaupun kemudian banyak terjadi kegagalan pada software dan hardwarenya.

Kombinasi penggunaan RDBMS, SQL, dan ACID memberikan infrastruktur pemrograman yang baik untuk aplikasi perbankan, toko online, dan aplikasi-aplikasi lain yang menitik beratkan pada manajemen transaksi. Tetapi tidak semua aplikasi memerlukan penggunaan ACID karena

konsekwensi yang harus ditanggung dalam kinerja aplikasi dan fitur-fiturnya.

Generasi terbaru dari perangkat lunak internet sangat tergantung pada penggunaan sistem basis data. Kenyataannya untuk meningkatkan kemampuan dan kecepatan operasi, pengembang dapat merelaksasikan beberapa aturan ketat yang ada pada RDBMS seperti consistency dan atomicity. Sebagai realisasinya jenis basis data baru telah diperkenalkan beberapa tahun lalu yang dikenal dengan NoSQL (singkatan dari Not-Only-SQL). Beberapa aplikasi berbasis web yang telah menerapkan penggunaan NoSQL diantaranya adalah Google dengan BigTable, Amazon dengan Dynamo, dan Facebook dengan Cassandra dan Hadoop.

Dengan mengesampingkan aspek ACID, NoSQL menerapkan konsep BASE (Basically Available, Soft state, dan Eventually consistent) untuk meningkatkan availability dan partitioning data. Basically Available memastikan bahwa sistem bekerja sepanjang waktu. Soft state merupakan suatu kondisi dimana sistem tidak harus konsisten setiap saat, dan Eventually konsisten yang menekankan bahwa sistem akan menjadi konsisten beberapa waktu kemudian. Makalah ini menjelaskan tentang permasalahan pada RDBMS yang melatar belakangi kemunculan NoSQL, konsep umum NoSQL, dan juga beberapa kritik terhadap konsep NoSQL. Karena NoSQL bukanlah Peluru Emas yang dapat menyelesaikan semua masalah basis data, makalah ini dapat memberikan gambaran kapan seseorang/instansi perlu menggunakan NoSQL, dan kapan harus menggunakan RDBMS.

NoSQL merupakan basis data non relasional dengan *schema-free* yang memunculkan pertanyaan bagaimana *NoSQL* dapat melakukan partisi untuk data yang berukuran besar, melakukan *query*, replikasi data, dan mendukung adanya konsistensi. Ada empat model data *NoSQL*, yaitu:

1. Column-oriented

Penerepan *column-oriented* terdapat dalam *Casssandra*. *Cassandra* menggunakan distribusi multidimensional *map indexed* dengan sebuah

key. Baris kunci yang sering digunakan dalam *Cassandra* adalah *string* dengan panjang 16-36 *bytes*. Setiap kolom digabungkan menjadi sebuah *coloumn families*. Suatu kolom unit *atomic* dari informasi didukung oleh *Cassandra* yang diekspresikan dengan nama *value*. *Super-column* merupakan gabungan dari kolom dengan nama yang umum dan digunakan untuk pemodelan tipe data yang kompleks. Baris secara unik mengidentifikasi data yang terdapat dalam *column* dan *super-column*. Dalam *Cassandra* baris dapat dikenali dengan sebuah kunci. *Column Family* merupakan bagian dari suatu unit abstraksi yang berisi baris kunci yang tergabung dalam *column* dan *super column* yang memiliki struktur data yang tinggi. *Keyspace* merupakan level tertinggi dari unit informasi yang terdapat dalam *Cassandra*. Kumpulan *column families* sebenarnya merupakan subordinat dari satu *keyspace*. Pada intinya model data *column-oriented* memungkinkan suatu aplikasi secara bebas untuk mengembangkan bagaimana informasi disusun berdasarkan suatu desain *schema*.

2. Document-oriented

Contoh penerapan model data *document-oriented* terdapat dalam *CouchDB* yang dibangun oleh *IBMTM*. Basis data yang memiliki model data *document-oriented* sangat bermanfaat untuk suatu domain yang bentuk masukannya dokumen yang tidak terstruktur seperti *web pages*, *wikis*, *discussion forums*, dan *blogs*. Data tersimpan dalam basis data *CouchDB* yang mencakup serangkaian dokumentasi yang berisi beberapa atribut dan nilai dengan masing-masing *id* yang unik dan *metadata*. *CouchDB* tidak pernah melakukan *overwrite document*, melainkan menambahkan dokumen baru ke basis data bila diperlukan seperti ketika terjadi proses *update*.

3. Object-oriented

Basis data berorientasi objek adalah model basis data dimana informasi direpresentasikan dalam bentuk objek yang digunakan dalam pemrograman berorientasi obyek. *OrientDB* adalah contoh dari basis data *object-oriented*. Dokumen *OrientDB* yang ada dalam suatu *cluster*,

dapat berupa fisik, logis atau *in-memory*, yang digunakan untuk menyimpan *link* ke dalam data. *Cluster* adalah cara yang sangat umum untuk mengelompokkan *record*, hal ini merupakan suatu konsep yang tidak ada dalam basis data relasional. Cara ini dapat mengelompokkan semua *record* pada jenis tertentu, atau dengan nilai-nilai tertentu. *OrientDB* menggunakan segmen data untuk menyimpan isi *record*. Segmen data mirip dengan *file physical cluster* yang menggunakan dua atau lebih *file*, yaitu satu atau beberapa *file* dengan ekstensi "*oda*" (*Orient Data*) dan hanya satu *file* dengan ekstensi "*odh*"

4. Graph-oriented

Basis data grafik (*GraphDB*) adalah basis data yang menggunakan struktur grafik yang berisi *node*, *edge*, dan properti untuk mewakili dan menyimpan informasi. *GraphDB* diperlukan untuk data grafik yang berskala besar, terutama yang dipergunakan oleh para peneliti biologi jaringan dan situs jaringan sosial, seperti *Facebook*, dan *Twitter*. *GraphDB* memetakan secara langsung objek ke aplikasi dan lebih intuitif untuk menggambarkan *data set* asosiatif. Beberapa keuntungan dari *GraphDB* adalah *Intuitive*, dimengerti oleh pikiran manusia, yaitu menggambarkan entitas dan hubungan sebagai grafik masalah umum yang akrab dengan manusia; *Elemental* untuk ilmu komputer, yaitu grafik, terutama grafik pohon (seperti *binary-tree*, *B+ tree*, *red-black tree*) berfungsi sebagai struktur data dasar dalam ilmu komputer dan berbagai masalah (*shortest path* dan *max-flow*) dapat diubah dan diselesaikan dengan algoritma grafik; *Ubiquitous*, yaitu pemodelan *ER* ke model jejaring sosial selalu dikelilingi oleh grafik baik di komputer ataupun dalam kenyataan.

(Fahri Firdausillah, 2002, NoSQL: Latar Belakang, Konsep dan Kritik, 4)

2.2.5 Search Engine

Mesin pencari atau 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 peladen dalam suatu jaringan. Search engine

merupakan perangkat pencari informasi dari dokumen-dokumen yang tersedia. Hasil pencarian umumnya ditampilkan dalam bentuk daftar yang seringkali diurutkan menurut tingkat akurasi ataupun rasio pengunjung atas suatu berkas yang disebut sebagai hits. Informasi yang menjadi target pencarian bisa terdapat dalam berbagai macam jenis berkas seperti halaman situs web, gambar, ataupun jenis-jenis berkas lainnya. Beberapa mesin pencari juga diketahui melakukan pengumpulan informasi atas data yang tersimpan dalam suatu basisdata ataupun direktori web.

fungsi search engine adalah untuk mencari alamat-alamat website yang berisi berbagai bentuk informasi seperti tulisan, gambar, video dengan mudah dan cepat.

kekurangan dalam menggunakan MSN :

1. Pencarian. Dibutuhkan waktu yang lama sampai hasil ditampilkan di halaman MSN.
2. Pengguna juga sulit membedakan antara iklan dan bukan iklan pada halaman pencariannya.
3. Penampilan iklan hanya dibedakan dari hasil asli melalui sebuah titik di depannya.

Bagian-bagian dari MSN adalah

1. Windows Live Hotmail

Windows Live Hotmail, atau lebih dikenal dengan Hotmail adalah layanan email gratis populer dari Microsoft, bagian dari layanan-layanan Windows Live. Versi terbaru ini diumumkan pada tanggal 1 November 2005 sebagai update dari layanan MSN Hotmail yang sudah ada. Setelah diluncurkannya versi beta, Windows Live Hotmail diluncurkan secara resmi pada tanggal 7 Mei 2007

2. Windows Messenger

Windows Live Messenger adalah program perangkat lunak pengirim pesan instan yang dikembangkan oleh MSN. Terdapat fasilitas untuk mencari orang yang ada di situs tersebut. Program yang dapat didownload

meliputi messenger, surel, penulis atau writer, galeri foto, pembuat film beta, keamanan keluarga dan kotak peralatan.

3. My MSN

Microsoft Internet Explorer 8. MSN di dalamnya juga terdapat beberapa site akses diantaranya MSN Image, MSN Videos, MSN Shopping, MSN News dan MSN Maps:

a) MSN Image

MSN Image adalah situs web yang digunakan untuk mencari gambar tertentu. Sama seperti Google image search, pengguna hanya mengetikkan kata kunci tersebut, lalu gambar tersebut ditentukan.

b) MSN Videos

MSN Videos adalah situs web yang digunakan untuk mencari video tertentu, sama seperti YouTube. Tetapi pertama pengguna harus menentukan negara dan bahasanya, seperti negara Indonesia dengan bahasa Inggris.

c) MSN Shopping

MSN Shopping adalah situs web portal berbelanja online yang dimiliki oleh MSN dan berkompetisi dengan Yahoo!. MSN Shopping juga terdapat fasilitas pencarian produk. MSN Shopping kini tersebar di seluruh dunia.

d) MSN News

MSN News adalah situs web yang digunakan untuk mencari berita tertentu.

e) MSN Maps

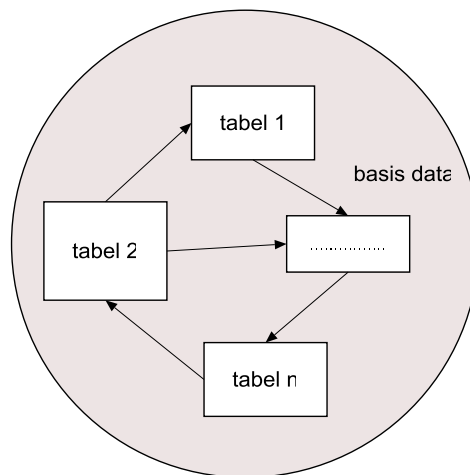
MSN Maps adalah situs web yang digunakan untuk pelayanan pemetaan seperti Google Maps.

(M. Ardi Yudha Prawir, 2017, Search Engine, <https://fryunfirst.blogspot.com/2013/05/search-engine.html>)

2.2.6 Basis Data

1. Pengertian basis data

Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah di olah atau informasi dan membuat informasi tersedia saat di butuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat.



Gambar 2.1 Ilustrasi Basis Data

Sistem informasi tidak dapat di pisahkan dengan kebutuhan akan basis data apapun bentuknya, entah berupa file teks ataupun *Database Management System (DBMS)*.

Kebutuhan basis data dalam sistem informasi meliputi:

- Memasukan, menyimpan dan mengambil data.
- Membuat laporan berdasarkan data yang telah di simpan.

Tujuan dari dibuatnya tabel-tabel disini adalah untuk menyimpan data ke dalam tabel-tabel agar mudah di akses. Oleh karena itu untuk merancang tabel-tabel yang akan dibuat maka dibutuhkan pola pikir penyimpanan data nantinya jika dalam bentuk baris-baris data (record) .

2. DBMS

DBMS (*Database Management System*) atau dalam bahasa indonesia sering disebut sebagai Sistem Manajemen Basis Data adalah

suatu sistem aplikasi yang di gunakan untuk menyimpan, mengelola, dan menampilkan data. Suatu aplikasi dikatakan DBMS jika memenuhi persyaratan minimal sebagai berikut:

- Menyediakan fasilitas untuk mengelola akses data.
- Mampu menangani integritas data.
- Mampu menangani akses data yang dilakukan
- Mampu menangani *backup data*.

Karena pentingnya data bagi suatu organisasi/perusahaan, maka sebagian perusahaan memanfaatkan DBMS Dalam mengelola data yang mereka miliki. Pengelolaan DBMS sendiri biasanya ditangani oleh tenaga ahli yang spesialis menangani DBMS yang disebut dengan DBA (*Database Administrator*).

Berikut ini adalah 4 macam DBMS versi komersial yang paling banyak digunakan didunia saat ini yaitu:

- Oracle
- Microsoft SQL Server
- IBM DB2
- Microsoft Access

Sedangkan DBMS versi open source yang cupup berkembang dan paling banyak digunakan saat ini adalah sebagai berikut:

- MySQL
- PostragreSQL
- Firebrid
- SQLite

3. SQL

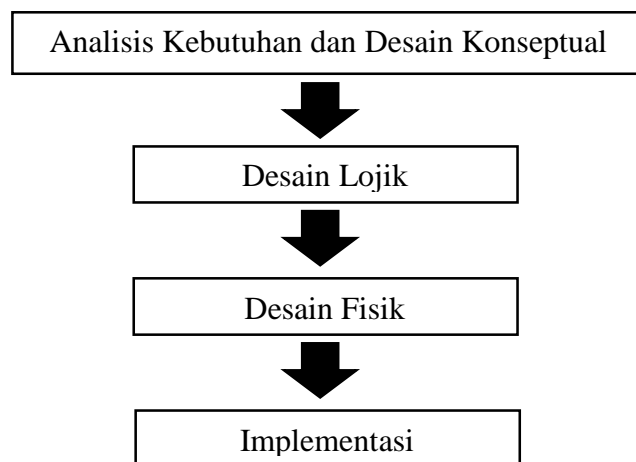
SQL (*Structured Query Language*) adalah bahasa yang digunakan untuk mengelola data pada RDBMS. SQL awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus. Berikut ini sejarah perkembangan SQL sampai saat ini:

Tabel 2.1 Sejarah perkembangan SQL

NO.	Tahun	Nama
1	1986	SQL-86
2	1989	SQL-89
3	1992	SQL-92
4	1999	SQL:1999
5	2003	SQL:2003
6	2006	SQL:2006
7	2008	SQL-2008
8	2011	SQL:2011

4. Alur Hidup Basis Data

Tidak hanya perangkat lunak yang memiliki alur hidup, dalam perancangan membuat basis data juga memiliki alur hidup atau *Database Live Cycle* (DBLC). Alur hidup basis data dapat dilihat pada gambar berikut:



Gambar 2.2 Alur Hidup Basis Data

Fase-fase DBLC antara lain:

a. Analisis kebutuhan

Hal-hal yang harus dilakukan pada tahap ini adalah:

- 1) Didefinisikan dengan mewawancarai produsen dan pemakai data, data apa sajakah yang butuh untuk disimpan dan terkait dengan aplikasi komputer yang akan dibangun.
- 2) Membuat kontrak spesifikasi data.

3) *Entity Relationship Diagram I(ERD)* sebagai bagian dari desain konseptual.

b. Desain logik basis data

Pada tahap ini harus dibuat rancangan logik basis data. Biasanya pada tahap ini di buat *Conceptual Data Model (CDM)*.

c. Desain fisik basis data

Pada tahap ini harus di buat rancangan fisik basis data biasanya pada tahap ini dibuat *Physical Data Model (PDM)*.

d. Implementasi

1) Membuat Queri SQL

2) Aplikasi ke DBMS atau *file*.

(Rosa A.S, M.Shalahuddin, 2015, Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek, 43).

2.2.7 Database MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: database management system) atau DBMS yang multialur, multipengguna, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual di bawah lisensi komersial untuk kasus-kasus di mana penggunaannya tidak cocok dengan penggunaan GPL.

Tidak sama dengan proyek-proyek seperti Apache, di mana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, di mana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael “Monty” Widenius.

(Fikriansyah, 2017, Apa itu MySQL, Sejarah dan Fungsinya, <https://www.tutorialpedia.net/apa-itu-mysql/>)

2.2.8 JAVA

1. Sekilas Tentang Java

Java merupakan bagian dari bahasa pemrograman yang berorientasi objek. Mudah-mudahan, berbicara mengenai bahasa pemrograman java adalah sebuah bahasa yang berisi baris-baris perintah untuk berkomunikasi dengan komputer dan menghasilkan sebuah program. Ada banyak bahasa pemrograman, mulai dari bahasa pemrograman tingkat rendah sampai dengan bahasa pemrograman tingkat tinggi. Java adalah bahasa pemrograman yang tergolong kedalam bahasa pemrograman tingkat tinggi karena lebih mudah dioperasikan.

Sama seperti bahasa yang lain, java juga memiliki sintaks, tata bahasa dan aturan penulisan sendiri.

Java lahir pada tahun 1991, diciptakan oleh sebuah tim yang bernama green dan berjalan selama 18 bulan. Green dimotori oleh James Gosling, Mike Sherdian, Bill Joy dan Patrick Naughton bersama 9 programmer lainnya. Mereka adalah insinyur dari Sun Microsystems. Awalnya bahasa java diberi dengan nama "oak", yaitu nama dari sebuah pohon yang tumbuh didepan jendela ruang kerja James Gosling. Tetapi nama tersebut tidak digunakan karena sebelumnya sudah ada perangkat lunak yang dirilis dengan nama "oak". Akhirnya nama "oak" diganti dengan nama "java", nama ini diambil dari kopi murni yang digiling langsung dari biji kopi (kopi tubruk) kesukaan Gosling. Pada tahun 1996, java versi pertama dikeluarkan dengan nama rilis java 1.02. Beberapa fitur yang dimiliki oleh java diantaranya:

1. Java Virtual Machine / Cross Platform
2. Garbage Collection
3. Code Security

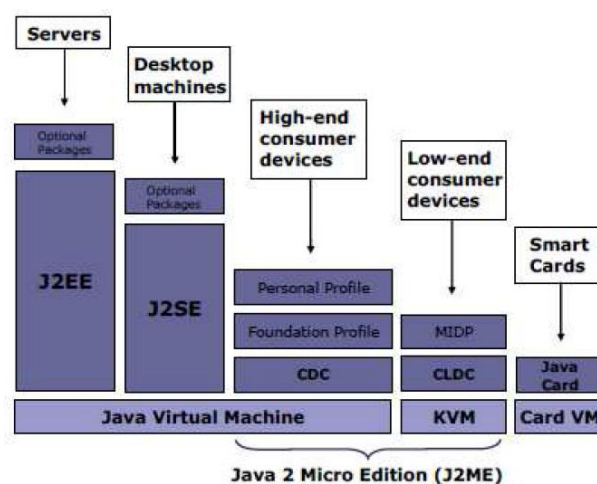
Dengan keluarnya versi 1.2, platform java telah dipilah-pilah menjadi beberapa edisi yaitu *The Standard Edition* (J2SE), *Enterprise Edition* (J2EE), *Mobile Edition* (J2ME), dan *JavaCard API*. Masing-masing edisi berisi Java 2 Software Development Kit (J2SDK) untuk mengembangkan aplikasi dan Java 2 Runtime Environment (J2RE) untuk

menjalankan aplikasi. Adapun kegunaan dari masing-masing edisi ini ditunjukkan pada tabel berikut:

Tabel 2.2 Kegunaan Dari Masing-masing Edisi

J2SE – Java 2 Platform, Standard Edition	Aplikasi Dekstop
J2EE – Java 2 Platform, Enterprise Edition	Aplikasi enterprise dengan fokus pada pengembangan sisi webserver, termasuk servlet, JSP, EJB dan XML
J2ME – Java 2 Platform, Micro Edition	Perangkat Mobile
JavaCards	Smart Cards

Adapun platformnya yang digamarkan pada JENI (Java Education Network Indonesia) adalah sebagai berikut :



Gambar 2.3 Platformn Pada JENI

2. Karakteristik Bahasa Pemrograman Berorientasi Objek (OOP)

Object Oriented Programing (OOP) adalah sebuah bahasa pemrograman yang memandang segala sesuatu menjadi sebuah objek. Pradigma dari OOP adalah menyelesaikan masalah dengan mempresentasikan masalah ke model objek. Karakteristik dari Objek Oriented Programing adalah :

c. *Encapsulation* / Enkapsulasi

Disebut juga dengan pembungkusan, yaitu melindungi program yang sedang diolah agar tidak diakses secara sembarangan oleh program

lain. Dalam Java dasar Enkapsulasi adalah *Class*. *Variable* atau *Method* pada sebuah *Class* tidak dapat diakses oleh *Class* lain dengan menjadikan *Class* tersebut menjadi *private*, atau menjadikan *class* tersebut menjadi *protected*, yang hanya bisa diakses oleh turunannya (*inheritance*) atau menjadikan *class* tersebut menjadi *public*, sehingga bisa diakses oleh sembarang *class*.

d. *Inheritance* / Inheritansi

Disebut juga dengan turunan. Prinsipnya adalah sebuah *class* dapat diturunkan dari *class* yang lain. *Class* yang menurunkan ke *class* yang lain disebut dengan *superclass*, *parent class* atau *base class* atau kelas induk, sedangkan *class* yang merupakan *class* turunan disebut sebagai *subclass*, *child class* atau *derived class* atau kelas turunan. *Class* turunan secara otomatis memiliki sifat (*variable*) dan kelakuan (*behavior*, *method*) yang dimiliki oleh *superclass*-nya. *class* turunan bisa menambahkan fitur atau *behavior* dengan mendefinisikan suatu *methode* di dalam *class* turunan tersebut

e. *Polimorphisme* / polimorfisme

Polimorfisme secara bahasa dapat diartikan dengan memiliki banyak bentuk. Kegunaan dari Polimorfisme adalah dapat mendefinisikan beberapa konstruktor atau beberapa metode dengan karakteristik yang berbeda-beda agar nantinya dapat digunakan untuk kasus-kasus yang berbeda. Method atau perilaku yang sama tapi implementasinya / caranya yang berbeda-beda.

3. Aplikasi Bahasa Pemrograman Java

- a. Notepad
- b. Textplus
- c. Editplus
- d. Jcreator
- e. Netbeans
- f. Crimson Editor
- g. Eclipse

- h. BlueJ
- i. Dr.Java
- j. JDeveloper

(Okkita Rizan, 2013, Bahasa Pemrograman Menggunakan JAVA Dengan Netbeans dan Ireport, 1)

2.2.9 Java Script Object Notation (JSON)

JSON adalah singkatan dari Javascript Object Notation atau Bahasa Indonesianya Notasi Objek Javascript. JSON juga bisa diartikan format file yang digunakan untuk transfer data baik membaca dan menulis. JSON sangat mudah bagi engine manapun untuk melakukan parsing dan generate sebuah data.

Banyak bahasa pemrograman yang mendukung JSON. JSON memiliki fungsi yang sama dengan XML yaitu berfungsi untuk merepresentasikan data. Jika dibanding dengan XML, JSON memiliki kelebihan yaitu mudah dibaca dan tidak perlu menulis tag JSON tetapi justru struktur JSON didefinisikan oleh data. Cara ini lebih baik dan lebih mudah difahami dan mempelajari JSON.

(Arya Febiyan ,2014, Pengertian JSON, <https://www.dumetschool.com/blog/Pengertian-JSON>)

2.2.10 Netbeans IDE

Netbeans adalah aplikasi Integrated Development Environment (IDE) yang berbasiskan Java Menurut Nofriadi (2015:4) "Netbeans Merupakan Sebuah aplikasi Integrated Development Environment (IDE) yang Menggunakan Bahasa Pemograman Java dari Sun Microsystems yang berjalan diatas swing", sedangkan menurut www.netbeans.org/index_id.html (2017:1) "NetBeans IDE adalah sebuah lingkungan pengembangan sebuah kakas untuk pemrogram menulis, mengompilasi, mencari kesalahan dan menyebarkan program. Netbeans IDE ditulis dalam Bahasa Pemograman Java, namun dapat mendukung bahasa pemrograman lain". Dari kedua pendapat diatas dapat ditarik

kesimpulan bahwa Netbeans adalah Suatu aplikasi untuk membuat suatu program yang dimana bahasa pemograman yang di pakai adalah java, maupun bahasa pemograman yang lain dengan mudahnya penggunaan aplikasi netbeans ini maka programmer sering menggunakan aplikasi ini sebagai wadah untuk membuat berbagai macam program yang dimana netbeans IDE ini bersifat Open Source.

(Pelangi, 10, Des, 2018 *NetBeans IDE*,
<http://pelangiilmu2.blogspot.com/2018/12/assalamualaikum-wr.html>)

2.2.11 *POSTMAN*

POSTMAN adalah sebuah aplikasi (berupa plugin) untuk browser chrome, fungsinya adalah sebagai *REST Client* atau istilahnya adalah aplikasi yang digunakan untuk melakukan uji coba REST API yang telah kita buat.

(Pranawa, 19, Maret, 2019, Step by Step Belajar REST API Menggunakan Flask di Windows (Instalasi POSTMAN),
<https://blog.pranawa.com/step-by-step-belajar-rest-api-menggunakan-flask-di-windows-instalasi-postman/>)

2.2.12 *KIBANA*

Kibana adalah alat visualisasi dan eksplorasi data open-source yang digunakan untuk analisis log dan time-series, pemantauan aplikasi, dan kasus penggunaan intelijen operasional. Kibana menawarkan fitur yang kuat dan mudah digunakan seperti histogram, grafik garis, diagram lingkaran, dan dukungan geospasial built-in dan menyediakan integrasi ketat dengan Elasticsearch, analitik populer dan mesin pencari, yang membuat Kibana menjadi pilihan default untuk memvisualisasikan data yang disimpan dalam Elasticsearch. Berikut fitur-fitur pada kibana:

a. Bagan Intuitif

Kibana menyediakan bagan dan laporan intuitif yang dapat Anda gunakan untuk menavigasi secara interaktif melalui data logo yang besar.

Anda dapat secara dinamis menyeret window, memperbesar dan memperkecil subset data tertentu, dan menelusuri laporan untuk mengekstrak wawasan yang dapat ditindaklanjuti dari data Anda.

b. Dukungan Pemetaan

Kibana hadir dengan kemampuan geospasial yang kuat sehingga Anda dapat dengan mudah melapisi informasi geografis di atas data Anda dan memvisualisasikan hasil pada peta.

c. Membangun Agregasi dan Filter

Menggunakan agregasi dan filter buatan Kibana, Anda dapat menjalankan berbagai analitik seperti histogram, kueri top-N, dan tren hanya dengan beberapa klik.

d. Dasbor yang mudah di akses.

Anda dapat dengan mudah mengatur dasbor dan laporan serta membaginya dengan yang lain. Yang Anda butuhkan hanyalah browser untuk melihat dan menjelajahi data.

(Point star, 2019, kibana, <https://www.pointstar.co.id/id/elastic-search-and-logging/kibana/>)

2.2.13 Pengertian dan Fungsi XAMPP.

XAMPP adalah program aplikasi pengembang yang berguna untuk pengembangan website berbasis PHP dan MySQL. Versi terbaru program ini adalah XAMPP 1.7.7, yang dirilis pada tanggal 20 September 2011. Software XAMPP dibuat dan dikembangkan oleh Apache Friends. Perangkat lunak komputer ini memiliki kelebihan untuk bisa berperan sebagai server web Apache untuk simulasi pengembangan website. Tool pengembangan web ini mendukung teknologi web populer seperti PHP, MySQL, dan Perl. Melalui program ini, programmer web dapat menguji aplikasi web yang dikembangkan dan mempresentasikannya ke pihak lain secara langsung dari komputer, tanpa perlu terkoneksi ke internet.

XAMPP juga dilengkapi fitur manajemen database PHPMyAdmin seperti pada server hosting sungguhan, sehingga pengembang web dapat mengembangkan aplikasi web berbasis database secara mudah. Program XAMPP banyak diaplikasikan dan digunakan oleh kalangan pengguna komputer di bidang pemrograman web. XAMPP merupakan software gratis. XAMPP dapat dijalankan di sistem operasi Windows 2000/XP/Vista/7/10 dan sistem operasi lain. XAMPP bisa dijalankan di 4 OS besar yang sering digunakan oleh pengguna komputer saat ini. Dan 4 OS tersebut tidak lain dan tidak bukan adalah Windows, Linux, Mac OS dan Solaris.

(Tester, S. (2013, Maret). *Penjelasan dan fungsi xampp*. Retrieved September 6, 2017, from <http://solylight.blogspot.co.id:http://solylight.blogspot.co.id/2013/03/penjelasan-dan-fungsi-xampp.html>).

2.2.14 Model Driven Development (MDD)

Model Driven Development (MDD) merupakan suatu teknik yang menekankan pembuatan gambar model-model sistem untuk membantu visualisasi dan analisis masalah, mendefinisikan persyaratan bisnis, dan mendesain sistem informasi. Model sistem adalah gambar sebuah sistem yang mewakili realitas atau realitas yang diharapkan.

Keunggulan model-driven:

1. Spesifikasi persyaratan lebih menyeluruh dan didokumentasikan dengan baik
2. Persyaratan bisnis dan desain sistem lebih mudah divalidasi dengan gambar daripada dengan kata-kata
3. Lebih mudah mengidentifikasi, mengkonseptualkan, dan menganalisis solusi-solusi teknis alternatif
4. Spesifikasi desain cenderung solid, stabil, dapat beradaptasi, dan fleksibel karena berbasis model dan dianalisis lebih menyeluruh sebelum dibangun
5. Sistem dapat dikonstruksikan dengan lebih tepat pertama kali saat dibangun dari spesifikasi berbasis model yang menyeluruh dan jelas.

Kelemahan model-driven:

1. Banyak memakan waktu
2. Model tersebut dapat sebagai pemahaman para pengguna akan persyaratan tersebut.
3. Model bukanlah perangkat lunak
4. Tidak fleksibel.

(Ttisinadi, 2019, Model-driven development, <http://trisinadi169.blogspot.com/search/label/modeldriven%20development>)

2.2.15 Clickchart Free Diagram And Flowchart Marker

ClickCharts Free Diagram And Flowchart Marker adalah aplikasi yang sempurna untuk membantu anda merancang pilar dasar dari sebuah basis data dan juga mengatur jenis struktur apapun menggunakan diagram.

Anda bisa membuat jenis diagram apa saja menggunakan alat yang lengkap ini; mulai dari bagan dengan simbol-simbol tertentu hingga UML (Unified Modeling Language) dengan sistem penyusunan dan penempatan bagan secara otomatis dengan sangat mudah. Tetapi, jika sistem pengaturan cepat ini tidak sesuai dengan yang anda inginkan, anda bisa mengedit posisi bagan diagram tersebut secara manual. Membuat mata rantai dengan penghubung-penghubung yang benar-benar bisa disesuaikan, mengubah karakteristik diagram tersebut sesuai dengan keinginan anda dan meningkatkan kualitas diagram anda dengan menambahkan gambar-gambar, ilustrasi, dan filter-filter yang ditawarkan oleh ClickCharts Free Diagram And Flowchart Marker untuk memberikan sebuah tampilan yang lebih profesional.

ClickCharts Free Diagram And Flowchart Marker membolehkan anda bekerja dengan diagram yang terdiri dari beberapa bagian sekaligus dan juga menyalin dan menempel elemen-elemen di dalamnya. Setelah anda selesai dengan tugas ini, anda bisa memindahkan hasil akhirnya ke format gambar apapun atau mencetaknya langsung dalam resolusi tinggi dari antarmuka program ini.

(Uptodown, Membuat diagram-diagram rumit dengan cara tercepat dan termudah , <https://clickcharts-free-diagram-and-flowchart-maker.id.uptodown.com/windows>)

2.2.16 Balsamic mockup.

Mockup artinya model atau replika mesin atau struktur, yang digunakan untuk tujuan instruksional atau eksperimental.

Balsamiq mockup adalah program aplikasi yang digunakan dalam pembuatan tampilan user interface sebuah aplikasi. Software ini sudah menyediakan tools yang dapat memudahkan dalam membuat desain prototyping aplikasi yang akan kita buat. Software ini berfokus pada konten yang ingin digambar dan fungsionalitas yang dibutuhkan oleh pengguna.

Alih-alih menggambar sketsa (wireframe) atau prototype rancangan desain website di atas kertas balsamiq mockups membantu seorang web desainer membuat tampilan web dalam bentuk gambar di komputer. Tujuannya selain agar membuat tampilan (desain) website menarik juga dapat menyesuaikan dengan kebutuhan customer (pelanggan). Dengan alat pembuat mockup maka seorang web desainer dapat menganalisa tata letak, desain dan fungsi.

Kelebihan Balsamiq Mockups dibanding software pembuat mockup lainnya adalah aplikasi ini berbasis cloud, disertai aplikasi desktop yang memungkinkan kita dengan cepat dan mudah membuat rancangan website. Dengan konten yang terbuat seperti dari gambaran tangan, akan membuat kita fokus pada pemecahan masalah user interface yang lebih besar, daripada pada perincian website. Di websitenya sendiri ada dua pilihan untuk para pengguna, ada versi trial for dekstop dan ada juga yang bisa kita download untuk versi dekstop. Namun ada juga yang disediakan dalam versi berbayar. Aplikasi ini bisa digunakan untuk sistem operasi Windows, Mac OS, dan Linux.

(Erlangga, 2016/01/07 , Balsamiq Mockup , <http://enggar.net/2016/01/balsamiq-mockup/>)

2.2.17 UML

UML (*Unified Modeling Language*) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun, dan endokumentasian dari sebuah sistem pengembangan software berbasis OO (*Object-Oriented*). UML sendiri juga memberikan standar penulisan sebuah sistem blue print, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema *database*, dan komponen-komponen yang diperlukan dalam sistem *software*. Ada beberapa jenis UML yaitu *Use Case Diagram*, *Activity Diagram*, *Package Diagram*, *State Diagram*, *Sequence Diagram*, *Class Diagram*, *Communication Diagram*, *Composite Structure Diagram*, *Object Diagram*, *Timing Diagram*, *Component Diagram*, *Deployment Diagram*, *Interaction Overview Diagram*.

Jenis-jenis Diagram UML, yaitu :

1. *Use Case Diagram*

Use case adalah abstraksi dari interaksi antara system dan *actor*. *Use case* bekerja dengan cara mendeskripsikan tipe interaksi antara user sebuah system dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah system dipakai.

2. *Activity Diagram*

Activity diagram menyediakan analisis dengan kemampuan untuk memodelkan proses dalam suatu sistem informasi. *Activity diagram* dapat digunakan untuk alur kerja model, *use case individual*, atau logika keputusan yang terkandung dalam metode individual³. *Activity diagram* juga menyediakan pendekatan untuk proses pemodelan paralel. *Activity diagram* lebih lanjut. Pada dasarnya, diagram aktifitas canggih dan merupakan diagram aliran data yang terbaru. Secara teknis, diagram aktivitas menggabungkan ide-ide proses pemodelan dengan teknik yang berbeda termasuk model acara, statecharts, dan Petri Nets.

3. *Package Diagram*

Package diagram utamanya digunakan untuk mengelompokkan elemen diagram UML yang berlainan secara bersama-sama ke dalam tingkat pembangunan yang lebih tinggi yaitu berupa sebuah paket. Diagram paket pada dasarnya adalah diagram kelas yang hanya menampilkan paket, disamping kelas, dan hubungan ketergantungan, disamping hubungan khas yang ditampilkan pada diagram kelas. Sebagai contoh, jika kita memiliki sistem pendaftaran untuk kantor dokter, mungkin masuk akal untuk kelompok kelas pasien dengan kelas sejarah medis pasien bersama-sama untuk membentuk paket kelas pasien. Selain itu, dapat berguna untuk membuat paket perawatan yang mengandung gejala penyakit, penyakit, dan obat-obatan khas yang diresepkan untuk mereka.

4. *State Diagram*

State diagram menggambarkan urutan keadaan yang dilalui objek dalam suatu kelas, karena suatu kejadian menyebabkan suatu perpindahan aktivitas/state. State dari objek adalah penggolongan dari satu atau lebih nilai attribute pada kelas. Bersifat dinamis. Diagram state ini memperlihatkan statestate pada system, memuat state, transisi, event, serta aktifitas. Diagram ini terutama penting untuk memperlihatkan sifat dinamis dari antarmuka, kelas, kolaborasi.

5. *Sequence Diagram*

Sequence diagram menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya *sequence diagram* adalah gambaran tahap demi tahap yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case diagram*. Bersifat dinamis. Diagram urutan adalah interaksi yang menekankan pada pengiriman pesan (message) dalam suatu waktu tertentu. *Sequence diagram* menekankan penyusunan berbasis

waktu untuk kegiatan yang dilakukan dengan satu set dari objek yang berkolaborasi. Sequence diagram sangat berguna dalam membantu analis, memahami spesifikasi *real-time* dan menggunakan kasus yang rumit (lihat di bawah). Diagram ini dapat digunakan untuk mendeskripsikan baik secara fisik dan logis interaksi antara objek.

6. *Class Diagram*

Class Diagram adalah dekripsi kelompok obyek-obyek dengan property, perilaku (operasi) dan relasi yang sama. Sehingga dengan adanya class diagram dapat memberikan pandangan global atas sebuah system. Hal tersebut tercermin dari class- class yang ada dan relasinya satu dengan yang lainnya. Sebuah sistem biasanya mempunyai beberapa class diagram. *Class* diagram sangat membantu dalam visualisasi struktur kelas dari suatu *system*. Bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada pemodelan system berorientasi objek. Kelas Diagram berfungsi untuk menjelaskan tipe dari object sistem dan hubungannya dengan *object* yang lain. *Object* adalah nilai tertentu dari setiap attribute kelas *entity*. Pada penggambaran kelas diagram ada dikenal dengan

7. *Communication Diagram*

Communication diagram menggambarkan interaksi antar objek seperti sequence diagram, tetapi lebih menekankan pada peran masing-masing objek. Setiap message memiliki *sequence number*, dimana message dari level tertinggi memiliki Nomor 1. Diagram membawa informasi yang sama dengan diagram Sequence, tetapi lebih memusatkan atau memfokuskan pada kegiatan obyek dari waktu pesan itu dikirimkan.

8. *Composite Structure Diagram*

Diagram struktur komposit adalah diagram yang menunjukkan struktur internal *classifier*, termasuk poin interaksinya ke bagian lain dari system. Hal ini menunjukkan konfigurasi dan hubungan bagian, yang bersama-sama melakukan perilaku *classifier*. Diagram struktur komposit merupakan jenis diagram struktur

9. *Object Diagram*

Object diagram merupakan sebuah gambaran tentang objek-objek dalam sebuah system pada satu titik waktu. Karena lebih menonjolkan perintah-perintah dari pada class, object diagram lebih sering disebut sebagai sebuah diagram perintah. *Object diagram* sangat mirip dengan diagram kelas. Perbedaan utama adalah bahwa diagram objek menggambarkan objek dan hubungan mereka. Tujuan utama dari diagram objek adalah untuk memungkinkan analis untuk mengungkap rincian tambahan kelas. Dalam beberapa kasus, pernyataan variabel dari sebuah class diagram dapat membantu pengguna atau analis dalam menemukan atribut tambahan yang relevan, hubungan, dan atau operasi, atau mungkin menemukan bahwa beberapa atribut, hubungan, atau operasi yang salah tempat Bersifat statis. Diagram objek memperlihatkan instansiasi statis dari segala sesuatu yang dijumpai pada diagram kelas.

10. *Timing Diagram*

Memperlihatkan interaksi ketika tujuan utama diagram adalah waktu. Menggambarkan perubahan dalam state atau kondisi dari pengelompokaninstance atau tugas berlebihan. Biasanya dipakai untuk memperlihatkan perubahan dalam state objectberlebihan dalam merespon ke external events. Dipakai untuk memperlihatkan perilaku dari sebuah/ beberapa *object* melaluperiode waktu. Diagram tersebut akan menjadi ideal ketika

kita mampu menyeimbangkan ke-6 elemen yang ada, bukan menariknya ke satu atau dua arah saja. Tiap orang biasanya punya satu elemen yang dominan, tinggal bagaimana mengoptimalkan elemen-elemen yang lain saja.

11. *Component Diagram*

Diagram ini bila dikombinasikan dengan diagram penyebaran dapat digunakan untuk menggambarkan distribusi fisik dari modul perangkat lunak melalui jaringan. Misalnya, ketika merancang sistem *client-server*, hal ini berguna untuk menunjukkan mana kelas atau paket kelas akan berada pada node klien dan mana yang akan berada di server. Diagram komponen juga dapat berguna dalam merancang dan mengembangkan sistem berbasis komponen. Karena berfokus pada analisis sistem berorientasi objek dan desain.

12. *Deployment Diagram*

Deployment diagram menggambarkan detail bagaimana komponen di deploy dalam infrastruktur *system*, dimana komponen akan terletak (pada mesin, server atau piranti keras), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisik. Hubungan antar node (misalnya TCP/IP) dan requirement dapat juga didefinisikan dalam diagram ini.

13. *Interaction Overview Diagram*

Interaction Overview Diagram adalah pecangkolan secara bersama antara *activity diagram* dengan *sequence diagram*. *Interaction Overview Diagram* dapat dianggap sebagai *activity diagram* dimana semua aktivitas diganti dengan sedikit *sequence diagram*, atau bisa juga dianggap sebagai *sequence diagram* yang dirincikan dengan notasi *activity diagram* yang digunakan untuk menunjukkan aliran pengawasan.

(Elya Hestika Asiyah, 2014, SEJARAH UML DAN JENISNYA, Jurnal Ilmu Teknologi Informasi)

2.2.18 Javafx Scene Builder

Javafx Scene Builder atau lebih singkatnya Screen Builder adalah Tools yang digunakan untuk membuat desain User Interface dari sebuah aplikasi. Javafx adalah penerus dari java swing, digunakan untuk membuat aplikasi java berbasis Grafical User Interface (GUI). Scene builder dapat membuat file fxml secara otomatis. Seperti yang kita ketahui, fxml adalah file yang berformat xml, yang isinya adalah layout dari aplikasi javafx.

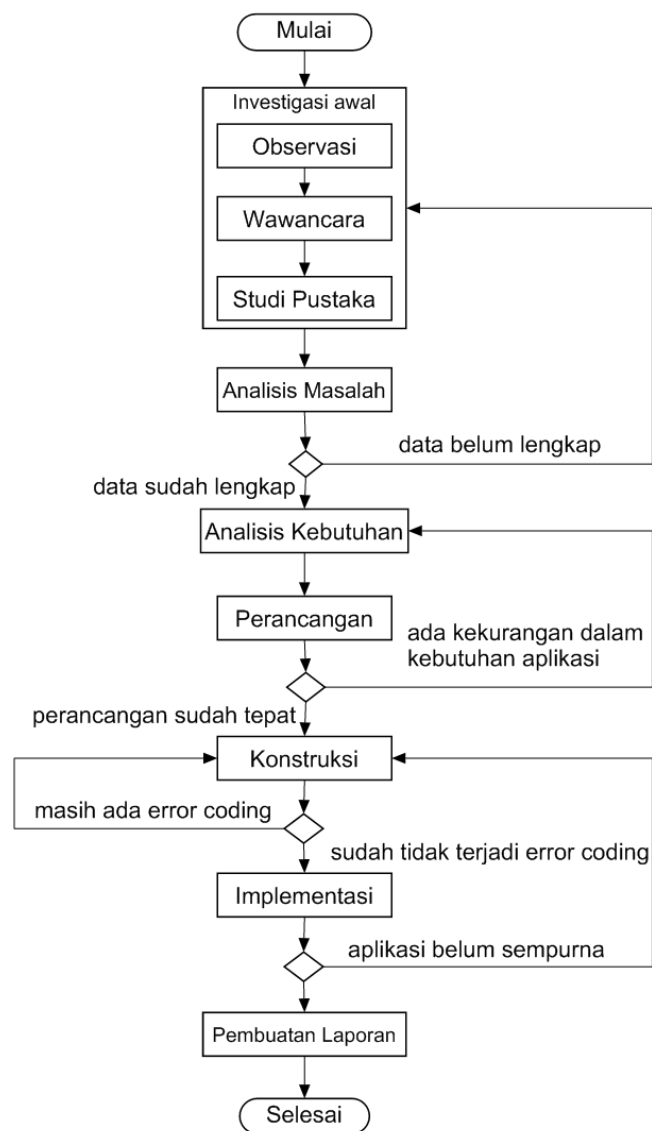
(Sigit Kurniawan, 2014, Javafx Scene Builder, <https://sigitkurniawanisme.wordpress.com/2015/01/02/javafx-scene-builder/>)

BAB III

METODOLOGI PENELITIAN

Bab ini berisi gambaran kerangka pikir dan deskripsinya. Adapun langkah langkah yang di lakukan untuk mencapai tujuan dari penelitian sebagai berikut :

3.1 Kerangka pikir



Gambar 3.1 Kerangka pikir

3.2 Deskripsi Kerangka Pikir

3.2.1 Investigasi Awal

Penelitian ini merupakan kolaborasi dari sebuah *project* pengembangan sistem informasi *repository* skripsi di Fakultas Teknologi Informasi. Penelitian ini terdiri dari penelitian berjudul “Implementasi Elasticsearch Untuk Pengelolaan Repository Karya Ilmiah Fakultas Teknologi Informasi Universitas Bale Bandung” yang lebih menekankan Pada Penyimpanan (*Storaging*) dan *Indexing* Dan Penelitian “Implementasi Elasticsearch Untuk Pencarian Dan Menentukan Skor Similarity Pada Proposal Skripsi Di Fakultas Teknologi Informasi Universitas Bale Bandung” yang lebih menekankan pada pencarian dan menentukan kemiripan (*similarity*) pada suatu dokumen.

Project ini diimplementasikan di Fakultas Teknologi Informasi Universitas Bale Bandung yang beralamat di Jl. Raden AA Wiranatakusumah No.7, Baleendah, Bandung, Jawa Barat 40375.

dalam tahap investigasi awal dilakukan investigasi berupa :

1. Observasi

Observasi dilakukan dengan mengamati sistem yang sedang berjalan saat ini bertujuan untuk mengetahui secara langsung bagaimana proses pengelolaan dokumen skripsi yang sedang berjalan.

Pada tahap observasi peneliti mengamati proses penyimpanan koleksi karya ilmiah yang hanya disimpan di rak buku sehingga sangat rentan mengalami kerusakan pada dokumen tersebut. Peneliti juga memperhatikan sistem pengelolaan data skripsi yang hanya sebatas pengetikan seluruh data skripsi yang telah di persiapkan sebelumnya dengan hanya menggunakan aplikasi Ms-Excel.

2. Wawancara

Pada tahap wawancara dilakukan pengumpulan data-data di Fakultas Teknologi Informasi UNIBBA khususnya di divisi Tata Usaha yang akan di jadikan bahan dasar dalam perancangan aplikasi dengan cara menemui staff Tata Usaha yang mengelola perpustakaan dan

menemui Dekan FTI untuk mencari informasi mengenai permasalahan yang sering terjadi di Fakultas Teknologi Informasi dan untuk meminta izin melakukan penelitian dan melakukan wawancara tentang masalah yang ada di bagian tata usaha. Dari wawancara yang dilakukan penulis mendapatkan Beberapa permasalahan yang sering timbul pada proses dokumen yang diantaranya adalah:

- a) Sulitnya temu kenali dokumen karya ilmiah baik dokumen berkas fisik maupun berkas digital, karena penyimpanan yang belum terindeks.
- b) Tidak tersedianya cloud storage yang berfungsi sebagai media penyimpanan berkas digital (Repository).
- c) Dalam proses pengelolaan dokumen skripsi Hanya menggunakan aplikasi Ms-Excel, sehingga didalam menghasilkan seluruh laporan yang akurat dan tepat relatif lama serta kurang lengkapnya laporan yang dihasilkan.
- d) banyak skripsi yang belum di sahkan oleh penguji dan pembimbing tetapi sudah tersimpan di rak sebelum di tandatangani oleh pembimbing, dekan dan ketua prodi.

3. Studi Pustaka

Penulis mencari beberapa jurnal terkait objek penelitian sebagai penunjang dan juga mencari beberapa referensi dari website, jurnal, dan buku digital (*ebook*) yang berkaitan dengan masalah yang ada di Fakultas Teknologi Informasi Universitas Bale Bandung. Berikut adalah judul dari referensi:

- a. Pengelolaan Institutional Repository Di Perpustakaan Utsman Bin Affan Universitas Muslim Indonesial (Nurhasanah,2017, jurnal ilmu perpustakaan)

Dalam jurnal ini Penulis mencoba menggunakan enggine yang lain yaitu menggunakan Elasticsearch sebagai database stand alone dengan harapan dapat melakukan akses penyimpanan

dokumen lebih cepat dan dapat di implementasikan juga dalam tahap penentuan similarity dokumen skripsi.

- b. Sistem Informasi Pengarsipan Dokumen Berbasis Web (Agustina Simangunsong, jurnal Mantik Penusa, 2018, Volume 2 No. 1)

Dalam jurnal ini penulis mencoba untuk menggunakan aplikasi yang berbasis client server sehingga arus informasi menjadi lebih cepat serta dan dapat menjaga keamanan data secara sentral, langsung dari servernya.

- c. Perbandingan Kemampuan Database NoSQL dan SQL dalam Kasus ERP Retail (Faizal Anugrah Bhaswara, 2017, Jurnal Teknik ITS Vol. 6, No. 2).

Pada penelitian ini penulis mencoba untuk menerapkan database dengan konsep NoSQL Tersebut pada proyek yang akan di buat, karena NoSQL terbukti memiliki kecepatan penyimpanan data yang lebih unggul dalam hal CRUD dari pada SQL.

3.2.2 Analisis Masalah

Dari fakta-fakta yang sudah di temukan maka tahap selanjutnya adalah melakukan analisis masalah. Berikut adalah hasil dari analisis masalah yang di lakukan :

- a. Sulitnya temu kenali dokumen karya ilmiah baik dokumen berkas fisik maupun berkas digital, karena penyimpanan yang belum terindeks.
- b. Tidak tersedianya cloud storage yang berfungsi sebagai media penyimpanan berkas digital (Repository).
- c. Didalam menghasilkan seluruh laporan yang akurat dan tepat relatif lama serta kurang lengkapnya laporan yang dihasilkan karena, proses pengelolaan dokumen skripsi Hanya menggunakan aplikasi Ms-Excel.

Berdasarkan hasil dari analisis masalah yang di lakukan maka di lakukanlah studi banding dengan pihak yang lebih berpengalaman, yaitu kepada bapak Moch. Ridwan, S.T., M.KOM dan sodara Hendra selaku alumni dari Fakultas Teknologi Informasi UNIBBA bahwa *elasticsearch*

yang lebih sesuai berdasarkan hasil survey, dan di sarankan pula menggunakan Kibana Untuk memudahkan dalam manajemen *elasticsearch*, mengembangkan aplikasi untuk *elasticsearch* dan juga membantu pelaporan yang dapat memvisualisasikan dalam bentuk grafik .

Jika di dalam tahap ini masih ada kekurangan maka akan kembali ke tahap investigasi awal sampai mendapat hasil yang tepat, kemudian di lanjutkan ketahap berikutnya.

3.2.3 Analisis Kebutuhan

1. Kebutuhan Prosedur

Prosedur sebelumnya mahasiswa mengumpulkan laporan skripsi dalam bentuk CD dan juga *hardcopy* untuk pengumpulan laporan juga mahasiswa diwajibkan untuk membuat file laporan skripsi dengan format pdf berukuran 1 MB, kemudian mengunggah atau menyimpan laporan dengan file pdf ke *repository*.

2. Aplikasi Utama

Berikut adalah analisis kebutuhan yang di perlukan untuk menjawab semua pertanyaan yang ada di Fakultas Teknologi Informasi Universitas Bale Bandung, yaitu Membuat User Interface yang dapat melakukan proses penyimpanan dokumen karya ilmiah yang memiliki performa pengaksesan data lebih cepat.

3. Perangkat Lunak Pendukung

Pembuat aplikasi utama dibutuhkan menggunakan aplikasi atau perangkat pendukung yakni sebagai berikut:

- a. *Elasticsearch* adalah *engine* yang digunakan untuk mengindex dan penyimpanan dokumen.
- b. *Kibana*, digunakan sebagai visualisasi, penggunaan *command* API, pengujian API, dan berinteraksi dengan data yang disimpan dalam indeks *Elasticsearch*.
- c. *Postman*, digunakan untuk pengujian API yang cara kerjanya sama seperti *kibana*.

- d. *Database MySQL*, digunakan untuk penyimpanan *database* dari atribut skripsi, author, dan jurnal publish.
 - e. *Java* adalah bahasa pemrograman yang digunakan.
 - f. NetBeans IDE, digunakan untuk membuat dan mengembangkan sebuah aplikasi dengan bahasa java..
 - g. *Apache Maven*, merupakan tools yang digunakan sebagai konfigurasi proyek aplikasi dengan bahasa Java.
 - h. Scene Builder tools untuk membuat desain UserInterface.
4. Perangkat Keras Pendukung

Setelah melakukan analisis kebutuhan langkah berikutnya adalah melakukan instalasi *elasticsearch* server dan kibana yang berperan membantu manajemen *elasticsearch* dan mengembangkan aplikasi *elasticsearch* di laptop dengan spesifikasi minimal sebagai berikut:

- a. AMD E1-6010 APU with AMD Radeon R2 Graphics (2 CPUs), 1.4GHz
- b. RAM 4 GB
- c. HDD 500 GB

dalam penelitian inipun penulis menggunakan spesifikasi perangkat keras minimal tersebut.

3.2.4 Desain

1. Perancangan Aplikasi

Pada tahap ini perancangan di lakukan dengan membuat *Unified Modeling Language* (UML). Perancangan dengan *Unified Modeling Language* (UML) ini bertujuan untuk merancang aplikasi dengan mudah dalam proses pengembangan dan juga untuk memvisualisasikan, membangun dan mendokumentasikan dari sebuah sistem pembangunan perangkat lunak berbasis objek, maka dibuatlah:

- a. *Use Case* untuk penyimpanan dokumen dan melengkapi indeks pada dokumen.

- b. *Class Diagram* yaitu diagram untuk menggambarkan struktur dan penjelasan hubungan antar class dalam sebuah sistem yang dibuat yaitu sistem login, penyimpanan dokumen dan melengkapi indeks dokumen.
- c. *Activity Diagram* yaitu diagram untuk menggambarkan alur aktivitas yang sedang di rancang yang diantaranya *Activity Diagram* untuk login , *Activity Diagram* Tambah Akun, *Activity Diagram* Reset Akun, *Activity Diagram* tambah dokumen, *Activity Diagram* Hapus dokumen, *Activity Diagram* Edit Data dokumen, *Activity diagram* pencarian judul dokumen, *Activity Diagram* untuk Lengkapi indeks dokumen.
- d. *Sequence Diagram* yaitu diagram untuk kemunculan menu aplikasi, diantaranya *sequence diagram* login, *sequence diagram* Tambah Akun, *sequence diagram* Reset Akun, *sequence diagram* Tambah dokumen, *sequence diagram* Hapus dokumen, , *sequence diagram* Edit data dokumen, *sequence diagram* pencarian judul dokumen dan *sequence diagram* Lengkapi indeks dokumen.

2. Perancangan Database

Selanjutnya adalah dilakukan perancangan *database* MySQL yang bertujuan untuk menyimpan data admin dan mahasiswa untuk proses Login .

Jika dalam tahap ini terdapat kesalahan atau kekurangan maka akan kembali ke tahap sebelumnya sampai mendapat hasil yang tepat, selanjutnya baru di lanjutkan ke tahap berikutnya

3.2.5 Konstruksi

Setelah tahap *Design* (Perancangan) selesai maka tahap selanjutnya adalah pembuatan aplikasi. Pada pembuatan aplikasi digunakan perangkat lunak dan bahasa pemrograman sebagai berikut:

1. *Elasticsearch*, digunakan untuk mengindex dokumen dan penyimpanan dokumen.

2. *Kibana*, digunakan sebagai visualisasi, penggunaan *command* API, pengujian API, dan berinteraksi dengan data yang disimpan dalam indeks *Elasticsearch*.
3. *Postman*, digunakan untuk pengujian API yang cara kerjanya sama seperti *kibana*.
4. *Database MySQL*, digunakan untuk penyimpanan *database* dari entitas skripsi, author, dan jurnal publish.
5. *Java* adalah bahasa pemrograman yang digunakan.
6. *NetBeans IDE*, digunakan untuk membuat dan mengembangkan sebuah aplikasi dengan bahasa java..
7. *Apache Maven*, merupakan tools yang digunakan sebagai konfigurasi proyek aplikasi dengan bahasa Java.

3.2.6 Implementasi

Sebelum implementasi dilakukan maka di awali melakukan penelitian terlebih dahulu terhadap aplikasi yang sudah dibuat. Pengujian terhadap aplikasi yang sudah dibuat, yaitu dilakukan pengujian dengan menggunakan metode *black box* yaitu untuk menguji fungsionalitas dari suatu aplikasi.

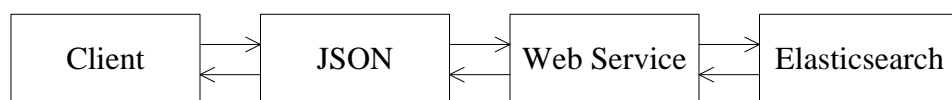
Dalam pengujian melalui beberapa tahapan:

1. Tahap Pertama
 - a. Pengujian Pertama, *Elasticsearch* dijalankan untuk ujicoba dengan cara melakukan *HTTP Request* untuk mengetahui keaktifan *Elasticsearch*.
 - b. Pegujian Kedua, mengirim JSON ke *Elasticsearch* untuk mengetes versi
2. Tahap Kedua

Dilakukan uji coba terhadap form yang dibuat, apakah sudah bisa mengeluarkan teks atau tidak, jika sudah maka langkah selanjutnya di uji coba dengan teks di konversi menjadi format JSON dengan menggunakan modul JSON.
3. Tahap Ketiga

- a. Pengujian *web service* untuk mengirim request ke *Elasticsearch* secara koding sederhana untuk mengeluarkan format JSON.
- b. Pengujian JSON di kirim ke *web service* kemudian akan mengirim ke *Elasticsearch*.
- c. Pengujian mengkonversi json dalam bentuk format visual yang bisa dibaca.

Berikut ini adalah blok-blok yang di buat untuk memudahkan pemahaman dalam pengujian.



Gambar 3.2 Blok Pengujian

Jika di dalam tahap ini masih ada kekurangan maka akan kembali ke tahap kontruksi sampai mendapat hasil *coding* yang tepat, kemudian di lanjutkan ketahap berikutnya.

3.2.7 Pembuatan Laporan

Melakukan pembuatan laporan dalam bentuk dokumen dari hasil penelitian yang dilakukan. Berikut sistematika penulisan laporan:

BAB I PENDAHULUAN

Pada bab ini dijelaskan beberapa hal yang mendasari atau yang melatarbelakangi penulisan tugas akhir ini. Di dalam bab I ini dideskripsikan tentang latar belakang, Rumusan Masalah, Batasan Masalah, Tujuan, Metodologi, dan sistemtematika penulisan yang digunakan.

BAB II LANDASAN TEORI

Uraian dan penjelasan teori yang berhubungan dengan judul yaitu tentang aplikasi Implementasi Elasticsearch Untuk Pengelolaan Repository Karya Ilmiah Fakultas Teknologi Informasi UNIBBA dan beberapa hal mengenai permasalahan konsep dasar yang dijadikan sebagai acuan di dalam penyusunan tugas akhir ini. Lengkapnya dalam bab ini dijelaskan beberapa teori pendukung yang diperlukan dalam penyusunan tugas akhir.

BAB III METODOLOGI

Bab ini membahas tentang waktu dan tempat penelitian, alat dan bahan, metode pengumpulan data dan metode perancangan.

BAB IV ANALISA DAN PERANCANGAN SISTEM

Pada bab ini berisi penjelasan mengenai analisis yang di gunakan dan juga pemodelan sistem yang di gunakan.

BAB V IMPLEMENTASI DAN PENGUJIAN

Bab ini berisi pengujian serta hasil pengamatan secara keseluruhan pada aplikasi Implementasi Elasticsearch Untuk Pengelolaan Repository Karya Ilmiah Fakultas Teknologi Informasi Universitas Bale Bandung.

Bab VI Kesimpulan Dan Saran

Bab ini akan diuraikan kesimpulan dan hasil analisis yang dapat memberikan saran-saran kepada Fakultas Teknologi Informasi Universitas Bale Bandung.

BAB IV

ANALISIS DAN PERANCANGAN

4.1 Analisis

Analisis merupakan tahap untuk pemahaman terhadap aplikasi yang akan dibuat. Pada tahap ini berisi Instrumen Penelitian, Analisis Sistem, Analisis Kebutuhan, Hasil Analisis dan Perancangan, dengan maksud untuk mengidentifikasi segala permasalahan yang terjadi. Analisis juga memiliki tujuan untuk mengetahui bagaimana mekanisme sistem bekerja, mekanisme yang terlibat dalam sistem serta hubungan antara mekanisme tersebut.

Analisis yang dilakukan dalam proses penelitian ini meliputi beberapa tahapan yaitu Instrumen Penelitian, Analisis Sistem, Analisis Kebutuhan dan Hasil Analisis.

4.1.1 Instrumen Penelitian

Instrumen Penelitian merupakan alat bantu yang digunakan untuk membantu mengumpulkan data pada sebuah penelitian dilakukan dengan observasi, wawancara, studi pustaka dan pembuatan laporan diperlukan sebuah alat bantu sebagai instrument. Instrumen yang digunakan dalam penelitian ini yaitu meliputi Perangkat Keras, Perangkat Lunak dan Form Wawancara:

1. Perangkat Keras

Perangkat keras yang digunakan dalam mengumpulkan data khususnya pada saat merekam wawancara yaitu menggunakan smartphone, Berikut ini merupakan instrumen penelitian pada perangkat keras yang dapat menjadi penunjang dalam proses penelitian, Berikut spesifikasi Smartphone yang digunakan:

Type	: Samsung J2 Core
OS	: Android Oreo

RAM : 1,5 GB
 Penyimpanan : 16 GB

2. Perangkat Lunak

Berikut ini merupakan instrumen penelitian pada perangkat lunak yang dapat menjadi penunjang dalam penelitian, berikut ini adalah beberapa perangkat lunak yang digunakan :

- 1) Sistem Operasi menggunakan Windows 7 64 bit
- 2) Microsoft Word 2013
- 3) Web server menggunakan XAMPP V3.2.2
- 4) *Code Editor* menggunakan Netbeans IDE 8.2
- 5) Web server menggunakan XAMPP V3.2.2
- 6) Database menggunakan MySQL
- 7) Penyimpanan dokumen menggunakan Elasticsearch 6.5.4.
- 8) REST Client API menggunakan Postman dan Kibana 6.5.4.
- 9) Pemodelan UML menggunakan ClickChart Diagram
- 10) Pemodelan *user interface* menggunakan Balsamiq Mockup 3 .
- 11) Pemodelan desain UI Menggunakan Scene builder.

3. Wawancara

Wawancara dilakukan peneliti dengan datang ke ruang Tata Usaha Fakultas Teknologi Informasi Universitas Bale Bandung, dan penulis mewawancarai staff yang mengelola perangkat perpustakaan di fakultas teknologi informasi menurut penjelasannya dalam pengelolaan dokumen skripsi masih berupa penginputan data skripsi kedalam sebuah aplikasi Mic-Excel dan belum di sediakannya sebuah aplikasi penyimpanan berkas digital atau repository.

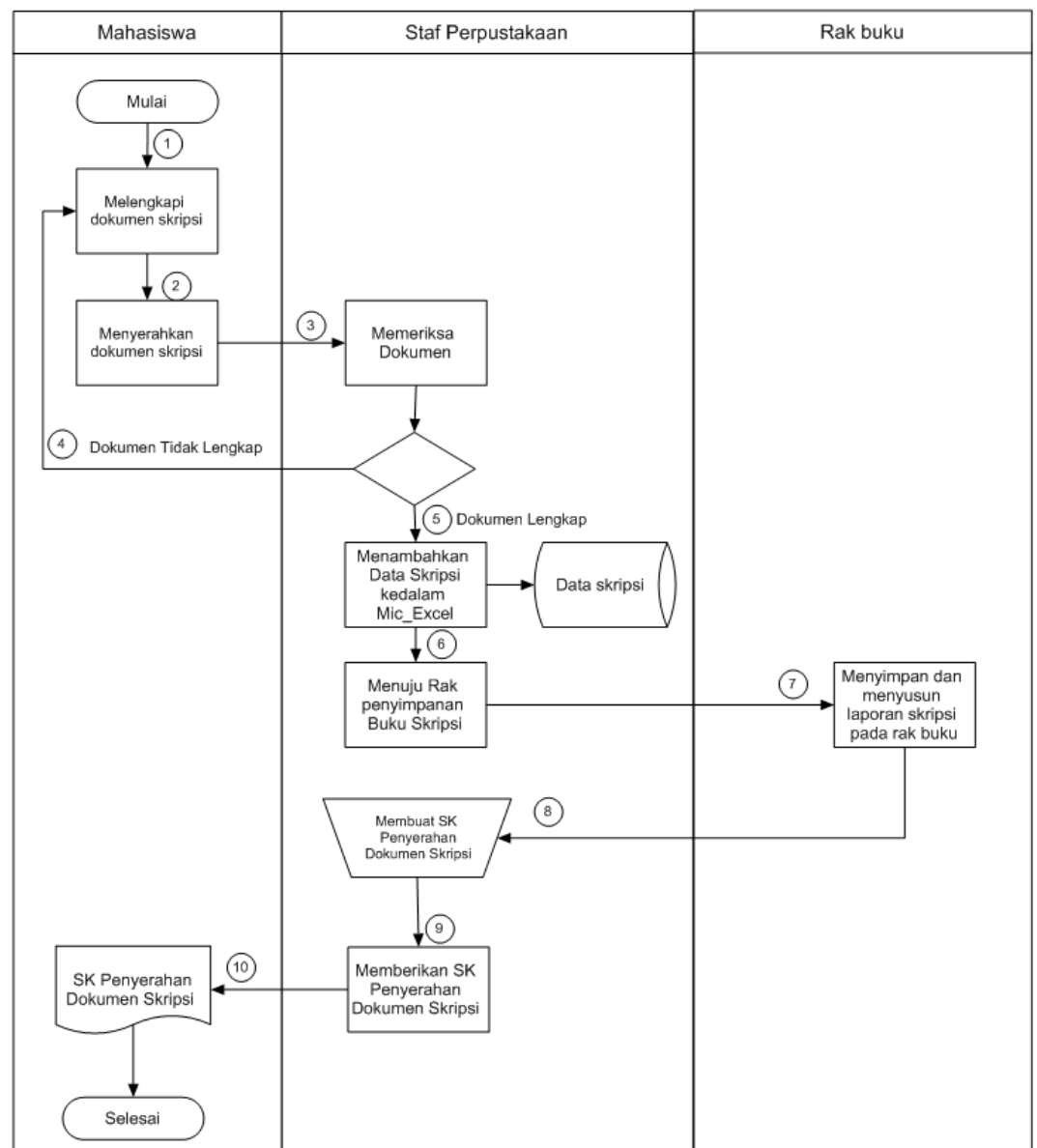
Selain melakukan wawancara kepada staf perpustakaan, penulis juga menanyakan masalah-masalah yang sering muncul dalam proses pengarsipan dokumen kepada dekan Fakultas Teknologi Informasi. Menurut penjelasannya sulitnya temu kenali dokumen baik berkas fisik maupun digital karena penyimpanan yang belum terindeks dan tidak tersedianya userinterface sebagai media komunikasi berkas digital.

4.1.2 Analisis Sistem

Pada tahap ini penulis membandingkan sistem yang sedang berjalan dan sistem yang akan di bangun dalam penelitian ini. Pada analisis sistem ditunjukkan dalam flowmap sebagai berikut:

1. Analisis Sistem Yang Sedang Berjalan

Berikut analisis Sistem Yang Sedang Berjalan Pada Pengelolaan dokumen skripsi di Fakultas Teknologi Informasi.

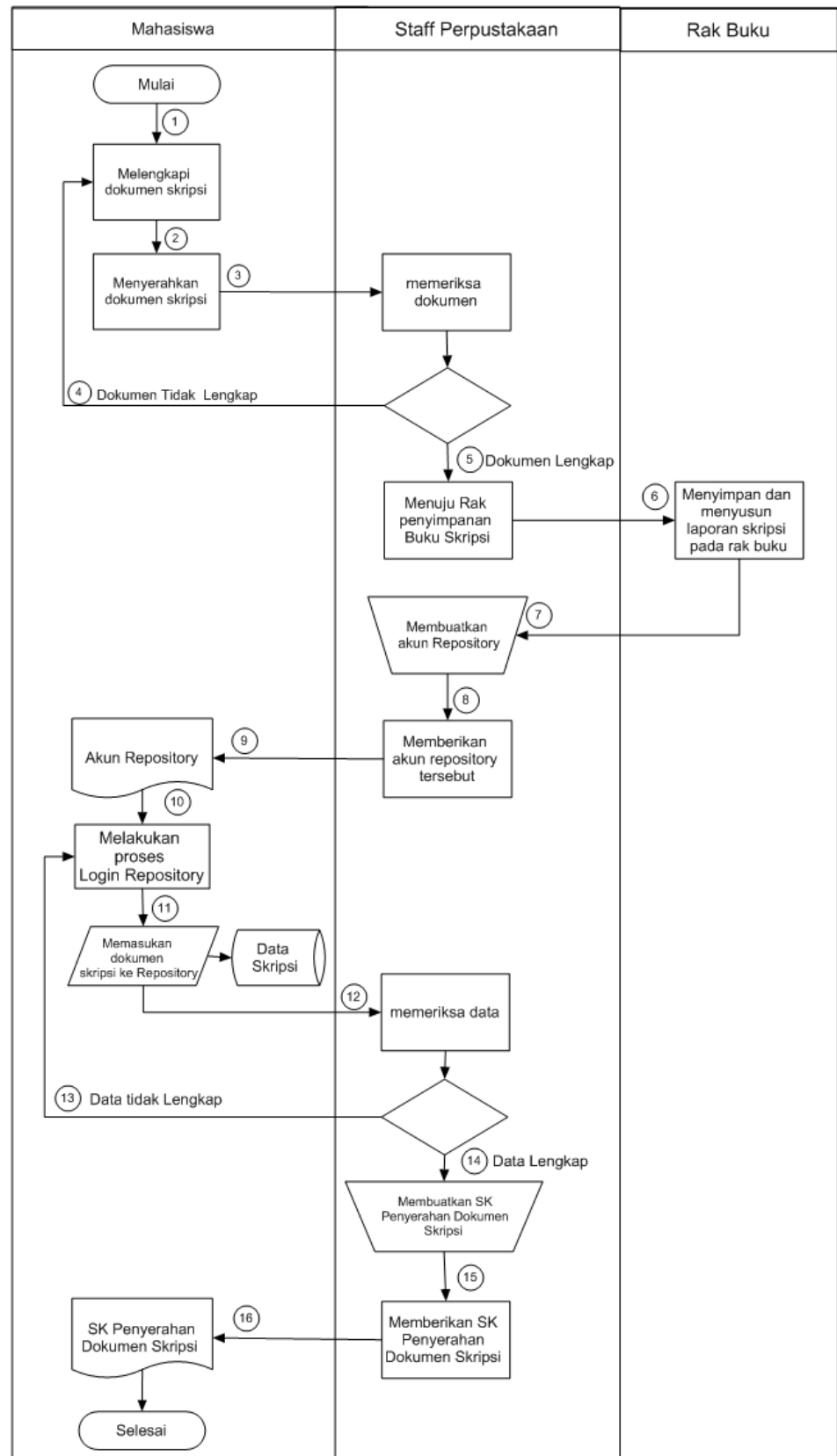


Gambar 4.1 Analisis Sistem Yang Sedang Berjalan

Berikut deskripsi dari flowmap sistem yang sedang berjalan:

- 1) Mahasiswa melengkapi dokumen skripsi sebelum dokumen tersebut diserahkan kepada staff perpustakaan .
 - 2) Setelah dokumen lengkap selanjutnya mahasiswa tersebut menyerahkan dokumen skripsi yang sudah dilengkapi kepada staff Perpustakaan.
 - 3) Setelah staff perpustakaan menerima dokumen skripsi, staff perpustakaan memeriksa dokumen tersebut memastikan kelengkapan dokumen.
 - 4) Jika dokumen belum lengkap maka staff perpustakaan akan mengembalikan kembali dokumen kepada mahasiswa agar mahasiswa melengkapi dulu kekurangan dokumennya.
 - 5) Jika dokumen sudah lengkap, staff perpustakaan akan memasukan data-data dokumen tersebut kedalam aplikasi mic-excel.
 - 6) Selanjutnya staff perpustakaan menuju rak penyimpanan buku skripsi sambil membawa dokumen tersebut.
 - 7) Setelah staff perpustakaan sampai di rak buku, staff perpustakaan menyimpan dan menyusun laporan skripsi tersebut kedalam rak buku..
 - 8) Selanjutnya staff perpustakaan akan membuatkan Surat Keterangan penyerahan dokumen skripsi untuk mahasiswa bersangkutan.
 - 9) staff perpustakaan memberikan Surat Keterangan penyerahan dokumen skripsi kepada mahasiswa,
 - 10) mahasiswa menerima Surat Keterangan penyerahan dokumen skripsi.
2. Analisis Sistem yang akan dibangun

Berikut analisis Sistem yang akan dibangun Pada Pengelolaan dokumen skripsi di Fakultas Teknologi Informasi.



Gambar 4.2 Analisis Sistem yang akan dibangun

Berikut deskripsi dari flowmap sistem yang akan dibangun:

- 1) Mahasiswa melengkapi dokumen skripsi.
- 2) mahasiswa menyerahkan dokumen skripsi yang sudah dilengkapi tersebut kepada staff Perpustakaan.
- 3) Setelah Staff perpustakaan Menerima dokumen skripsi, Staff perpustakaan akan memeriksa dokumen tersebut.
- 4) jika tanda dokumen belum lengkap maka staff perpustakaan akan mengembalikan kembali dokumen kepada mahasiswa.
- 5) jika dokumen sudah lengkap, staff perpustakaan akan membawa dokumen tersebut ke rak buku.
- 6) Sesampainya di rak buku Staff perpustakaan akan menyimpan dan menyusun laporan skripsi tersebut.
- 7) Selanjutnya staff perpustakaan membuat akun repository untuk mahasiswa.
- 8) Staff perpustakaan memberikan akun repository tersebut kepada mahasiswa.
- 9) Mahasiswa menerima akun repository tersebut.
- 10) Setelah Mahasiswa menerima akun repository mahasiswa melakukan proses login pada aplikasi.
- 11) Mahasiswa selanjutnya memasukan dokumen skripsinya dalam bentuk file pdf ke repository dan melengkapi indeks pada dokumen tersebut.
- 12) Selanjutnya staff perpustakaan akan memeriksa kembali data tersebut.
- 13) Jika data skripsi tidak lengkap, staff perpustakaan akan meminta mahasiswa memasukan kembali melakukan proses input.
- 14) Jika data skripsi yang di upload telah lengkap maka staff perpustakaan akan membuatkan SK Penyerahan dokumen skripsi untuk mahasiswa.
- 15) Setelah SK Penyerahan dokumen skripsi di buat, staff perpustakaan akan memberikannya kepada mahasiswa yang bersangkutan.

16) Mahasiswa mendapatkan SK Penyerahan dokumen skripsi.

4.1.3 Analisis Kebutuhan

Penulis melakukan analisis kebutuhan terhadap sistem yang akan dibangun berdasarkan kebutuhan yang dapat memenuhi proses pembuatan aplikasi yang meliputi kebutuhan software, kebutuhan hardware dan kebutuhan pengguna.

a. Kebutuhan Software Utama

- 1) *Elasticsearch*, digunakan sebagai *engine* yang dapat melakukan proses untuk mengindex, mencari dokumen. *Elasticsearch* juga merupakan server untuk menyimpan dokumen.
- 2) *Kibana*, digunakan untuk melakukan ujicoba API dalam mencari, melihat, dan berinteraksi dengan dokumen yang disimpan dalam *Elasticsearch*.
- 3) *Postman*, memiliki fungsionalitas yang sama dengan kibana yaitu digunakan untuk untuk melakukan ujicoba API dalam mencari, melihat, dan berinteraksi dengan dokumen yang disimpan dalam *Elasticsearch*. Dan *postman* ini berfungsi untuk mengecek apakah *elasticsearch* sudah berjalan atau belum
- 4) *Database MySQL*, digunakan untuk menyimpan data user untuk melakukan akses login yang dapat membantu dalam mengakses aplikasi.

b. Kebutuhan Software Pendukung

- 1) *Java* adalah bahasa pemrograman yang digunakan dalam pembuatan aplikasi utama. Menggunakan java pada NetBeans IDE karena library-nya lebih lengkap.
- 2) *Spring*, merupakan *framework* yang digunakan dalam pembuatan aplikasi di java.
- 3) *Apache Maven*, library yang digunakan dalam mengkonfigurasi beberapa modul aplikasi class yang ada di java antara client dan server, maven disimpan di pom.xml.
- 4) NetBeans IDE, digunakan untuk membuat dan mengembangkan sebuah aplikasi pencarian ini dengan dengan bahasa java.

- 5) Pemodelan UML menggunakan ClickChart
 - 6) Pemodelan mockup *user interface* menggunakan Balsamic Mhockup.
 - 7) Pemodelan *user interface* pengguna *Scene builder*
- c. Kebutuhan Hardware
- Proses dalam melakukan analisis kebutuhan langkah berikutnya adalah melakukan installasi *elasticsearch* sebagai server, *postman* dan kibana yang berperan membantu manajemen dan mengembangkan aplikasi *elasticsearch* di laptop yang memiliki spesifikasi minimal sebagai berikut:
- 1) AMD Dual Core E1-6010 1.35 GHz.
 - 2) RAM 4GB / HDD 500 GB

Dalam penelitian ini menggunakan spesifikasi minimal. Oleh karenanya dalam melakukan penelitian ini penulis membatasi jumlah data yang tidak terlampau besar. Dibuktikan dengan sekian halaman dapat melakukan penyimpanan, memastikan bahwa berupa pdf bisa tersimpan.

d. Kebutuhan Pengguna

Dalam aplikasi yang dibuat pengguna dibagi menjadi 2 kewajiban akses, yaitu meliputi mahasiswa sebagai user yang akan melakukan penyimpanan dokumen karya ilmiahnya ke repository dan staff Tata Usaha sebagai yang memiliki hak akses untuk melakukan penghapusan dokumen, pengeditan data dokumen, pembuatan akun baru untuk user dekaligus mereset akun user

4.1.4 Hasil Analisis

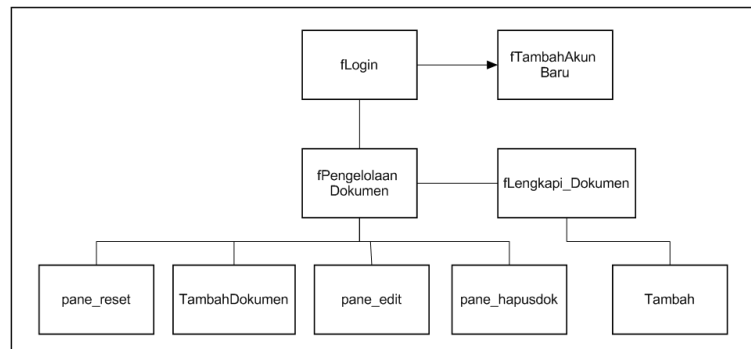
Hasil analisis dilakukan dengan membuat tabel dari beberapa kemungkinan sebuah dokumen dapat di inputkan dengan cara membandingkannya dengan dokumen yang sudah ada. Berikut ini adalah tabel kemungkinan yang telah dibuat:

Tabel 4.1 Tabel Hasil Analisis

ID Dokumen	Autor	Judul Karya Ilmiah	Tahun Terbit	Keterangan
Sama	Sama	Sama	Sama	TD
Sama	Sama	Sama	Beda	TD
Sama	Sama	Beda	Beda	TD
Sama	Beda	Beda	Beda	TD
Beda	Beda	Beda	Sama	DS
Beda	Sama	Beda	Sama	DS
Beda	Sama	Sama	Sama	DS
Beda	Beda	Beda	Beda	DS

Keterangan : TD = Tidak disimpan; DS = Dapat disimpan

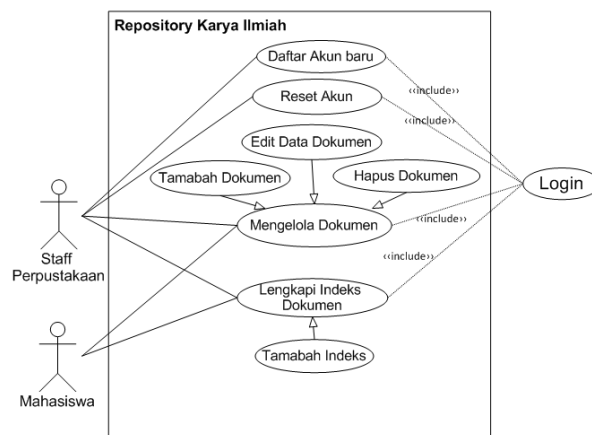
4.2 Perancangan



Gambar 4.3 Struktur Menu

Dari gambar struktur menu diatas menunjukan bahwa admin dapat melakukan proses tambah akun baru di form login dan melakukan reset akun, tambah dokumen, edit data dokumen dan hapus dokumen pada form pengelolaan dokumen, serta melengkapi indeks pada dokumen di form lengkapi dokumen.

4.2.1 Use Case Diagram

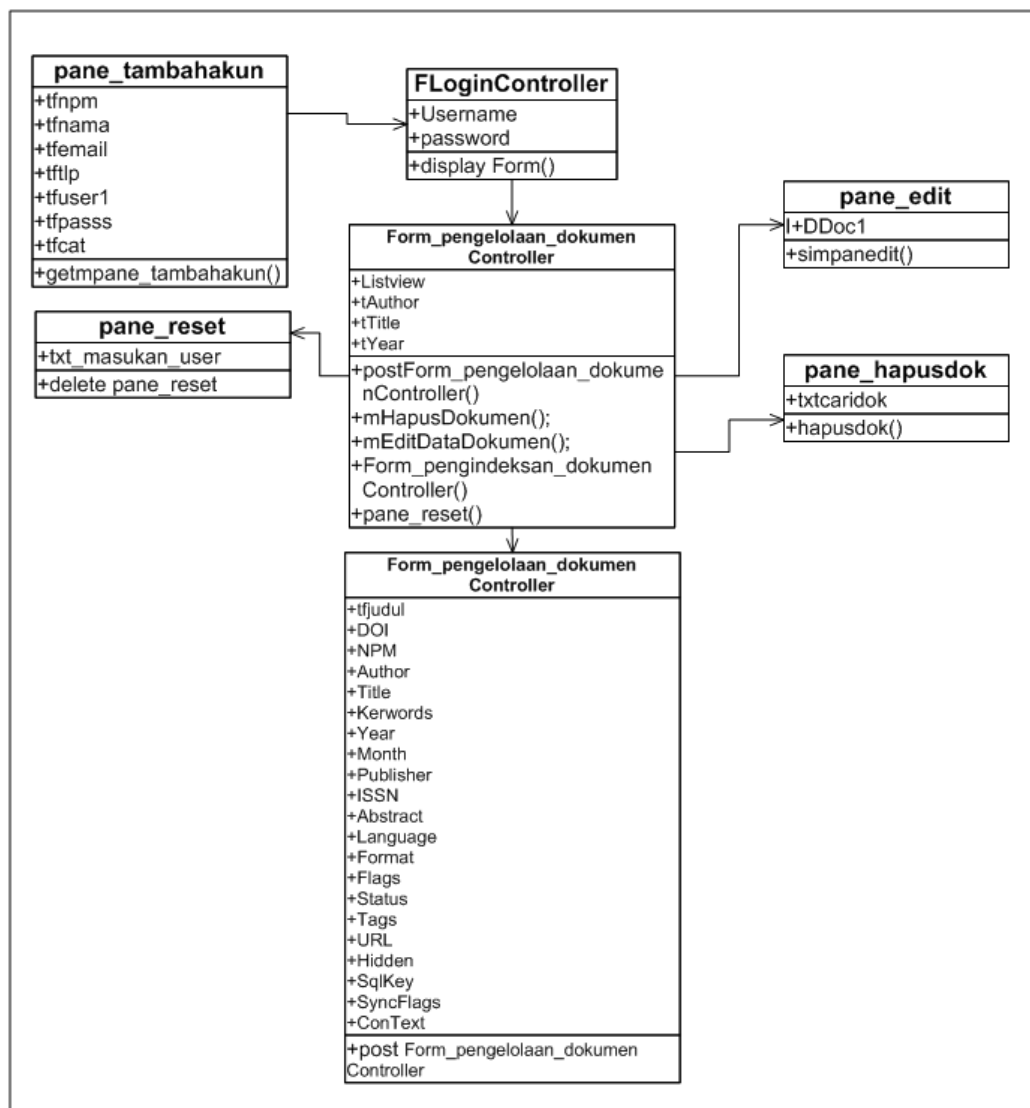


Gambar 4.4 Use Case Diagram

Dari gambar usecase di atas menunjukkan bahwa hak akses staff perpustakaan lebih dominan daripada mahasiswa yaitu staff perpustakaan dapat melakukan pengelolaan dokumen seperti, melakukan penambahan data, pengeditan data , penghapusan data, menambahkan akun baru, dan melakukan reset akun, yang dimana mahasiswa tidak memiliki hak akses tersebut. Mahasiswa hanya dapat melakukan tambah dokumen dan melengkapi indeks dokumen saja.

4.2.2 Class Diagram

Berikut adalah class diagram dari perancangan aplikasi.



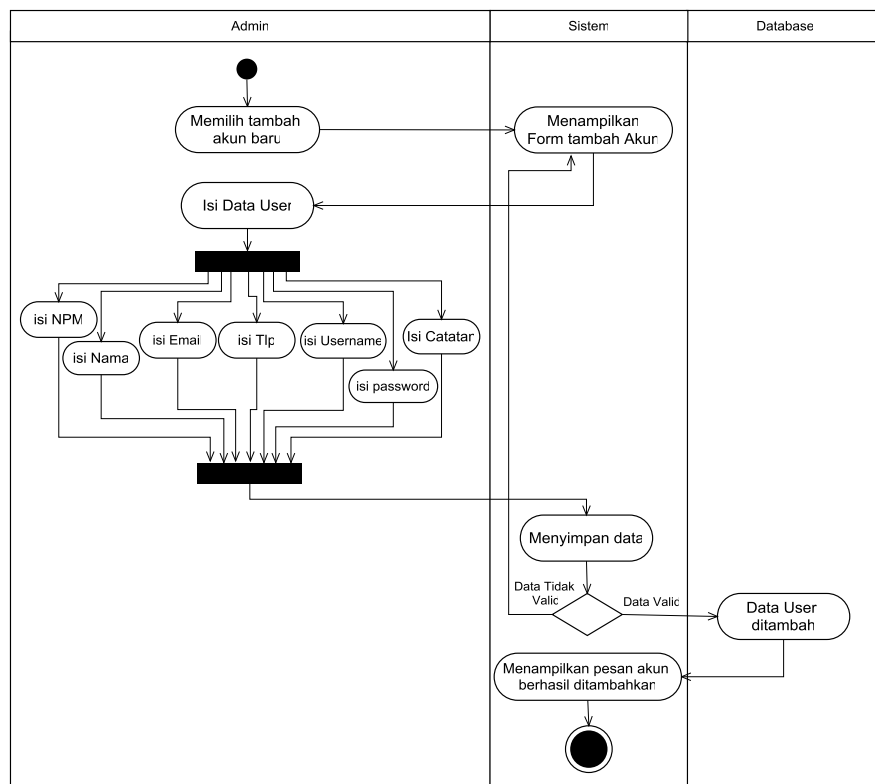
Gambar 4.5 Class diagram

Gambar diatas merupakan penggambaran dari class, atribut, dan objek, yang menggambarkan hubungan antara satu sama lainnya.

4.2.3 Activity Diagram

Berikut beberapa *Activity Diagram* yang sedang di rancang yang diantaranya *Activity Diagram* Tambah Akun, *Activity Diagram* untuk login *Activity Diagram* Reset Akun, *Activity Diagram* tambah dokumen, *Activity Diagram* penghapusan dokumen, *Activity Diagram* Edit dokumen, *Activity Diagram* pencarian judul karya ilmiah, *Activity Diagram* untuk Lengkapi indeks dokumen.

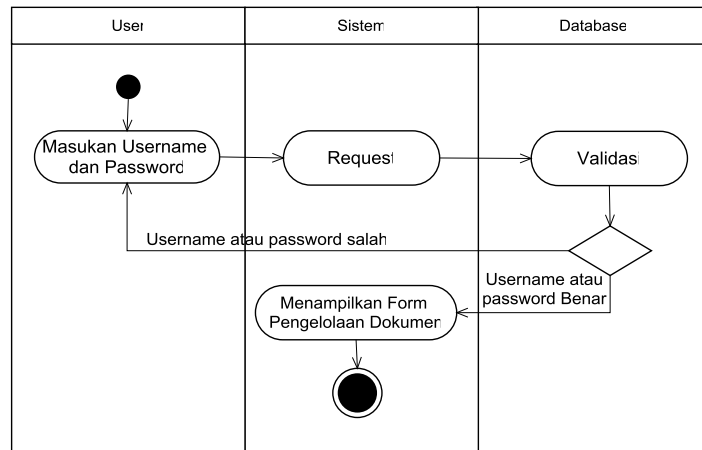
1. Activity Diagram Tambah akun baru



Gambar 4.6 Activity Diagram Tambah akun baru

Ketika user berada di form login user memilih tombol tambah akun baru yang ada pada form login tersebut, kemudian aplikasi akan menampilkan form tambah akun. selanjutnya user mengisi data data user pada form tambah user, jika data user valid maka akan ada pesan akun berhasil ditambahkan, jika tidak valid maka akan muncul pesan data tidak valid.

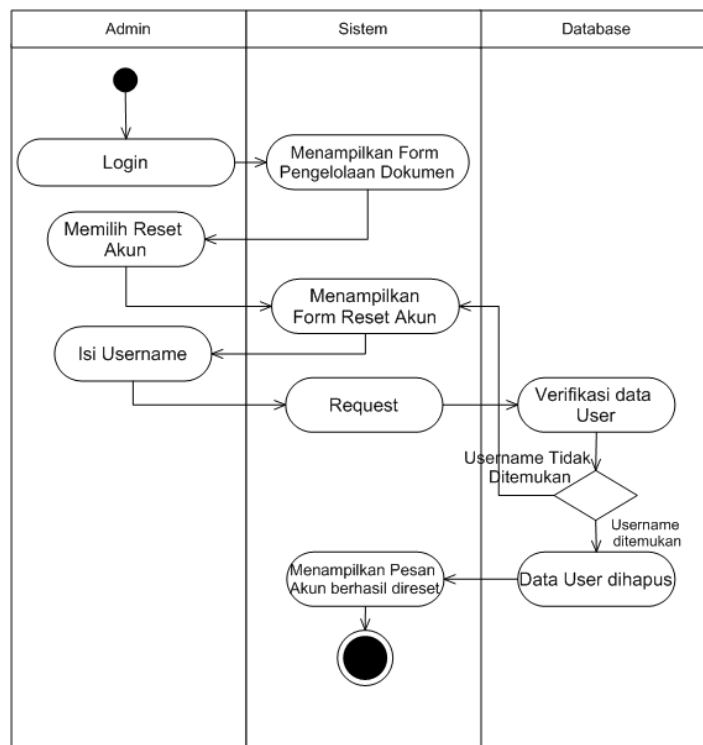
2. Activity Diagram Login



Gambar 4.7 Activity Diagram Login

Setelah user memiliki akun Repository, user wajib memasukan username serta password yang kemudian apabila username atau password valid atau sesuai maka aplikasi akan menampilkan form menu utama yaitu form pengelolaan dokumen tetapi apabila username atau password tidak valid maka akan memunculkan pesan username atau password salah.

3. Activity Diagram Reset Akun

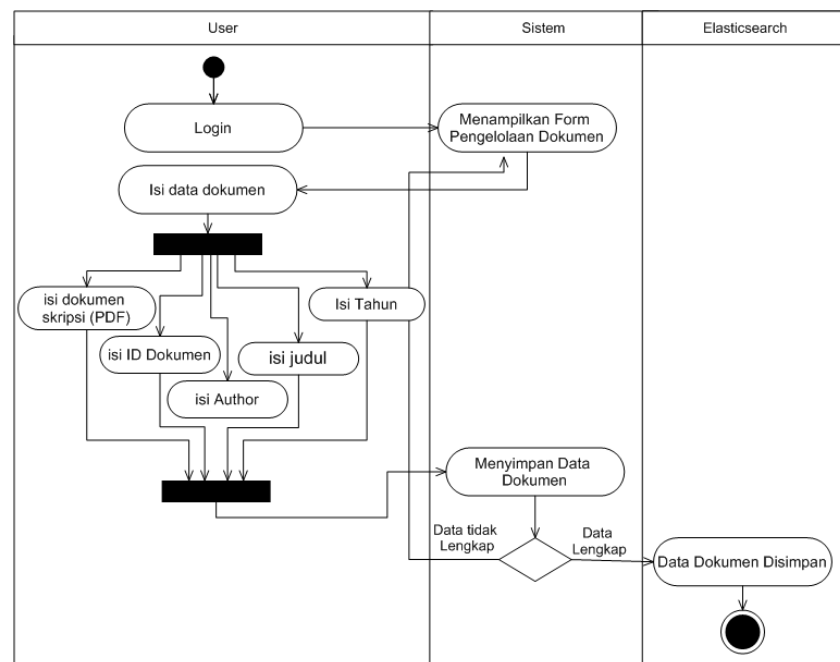


Gambar 4.8 Activity Diagram Reset Akun

Gambar diatas enunjukkan Tombol reset akun ini berada di form menu utama yaitu form pengelolaan dokumen, ketika user berada di form pengelolaan dokumen, user memilih tombol reset akun, kemudian akan muncul form reset akun. Selanjutnya user memasukkan username dari akun yang aka dihapus, jika username tersedia maka aplikasi akan menampilkan pesan Akun berhasil di reset dan akun direset. Jika username tidak ditemukan maka aplikasi akan memenampikan pesan username tidak ditemukan.

4. Activity Diagram tambah dokumen

Berikut adalah activity Diagram tambah dokumen

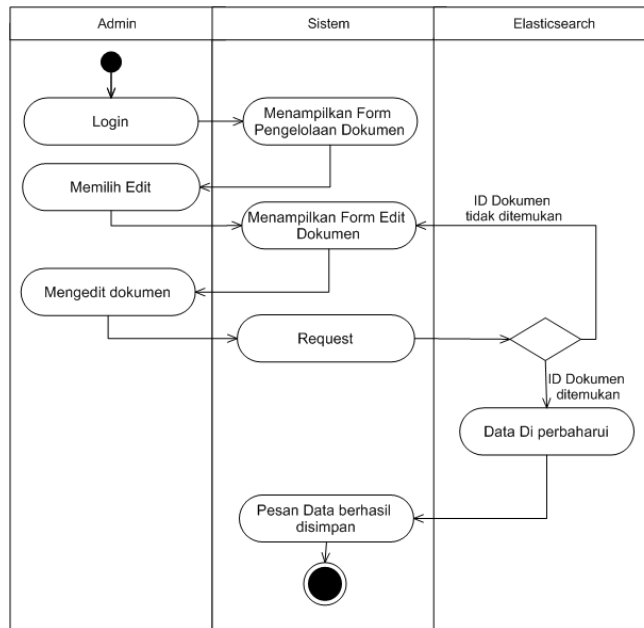


Gambar 4.9 Activity Diagram tambah dokumen

User mengisi data dokumen pada form pengelolaan dokumen tersebut dengan memasukkan file dokumen, beserta ID Dokumen, author, judul karya ilmiah, dan tahun terbit. Jika data sudah lengkap maka aplikasi akan menyimpan data dokumen tersebut kedalam database dan jika data tidak lengkap aplikasi akan menampilkan pesan data tidak lengkap.

5. Activity Diagram Edit data Dokumen.

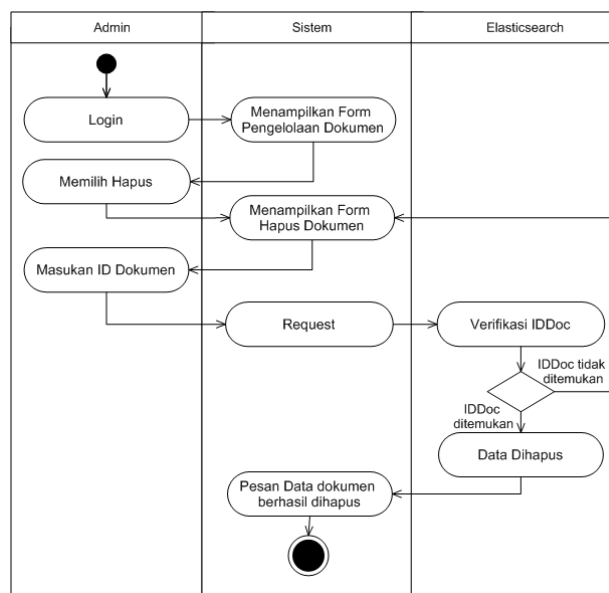
Berikut adalah Activity Diagram Edit data Dokumen:



Gambar 4.10 Activity Diagram Edit data dokumen

Ketika admin menekan tombol edit di form pengelolaan dokumen, sistem akan menampilkan form edit dan kemudian admin bisa memasukan ID Dokumen yang akan di edit datanya jika id dokumen tersedia admin langsung mengedit dokumen nya kemudian tekan tombol simpan untuk menyimpan perubahan data.

6. Activity Diagram hapus dokumen.

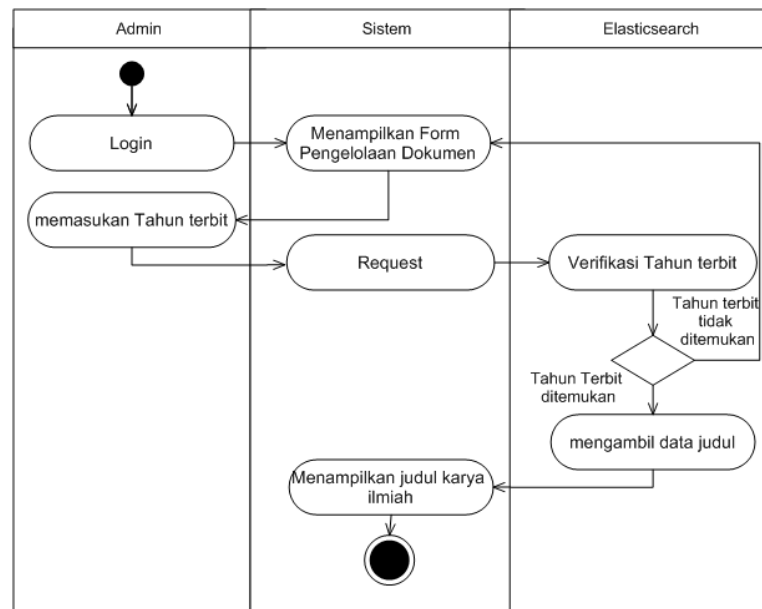


Gambar 4.11 Activity Diagram Hapus Dokumen

pada saat aplikasi menampilkan form pengelolaan dokumen, user memilih tombol hapus, selanjutnya aplikasi akan menampilkan form hapus dokumen. Ketika berada di form hapus dokumen user memasukkan ID Dokumen yang akan dihapus. Jika ID Dokumen yang akan dihapus ditemukan maka aplikasi akan menghapus dokumen berdasarkan ID Dokumen tersebut. Tetapi jika ID Dokumen yang akan dihapus tidak ditemukan aplikasi akan menampilkan pesan ID Dokumen yang anda cari tidak cocok.

7. Activity diagram Pencarian judul karya ilmiah.

Berikut adalah Activity diagram pencarian judul karya ilmiah.

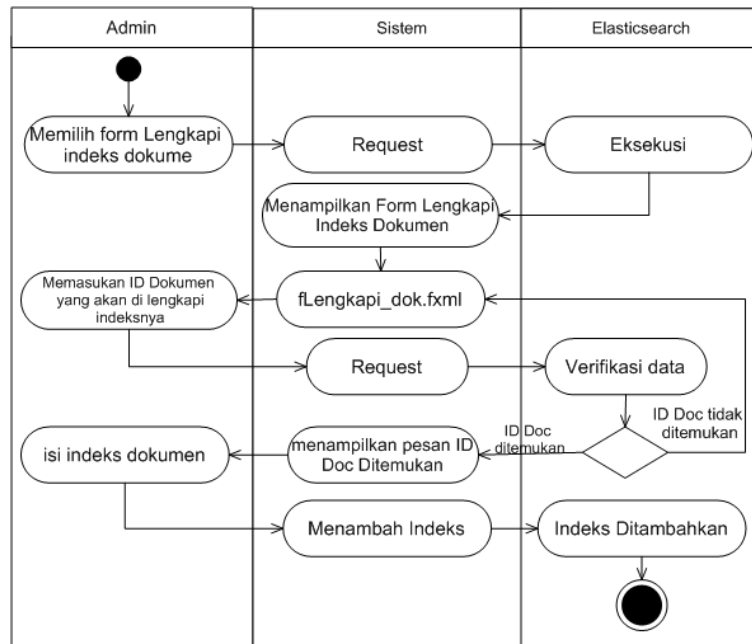


Gambar 4.12 Activity diagram cari judul karya ilmiah

Pada form pengelolaan dokumen, user memasukkan tahun terbit di kolom pencarian judul karya ilmiah, jika judul karya ilmiah tidak ditemukan pada tahun terbit yang di masukan aplikasi akan menampilkan pesan, data yang anda masukan tidak cocok, tetapi jika judul karya ilmiah di temukan maka aplikasi akan menampilkan semua judul karya ilmiah berdasarkan tahun terbit yang telah di masukan.

8. Activity Diagram lengkapi indeks dokumen

Berikut adalah activity diagram lengkapi indeks dokumen :



Gambar 4.13 Activity diagram lengkapi indeks dokumen

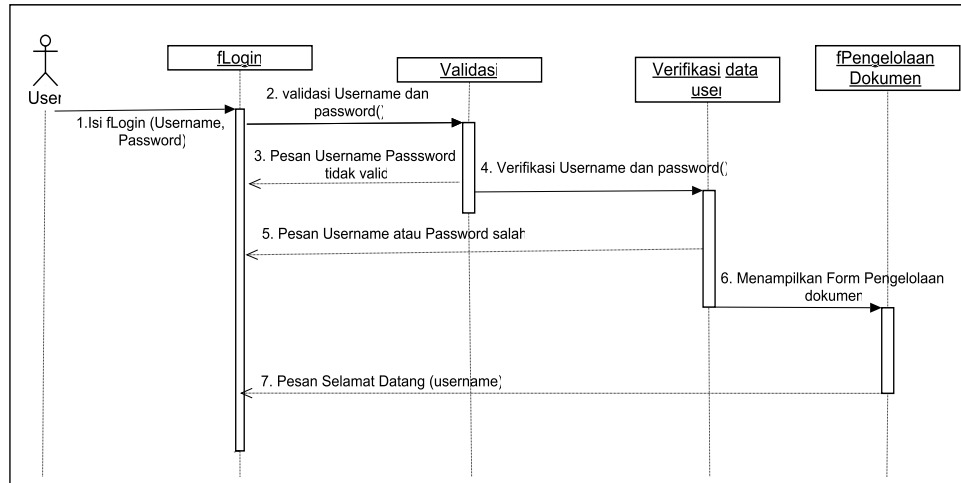
Gambar diatas menunjukan pada saat user berada di form menu utama, user pilih tombol lengkapi indeks dokumen. Kemudian aplikasi menampilkan form lengkapi indeks dokumen, selanjutnya user memasukan ID Dokumen yang akan dilengkapi indeks nya. jika ID Dokumen tidak ditemukan aplikasi akan memunculkan pesan ID Dokumen yang adnda masuka tidak cocok, jika ID Dokumen ditemukan maka aplikasi akan menampilkan pesan ID Dokumen ditemukan. Selanjutnya user mengisi indeks yang belum lengkap pada dokumen dan aplikasi menambahkan indeks tersebut.

4.2.4 Sequence Diagram

dalam perancangan menggunakan 8 squence diagram diantaranya adalah squence diagram Login, squence diagram tambah akun baru, squence diagram Reset Akun, squence diagram Tambah Dokumen, squence diagram Hapus Dokumen, squence diagram Edit data dokumen, squence diagram pencarian judul karya ilmiah dan squensdiagram lengkapi dokumen.

1. Sequence Diagram Login

Berikut ini adalah tampilan dari sequence diagram Login :

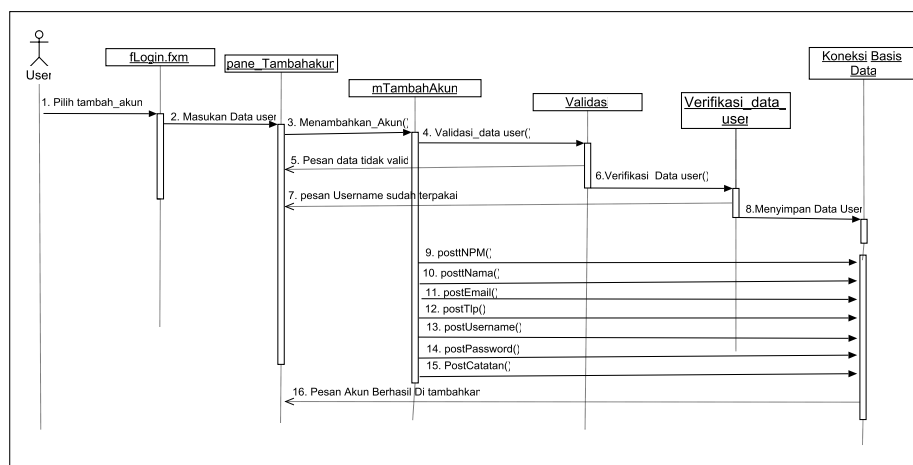


Gambar 4.14 Sequence Diagram Login

Berikut deskripsi Sequence Login

- 1) User memasukkan username dan password pada form Login.
- 2) Aplikasi melakukan validasi Username dan password.
- 3) Jika username atau password tidak valid maka akan menampilkan pesan username dan password tidak valid.
- 4) Jika username dan password valid aplikasi melakukan verifikasi data.
- 5) Jika username dan password salah maka akan menampilkan pesan username atau password salah.
- 6) username dan password sesuai dengan data user maka aplikasi akan menampilkan form pengelolaan dokumen.
- 7) Sistem menampilkan pesan selamat datang (username).

2. Sequence Diagram Tambah akun baru.

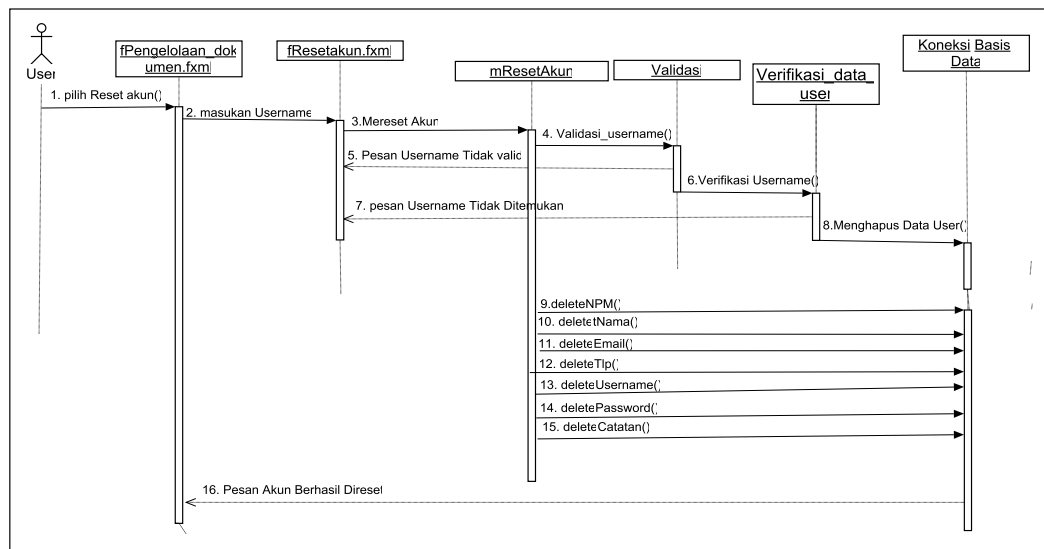


Gambar 4.15 Sequence Diagram Tambah akun baru

Deskripsi Sequence Diagram Tambah akun baru

- 1) User memilih tombol tambah akun baru pada form login.
 - 2) Kemudian memasukan data user pada form tambah akun baru.
 - 3) Kemudian pilih tambah.
 - 4) Aplikasi melakukan validasi data user
 - 5) Jika data tidak valid maka akan menampilkan pesan data tidak valid.
 - 6) Jika data user valid maka aplikasi akan melakukan verifikasi data user.
 - 7) Jika username sudah pernah terpakai maka aplikasi menampilkan pesan username sudah terpakai.
 - 8) Jika data user sudah sesuai maka aplikasi akan menyimpan data user dan menampilkan pesan akun berhasil ditambahkan.
3. Sequence Diagram Reset akun.

Berikut adalah Sequence diagram Reset Akun



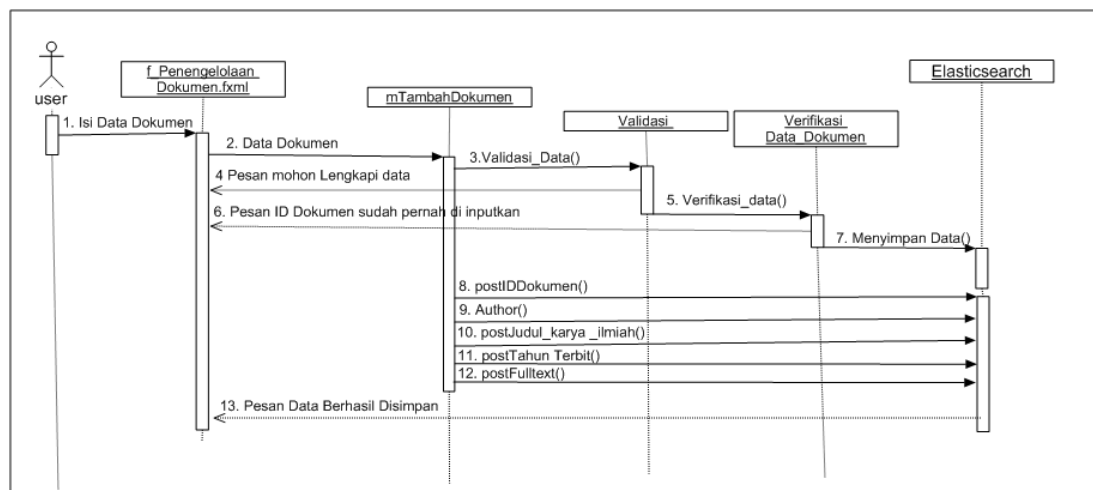
Gambar 4.16 Sequence Diagram Reset akun

Berikut deskripsi Sequence diagram Reset Akun:

- 1) Pilih reset akun pada form pengelolaan dokumen.
- 2) Masukan username akun yang akan direset.
- 3) Melakukan proses reset akun.
- 4) Validasi username.

- 5) Jika username tidak valid maka akan menampilkan pesan Username tidak valid.
 - 6) Jika username valid maka akan melakukan proses verifikasi data user.
 - 7) Jika username tidak ditemukan maka akan memunculkan pesan username tidak ditemukan.
 - 8) Jika username ditemukan maka aplikasi akan melakukan penghapusan data user dan selanjutnya menampilkan pesan akun berhasil direset.
4. Sequence Diagram Tambah Dokumen

Berikut adalah tampilan Sequence diagram Tambah Dokumen:



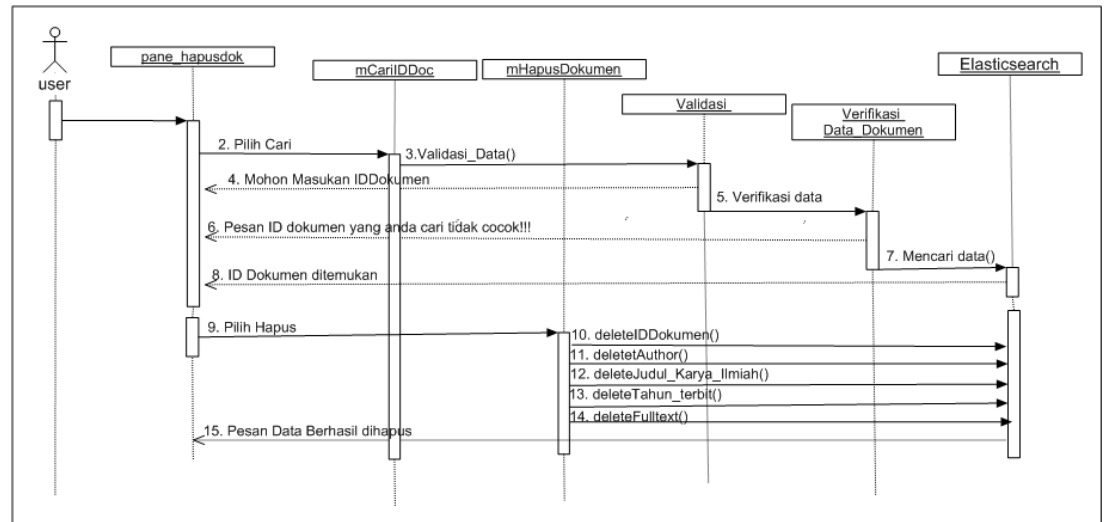
Gambar 4.17 Sequence Diagram Tambah Dokumen

Deskripsi Sequence Diagram 4.16 Tambah Dokumen

- 1) Isi data dokumen.
- 2) Menambahkan data dokumen.
- 3) Aplikasi melakukan validasi data dokumen.
- 4) Jika data belum terisi lengkap maka aplikasi menampilkan pesan mohon lengkapi data.
- 5) Jika data sudah lengkap maka aplikasi melakukan proses verifikasi.
- 6) Jika ID dokumen mirip dengan ID dokumen sebelumnya maka aplikasi menampilkan pesan ID dokumen sudah pernah diinputkan.
- 7) ID dokumen belum pernah diinputkan sebelumnya maka aplikasi akan melakukan penyimpanan data dokumen.

- 8) Aplikasi menyimpan ID Dokumen, Author, Judul karya ilmiah, Tahun terbit, serta full text pada file pdf yang telah di inputkan.

5. Sequence Diagram Hapus Dokumen

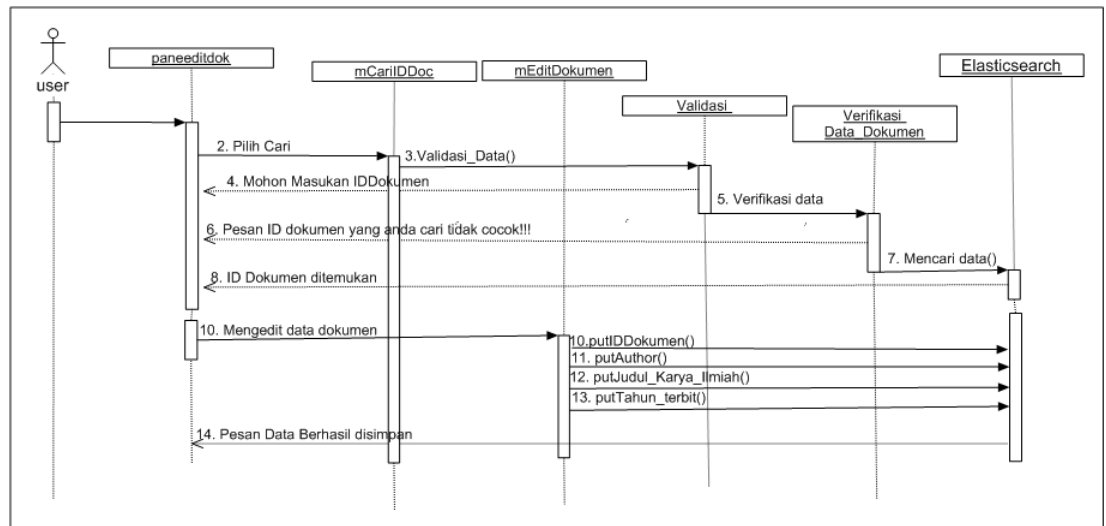


Gambar 4.18 Sequence Diagram Hapus Dokumen

Deskripsi Sequence Diagram Hapus Dokumen :

- 1) Masukan ID dokumen pada form hapus dokumen.
 - 2) Pilih tombol cari.
 - 3) Aplikasi melakukan validasi ID dokumen.
 - 4) Jika ID dokumen tidak valid maka aplikasi akan menampilkan pesan mohon masukan ID dokumen.
 - 5) Jika ID dokumen valid maka aplikasi akan melakukan proses verifikasi judul karya ilmiah.
 - 6) Jika ID dokumen tidak di temukan aplikasi akan menampilkan pesan ID dokumen yang anda cari tidak cocok!!!.
 - 7) Jika ID dokumen ditemukan maka aplikasi akan menampilkan pesan ID dokumen ditemukan.
 - 8) Kemudian user pilih tombol hapus.
 - 9) aplikasi menghapus data dokumen.
- ## 6. Sequence Diagram Edit Data Dokumen

Berikut adalah Sequence Diagram Edit Data Dokumen :

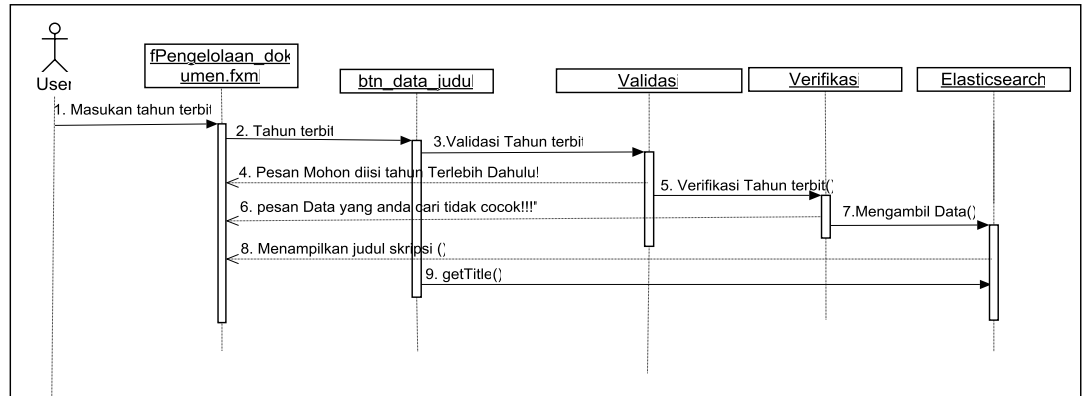


Gambar 4.19 Sequence Diagram edit data dokumen

Deskripsi Sequence Diagram Lengkapi Indeks Dokumen:

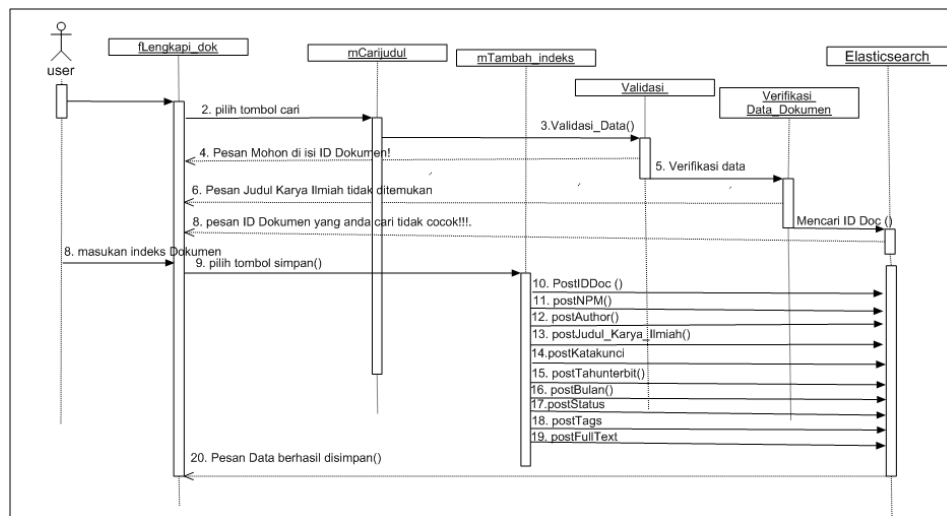
- 1) Masukan ID dokumen pada form Edit data dokumen.
 - 2) Pilih tombol cari.
 - 3) Aplikasi melakukan validasi ID dokumen.
 - 4) Jika ID dokumen tidak valid maka aplikasi akan menampilkan pesan mohon masukan ID dokumen.
 - 5) Jika ID dokumen valid maka aplikasi akan melakukan proses verifikasi ID Dokumen.
 - 6) Jika ID dokumen tidak di temukan aplikasi akan menampilkan pesan ID dokumen yang anda cari tidak cocok!!!.
 - 7) Jika ID dokumen ditemukan maka aplikasi akan menampilkan pesan ID dokumen ditemukan.
 - 8) Kemudian user mengedit data dokumen dan klik tombol Simpan.
 - 9) aplikasi akan menyimpan ID Dokumen, Author, Judul karya ilmiah dan tahun terbit.
 - 10) Aplikasi menampilkan pesan “Data berhasil disimpan”.
7. Sequence Pencarian judul skripsi.

Berikut adalah tampilan dari Sequence Pencarian judul skripsi berdasarkan tahun terbit :



Gambar 4.20 Sequence Pencarian judul skripsi

- 1) Masukan tahun terbit pada kolom tahun terbit di form pengelolaan dokumen.
 - 2) Tekan tombol cari.
 - 3) Proses validasi tahun terbit
 - 4) Jika kolom tahun terbit kosong maka akan tampil pesan “mohon diisi tahun terbit terlebih dahulu!”.
 - 5) Proses verifikasi tahun terbit.
 - 6) Jika tahun terbit tidak di temukan maka muncul pesan “ Data yang anda cari tidak cocok!!”.
 - 7) Jika tahun terbit di temukan maka sistem akan mengambil data judul skripsi di elasticsearch.
 - 8) Sistem Menampilkan judul skripsi berdasarkan tahun terbit.
8. Sequence Diagram Lengkapi Indeks Dokumen.



Gambar 4.21 Sequence Diagram Lengkapi Indeks Dokumen

Deskripsi Sequence Diagram Lengkapi Indeks Dokumen:

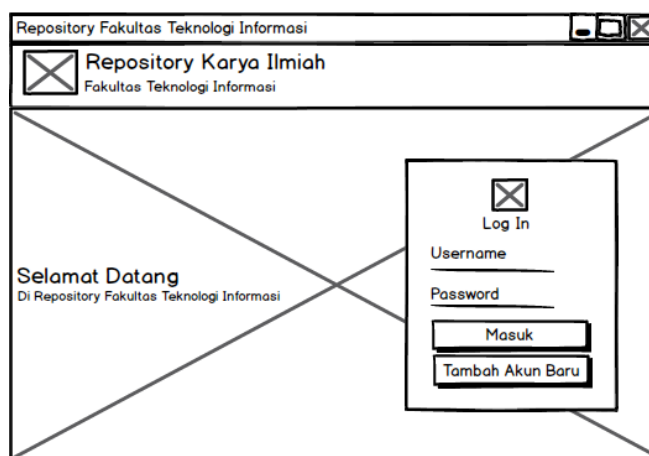
- 1) Masukan ID Dokumen pada form lengkapi dokumen.
- 2) Pilih tombol cari.
- 3) Aplikasi melakukan validasi data.
- 4) Jika data tidak valid aplikasi akan menampilkan pesan "Mohon di isi ID Dokumen!".
- 5) Jika data valid maka aplikasi akan melakukan proses verifikasi data.
- 6) Jika judul karya ilmiah tidak ditemukan maka aplikasi akan menampilkan pesan ID Dokumen yang anda cari tidak cocok!!!.
- 7) Jika ID Dokumen ditemukan maka aplikasi akan menampilkan pesan Data dokumen tersebut.
- 8) User memasukkan indeks dokumen yang belum lengkap.
- 9) User memilih tombol simpan dan aplikasi menyimpan indeks pada dokumen yang baru di lengkapi.

4.2.5 Perancangan User Interface

Di dalam melakukan perancangan User Interface ada beberapa Mockup yang penulis buat diantaranya Mockup form Login, Mockup form Tambah Akun, Mockup Edit data dokumen, Mockup Hapus dokumen, Mockup form Reset Akun, Mockup form menu utama admin, , Mockup form menu utama mahasiswa dan Mockup form Lengkapi indeks dokumen, Berikut mockup nya:

1. Mockup login

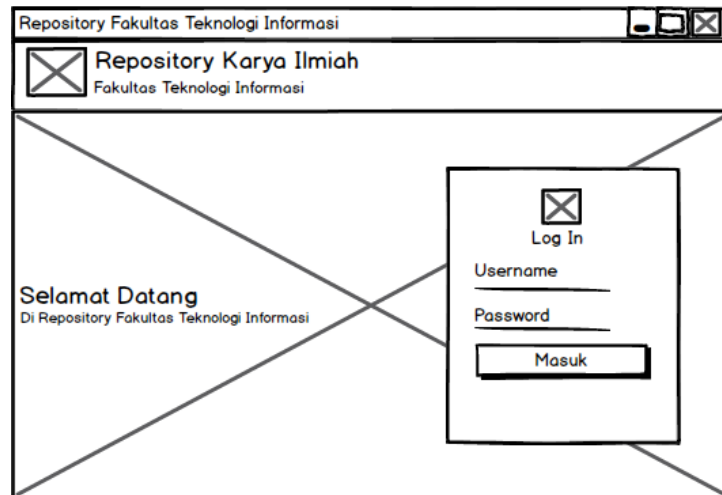
a. Tampilan Mockup Login Admin :



Gambar 4.22 Mockup Login

Gambar diatas merupakan perancangan antar muka atau user interface Login. Didalamnya terdapat fitur untuk melakukan Login dan terdapat fitur tombol tambah akun.

b. Tampilan Mockup Login Mahasiswa



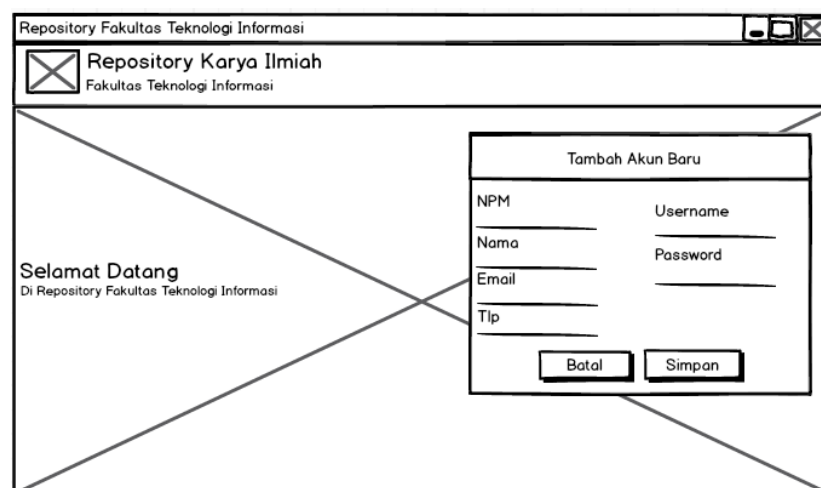
The mockup shows a web browser window titled "Repository Fakultas Teknologi Informasi". The header contains a logo and the text "Repository Karya Ilmiah" and "Fakultas Teknologi Informasi". The main content area is split: on the left, it says "Selamat Datang" and "Di Repository Fakultas Teknologi Informasi"; on the right, there is a "Log In" form. The form includes fields for "Username" and "Password", and a "Masuk" button.

Gambar 4.23 Mockup Login Mahasiswa

Gambar diatas merupakan perancangan antar muka atau user interface Login. Didalamnya terdapat fitur untuk melakukan Login. Berbeda dengan tampilan login admin, dimana tampilan login mahasiswa tidak dilengkapi fitur tambah akun baru.

2. Mockup Tambah akun baru

Berikut tampilan Mockup Tambah akun baru



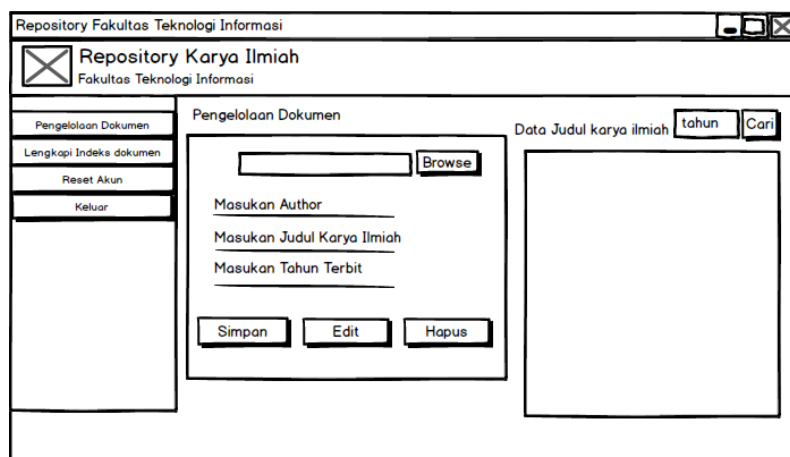
The mockup shows a web browser window titled "Repository Fakultas Teknologi Informasi". The header contains a logo and the text "Repository Karya Ilmiah" and "Fakultas Teknologi Informasi". The main content area is split: on the left, it says "Selamat Datang" and "Di Repository Fakultas Teknologi Informasi"; on the right, there is a "Tambah Akun Baru" form. The form includes fields for "NPM", "Nama", "Email", "Tlp", "Username", and "Password", and "Batal" and "Simpan" buttons.

Gambar 4.24 Mockup Tambah akun baru

Gambar diatas merupakan perancangan antar muka atau user interface Tambah akun baru. Didalamnya terdapat fitur untuk melakukan tambah akun baru . terdapat dua tombol yaitu tombol simpan untuk menyimpan akun dan tombol batal untuk membatalkan proses tambah akun.

3. Mockup Menu Utama Admin

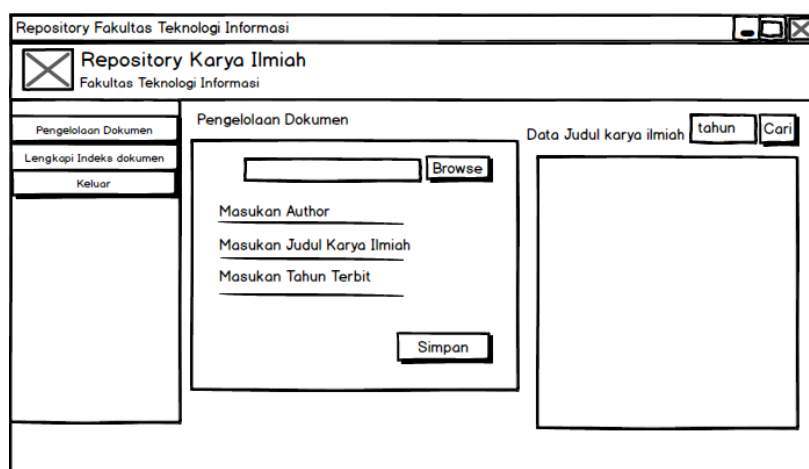
Berikut tampilan Menu utama admin:



Gambar 4.25 Mockup Menu utama admin.

Gambar diatas merupakan perancangan antarmuka atau userinterface tampilan menu utama admin, didalamnya terdapat fitur penyimpanan dokumen, Edit data dokumen, tombol hapus dokumen, tombol lengkapi indeks dokumen, tombol pencarian judul dokumen, tombol reset akun, serta tombol keluar.

4. Mockup Menu Utama Mahasiswa



Gambar 4.26 Mockup Menu utama mahasiswa.

Gambar diatas merupakan perancangan antarmuka tampilan menu utama mahasiswa, didalamnya hanya terdapat terdapat fitur penyimpanan dokumen, lengkapi indeks dokumen serta tombol keluar.

5. Mockup Hapus Dokumen

Gambar 4.27 Mockup Hapus dokumen

Gambar diatas merupakan tampilan dari mockup hapus dokumen, fitur hapus dokumen ini hanya dapat diakses oleh admin saja. Pada tampilan diatas terdapat, tombol cari untuk mencari id dokumen yang akan dihapus, tombol hapus untuk menghapus dokumen dan tombol batal untuk membatalkan proses hapus dokumen.

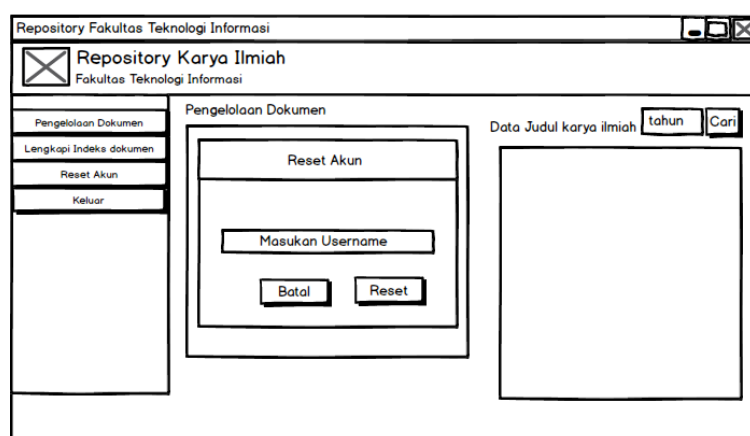
6. Mockup Edit data dokumen

Gambar 4.28 Mockup Edit data dokumen

Gambar diatas merupakan tampilan dari mockup edit data dokumen, fitur edit data dokumen ini hanya dapat diakses oleh admin saja. Pada tampilan diatas terdapat tombol simpan untuk menyimpan data yang telah diedit dokumen dan tombol batal untuk membatalkan proses edit data dokumen.

7. Mockup Reset akun

Berikut tampilan Mockup Reset akun :

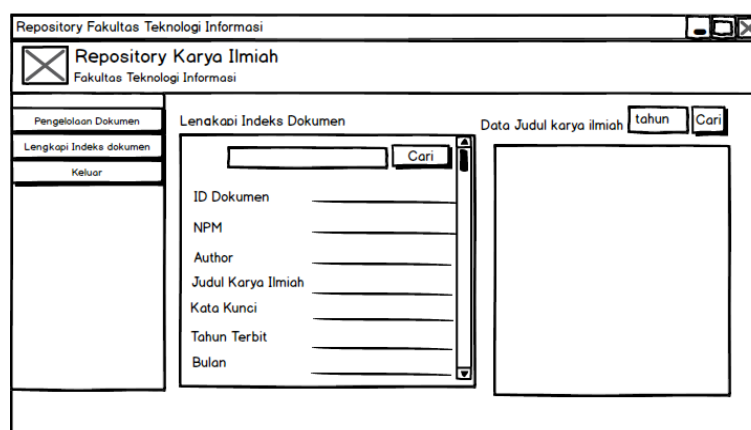


The mockup shows a web application window titled "Repository Fakultas Teknologi Informasi". The main header contains the logo and text "Repository Karya Ilmiah" and "Fakultas Teknologi Informasi". On the left, there is a sidebar menu with options: "Pengelolaan Dokumen", "Lengkapi Indeks dokumen", "Reset Akun", and "Keluar". The main content area is titled "Pengelolaan Dokumen" and features a "Reset Akun" form. This form includes a text input field labeled "Masukan Username" and two buttons: "Batal" and "Reset". To the right of the form, there is a section titled "Data Judul karya ilmiah" with a "tahun" dropdown menu and a "Cari" button.

Gambar 4.29 Mockup Reset akun

Gambar diatas merupakan perancangan antarmuka atau userinterface tampilan Reset akun fitur ini hanya dapat diakses oleh admin, didalamnya terdapat tombol reset untuk melanjutkan mereset akun dan tombol batal untuk membatalkan proses reset akun.

8. Mockup Lengkapi indeks dokumen



The mockup shows a web application window titled "Repository Fakultas Teknologi Informasi". The main header contains the logo and text "Repository Karya Ilmiah" and "Fakultas Teknologi Informasi". On the left, there is a sidebar menu with options: "Pengelolaan Dokumen", "Lengkapi Indeks dokumen", and "Keluar". The main content area is titled "Lengkapi Indeks Dokumen" and features a form with several input fields: "ID Dokumen", "NPM", "Author", "Judul Karya Ilmiah", "Kata Kunci", "Tahun Terbit", and "Bulan". There is a "Cari" button next to the "ID Dokumen" field. To the right of the form, there is a section titled "Data Judul karya ilmiah" with a "tahun" dropdown menu and a "Cari" button.

Gambar 4.30 Mockup Lengkapi indeks dokumen

Gambar diatas merupakan perancangan antarmuka atau userinterface tampilan Lengkapi indeks dokumen, didalamnya terdapat fitur untuk melengkapi indeks dokumen yang masih kosong dengan cara memasukan ID Dokumen sesuai dengan dokumen yang akan di edit terlebih dahulu kemudian lakukan proses pelengkapan indeks dokumen dokumen.

BAB V

IMPLEMENTASI DAN PENGUJIAN

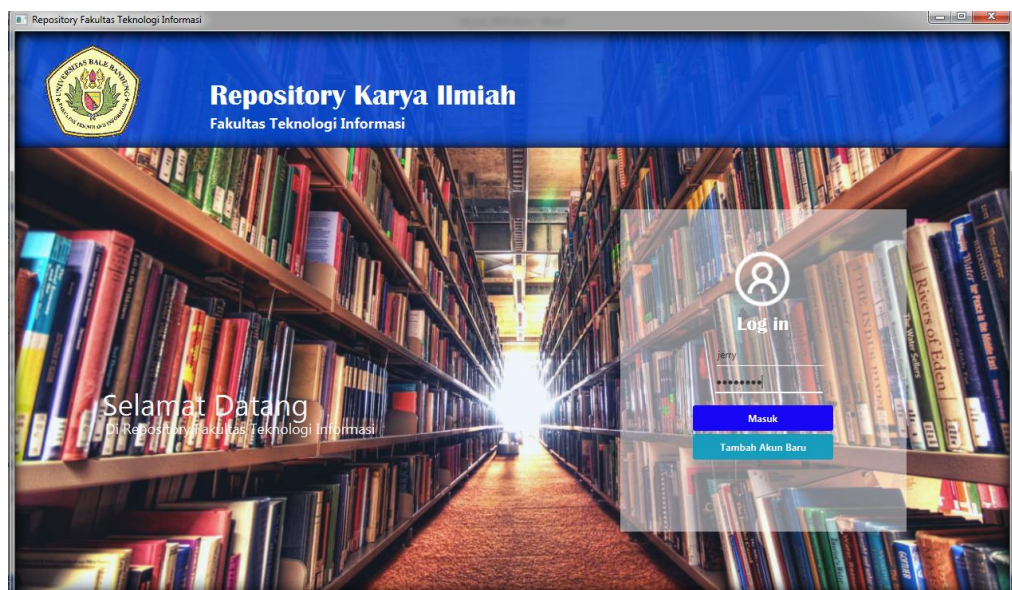
5.1 Implementasi

Implementasi ini dilakukan sesuai dengan perancangan aplikasi yang dilakukan pada bab sebelumnya. Terdiri dari tampilan struktur menu dan pengujian aplikasi. Berikut tampilan struktur menu di aplikasi.

5.1.1 Struktur Menu Form Login

a) Struktur menu Login Admin

Berikut tampilan struktur menu dari form Login Admin:

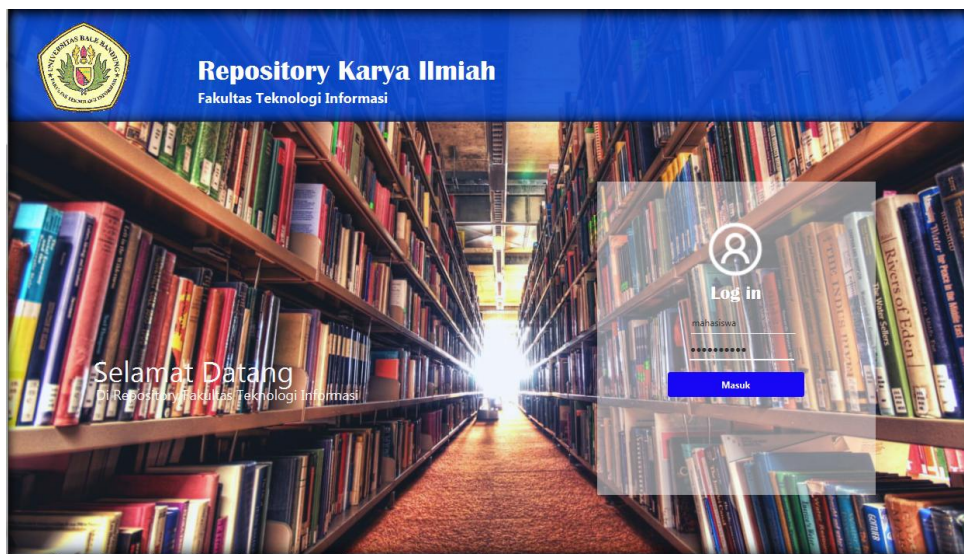


Gambar 5.1 Struktur Menu Form Login

Pada form login ini user diharuskan untuk memasukkan username dan password yang sebelumnya telah terdaftar dengan menekan tombol Masuk dan nantinya akan diarahkan ke Form Menu utama admin Pengelolaan dokumen.

b) Struktur menu Login Mahasiswa.

Berikut tampilan struktur menu dari form Login Mahasiswa:

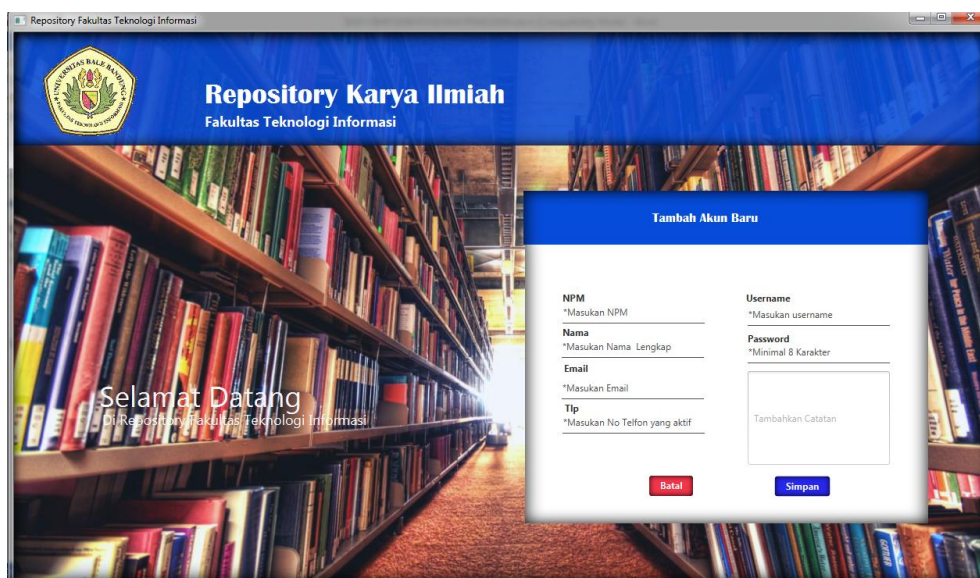


Gambar 5.2 Struktur menu Form Login Mahasiswa

Pada form login ini user diharuskan untuk memasukkan username dan password yang sebelumnya telah terdaftar selanjutnya tekan tombol Masuk untuk masuk ke Menu utama mahasiswa.

5.1.2 Struktur Menu Form Tambah Akun Baru

Berikut tampilan struktur menu dari form tambah akun baru :



Gambar 5.3 Struktur Menu Tambah Akun baru

Sebelum melakukan login, user diharuskan untuk menambah akun baru terlebih dahulu untuk mengakses aplikasi, yang nantinya akan

digunakan untuk login. dan yang memiliki hak akses terhadap form ini hanyalah staff perpustakaan.

5.1.3 Struktur Menu Form Menu Utama Admin

Berikut tampilan struktur menu dari Form menu utama admin :

Gambar 5.4 Struktur Menu Form menu utama admin

Form menu utama admin ini hanya dapat di akses oleh staff perpustakaan selaku admin., Fungsi dari form ini yaitu untuk menambahkan dokumen baru yang di lengkapi dengan indeks pada dokumen tersebut seperti ID Dokumen, Author, Judul Karya Ilmiah dan Tahun terbit, juga terdapat fitur reset akun, hapus data dokumen, edit data dokumen dan dapat melihat data judul karya ilmiah berdasarkan tahun terbit.

5.1.4 Struktur Menu Form Menu Utama Mahasiswa.

Gambar 5.5 Struktur Menu Form Menu Utama Mahasiswa.

Form menu utama Mahasiswa ini hanya dapat di akses oleh mahasiswa yang sudah memiliki akun repository, Fungsi dari form ini yaitu untuk menambahkan dokumen baru yang di lengkapi dengan indeks pada dokumen tersebut seperti ID Dokumen, Author, Judul Karya Ilmiah dan Tahun terbit dan dapat melihat data judul karya ilmiah berdasarkan tahun.

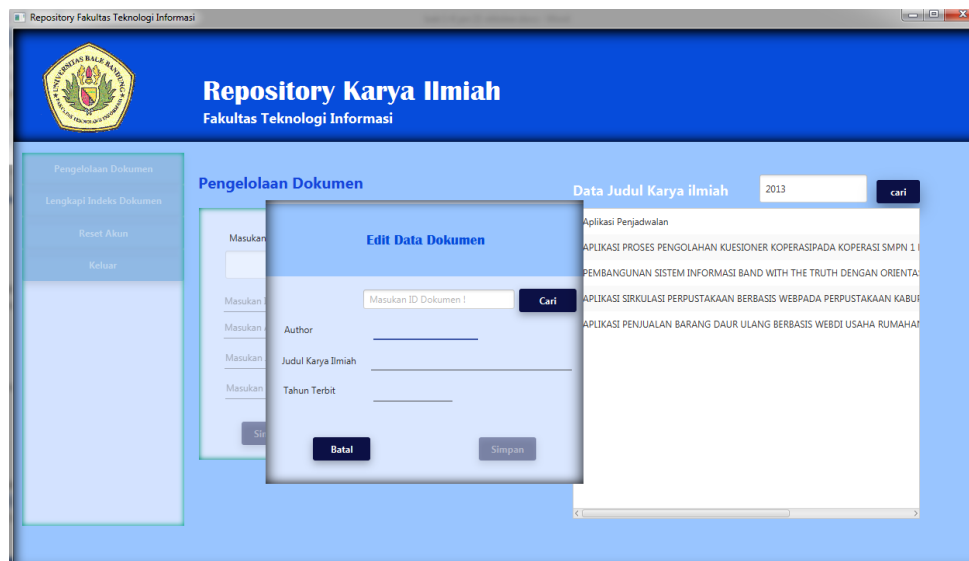
5.1.5 Struktur Menu Form Hapus Dokumen

Gambar 5.6 Struktur Menu Form Hapus Dokumen

Form hapus dokumen ini di gunakan untuk menghapus dokumen beserta indeks pada dokumen dan yang memiliki hak akses terhadap form ini hanyalah staff perpustakaan selaku admin. Pertama admin harus memasukan terlebih dahulu ID Dokumen pada kolom masukan ID Dokumen di form hapus dokumen dan selanjutnya pilih cari. Apabila ID Dokumen ditemukan sistem akan menghapus data dokumen berdasarkan ID Dokumen tersebut. Apabila ID dokumen tidak ditemukan maka aplikasi akan memunculkan pesan “Data yang anda cari tidak tersedia !”kemudian ada tombol batal untuk membatalkan perintah hapus, dan tombol hapus untuk melanjutkan perintah hapus.

5.1.6 Struktur menu Form Edit data dokumen

Berikut adalah tampilan Struktur menu Form Edit data dokumen:



Gambar 5.7 Struktur Menu Form Edit data dokumen

Form Edit data dokumen ini di gunakan untuk mengedit data dokumen dan yang memiliki hak akses terhadap form ini hanyalah staff perpustakaan selaku admin. Di dalam form ini terdapat tombol cari yang digunakan untuk mencari ID Dokumen yang akan diedit, kemudian ada tombol batal untuk membatalkan perintah edit, dan tombol simpan untuk melanjutkan perintah edit.

5.1.7 Struktur Menu Form Reset akun



Gambar 5.8 Struktur Menu Form Reset akun

Form reset akun digunakan untuk menghapus atau mereset akun user, dan yang memiliki hak akses terhadap form reset akun ini hanyalah staff

perpustakaan selaku admin. Dalam form ini terdapat tombol batal untuk membatalkan perintah reset, dan ada tombol hapus untuk melanjutkan proses reset akun.

5.1.8 Struktur menu Form Lengkapi indeks dokumen.

Gambar 5.9 Struktur Menu Form Lengkapi Indeks Dokumen

Form ini digunakan untuk melengkapi indeks pada dokumen yang telah di inputkan. Didalam form ini terdapat tombol cari untuk mencari ID Dokumen yang akan dilengkapi indeksnya, tombol batal untuk membatalkan proses pelengkapan indeks dokumen, dan tombol simpan yang berada paling bawah pada scroll barr untuk melanjutkan ke proses penyimpanan indeks.

5.2 Pengujian

5.2.1 Pengujian *Server Elasticsearch*.

- a. Uji coba keaktifan Elasticsearch melalui browser.

Tabel 5.1 Pengujian Aktif Elasticsearch di Browser

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan

- b. Uji coba keaktifan *Elasticsearch* melalui postman

localhost:9200	Browser menampilkan versi elasticsearch dalam format JSON	Browser menampilkan versi elasticsearch dalam format JSON	[x] diterima [] ditolak
----------------	---	---	-----------------------------

Tabel 5.2 Tampilan Aktif Elasticsearch di Postman

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
GET HTTP://localhost:9200	Postman menampilkan versi elasticsearch dalam format JSON	Postman menampilkan versi elasticsearch dalam format JSON	[x] diterima [] ditolak

- c. Uji coba menambahkan indeks kepada serfer elasticsearch menggunakan postman.

Tabel 5.3 menambahkan indeks kepada serfer elasticsearch

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
http://localhost:9200/skripsi/_doc/01	Indeks skripsi dengan id doc 1 berhasil di tambahkan pada server elasticsearch.	menampilkan respon: "_index": "skripsi", "_type": "_doc", "_id": "01", "_version": 3, "result": "created",	[x] diterima [] ditolak

5.2.2 Pengujian Login

Pada pengujian login terdiri dari beberapa user, yaitu login staff perpustakaan dan login mahasiswa sebagai berikut:

1. Pengujian Login Staff perpustakaan

Tabel 5.4 Pengujian login staff perpustakaan

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Username : admin Password : admin1234 Klik tombol masuk	Form menampilkan masuk untuk bagian staff perpustakaan, berfungsi sebagai pengontrol	Dapat masuk ketampilan utama staff perpustakaan dan	[x] diterima [] ditolak

	kegiatan sistem sehingga dapat mengakses semua data yang tersedia pada sistem	menampilkan pesan "Selamat Datang admin"	
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Username : admin Password : admin123 Klik tombol masuk	Tidak dapat login dan memunculkan pesan "username atau password salah"	Menampilkan pesan "username atau password salah"	[x] diterima [] ditolak

2. Pengujian Login Mahasiswa.

Tabel 5.5 Pengujian Login mahasiswa

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Username : mahasiswa Password : mahasiswa1 Klik tombol masuk	Form menampilkan masuk untuk mahasiswa, berfungsi untuk melakukan penambahan data pada repository.	dapat dapat masuk ketampilan utama mahasiswa.	[x] diterima [] ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Username : mahasiswa Password : mahasis123 Klik tombol masuk	Tidak dapat login dan memunculkan pesan "username atau password salah"	Menampilkan pesan "username atau password salah"	[x] diterima [] ditolak

5.2.3 Pengujian Tambah data baru .

Tabel 5.6 Ujicoba Tambah data baru

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Memasukan file pdf beserta indeks nya seperti IDDoc, Author, Judul Karya Ilmiah, Tahun Terbit	Sistem menyimpan data-data dokumen tersebut dan menyimpannya ke elasticsearch.	Sistem menyimpan data-data dokumen tersebut dan menyimpannya ke elasticsearch kemudian menampilkan pesan "data berhasil disimpan!".	[x] diterima [] ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Memasukan file pdf beserta indeks nya	Sistem tidak dapat menyimpan data-data	Sistem tidak dapat menyimpan data-data	[x] diterima [] ditolak

seperti IDDoc, Author, Judul Karya Ilmiah, Tahun Terbit. Namun ID Doc yang di masukan sudah pernah dimasukan.	dokumen tersebut kedalam elasticsearch.	dokumen tersebut kedalam elasticsearch.	
---	---	---	--

5.2.4 Pengujian Edit data dokumen.

Tabel 5.7 Uji coba Edit data dokumen.

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Memasukan ID Dokumen yang tersedia di kolom pencarian ID Dokumen, kemudian mengedit data,dan klik tombol simpan	Sistem berhasil mengupdate data yang ada pada elasticsearch sesuai ID Dokumen yang telah dimasukan, kemudian muncul pesan Data berhasil di perbaharui.	Sistem berhasil mengupdate data yang ada pada elasticsearch sesuai ID Dokumen yang telah dimasukan, kemudian muncul pesan Data berhasil di perbaharui.	[x] diterima [] ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Memasukan ID Dokumen yang asal atau tidak tersedia di kolom pencarian ID Dokumen kemudian klik tombol cari	menampilkan pesan “ID Dokumen yang anda cari tidak cocok!!!”.	Sistem tidak dapat menemukan data dan menampilkan pesan “ID Dokumen yang anda cari tidak cocok!!!”.	[x] diterima [] ditolak

5.2.5 Pengujian Hapus data dokumen.

Tabel 5.8 Uji coba Hapus data dokumen

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Memasukan ID Dokumen yang tersedia di kolom pencarian ID Dokumen kemudian klik tombol cari kemudian klik tombol hapus.	Sistem berhasil menghapus data yang ada pada elasticsearch sesuai ID Dokumen yang telah dimasukan, kemudian menampilkan pesan Dokumen dengan id tersebut berhasil di hapus.	Sistem berhasil menghapus data yang ada pada elasticsearch sesuai ID Dokumen yang telah dimasukan, kemudian menampilkan pesan Dokumen dengan id tersebut berhasil di hapus	[x] diterima [] ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan

Memasukan ID Dokumen yang tidak tersedia tersedia di kolom pencarian ID Dokumen, kemudian klik tombol cari.	Menampilkan pesan “ID Dokumen yang anda cari tidak cocok!!!”.	Menampilkan pesan “ID Dokumen yang anda cari tidak cocok!!!”.	[x] diterima [] ditolak
---	---	---	-----------------------------

5.2.6 Pengujian Lengkapi indeks dokumen.

Tabel 5.9 Uji Coba lengkapi indeks dokumen

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Memasukan ID Dokumen yang tersedia di kolom pencarian ID Dokumen, kemudian melengkapi indeks ,dan klik tombol simpan	Sistem berhasil melengkapi indeks data yang ada pada elasticsearch sesuai ID Dokumen yang telah dimasukan, kemudian muncul pesan Data berhasil disimpan.	Sistem berhasil melengkapi indeks data yang ada pada elasticsearch sesuai ID Dokumen yang telah dimasukan, kemudian muncul pesan Data berhasil disimpan.	[x] diterima [] ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Memasukan ID Dokumen yang asal atau tidak tersedia tersedia di kolom pencarian ID Dokumen, kemudian klik tombol cari.	Menampilkan pesan “ID Dokumen yang anda cari tidak cocok!!!”	Sistem tidak dapat menemukan ID Dokumen tersebut dan memunculkan pesan “ID Dokumen yang anda cari tidak cocok!!!”.	[x] diterima [] ditolak

5.2.7 Pengujian Pencarian judul karya ilmiah berdasarkan Tahun Terbit

Tabel 5.10 Pengujian Pencarian judul karya ilmiah

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Memasukan Tahun Terbit dan Tekan tombol Cari	Aplikasi menampilkan kumpulan judul skripsi berdasarkan Tahun terbit yang telah di inputkan.	Aplikasi menampilkan kumpulan judul skripsi berdasarkan Tahun terbit yang telah di inputkan.	[x] diterima [] ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan

Memasukan Tahun terbit yang tidak tersedia	Aplikasi tidak memunculkan data dan memunculkan pesan " Data yang anda cari tidak cocok!!"	Aplikasi tidak memunculkan data dan memunculkan pesan " Data yang anda cari tidak cocok!!"	[x] diterima [] ditolak
--	--	--	-----------------------------

5.2.8 Pengujian Tambah akun baru.

Tabel 5.11 Uji coba tambah akun baru

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
data isian lengkap	data user masuk pada database, dan menampilkan pesan "Akun (username) berhasil ditambahkan"	data user masuk pada database, dan menampilkan pesan "Akun (username) berhasil ditambahkan"	[x] diterima [] ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Username tidak valid atau username sudah terpakai.	data user tidak masuk pada database dan menampilkan pesan "username sudah terpakai"	data user tidak masuk pada database dan menampilkan pesan "username sudah terpakai"	[x] diterima [] ditolak

5.2.9 Pengujian Reset akun

Tabel 5.12 Pengujian Reset akun

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Username : mahasiswa	Data user terhapus dari database, dan menampilkan pesan " Akun mahasiswa berhasil direset"	Data user terhapus dari database, dan menampilkan pesan " Akun mahasiswa berhasil direset"	[x] diterima [] ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Username : mahasiswi	menampilkan pesan "Username tidak ditemukan!!!"	menampilkan pesan "Username tidak ditemukan!!!"	[x] diterima [] ditolak

BAB VI

KESIMPULAN DAN SARAN

Dari hasil analisis dan pengujian yang telah dilakukan maka penulis menyimpulkan bahwa

5.1 Kesimpulan

Berdasarkan hasil penelitian dan analisa serta perancangan aplikasi yang telah dihasilkan, maka penulis menyimpulkan bahwa :

1. Dengan adanya aplikasi ini proses pengaksesan data menjadi lebih cepat.
2. Aplikasi ini mampu mempermudah proses temu kenali dokumen.
3. Aplikasi ini dapat Mengefisienkan proses Pengarsipan dokumen karya ilmiah di Fakultas Teknologi Informasi Universitas Bale Bandung.
4. Dengan adanya aplikasi ini proses pengindeksan dokumen menjadi lebih mudah.

5.2 Saran

Aplikasi ini masih memiliki banyak kekurangan, Untuk itu penulis memberikan beberapa saran sebagai berikut:

1. Aplikasi ini hanya sebatas penyimpanan dokumen lokal saja, harapan kedepannya aplikasi ini mampu melakukan akses penyimpanan dokumen lebih luas lagi.
2. File pdf yang mampu di inputkan pada aplikasi ini hanya berukuran 1 MB saja, harapan kedepannya dapat memasukan file pdf yang berukuran lebih besar lagi.
3. Aplikasi ini belum dapat melakukan proses indexing secara otomatis, harapan kedepannya Aplikasi ini dapat melakukan proses indexing secara otomatis dengan menggunakan metode Regex.
4. Aplikasi ini dapat dikembangkan menjadi berbasis web maupun berbasis mobile.

DAFTAR PUSTAKA

- Agustiawan , S.S., M.IP, 2016, Buku Pedoman Pengelolaan Repository Institusi Di Upt Perpustakaan Isi Yogyakarta
- Agustina Simangunsong, 2018, Sistem Informasi Pengarsipan Dokumen Berbasis Web, jurnal Mantik Penusa, Volume 2 No. 1
- Agus Yusida, January 15 2018, Pengertian, Kelebihan Dan Kekurangan Rest Api, <https://ngide.net/posts/apa-itu-rest-api>)
- Arya Febiyan , 02/01/2014, Pengertian JSON, dari <https://www.dumetschool.com/blog/Pengertian-JSON>
- Elya Hestika Asiyah, 2014, Sejarah Uml Dan Jenisnya, Jurnal Ilmu Teknologi Informasi
- Erlangga, 2016/01/07 , Balsamiq Mockup , <http://enggar.net/2016/01/balsamiq-mockup>
- Fahri Firdausillah, 2002, NoSQL: Latar Belakang, Konsep dan Kritik,
- Faizal Anugrah Bhaswara, 2017, Perbandingan Kemampuan Database NoSQL dan SQL dalam Kasus ERP Retail , Jurnal Teknik ITS Vol. 6, No. 2.
- feridi, 21 Jan 2019, Mengenal Restfull API, di kutip di <https://www.codepolitan.com/mengenal-restful-web-services>
- Fikriansyah, 15, Nov, 2017, Apa itu MySQL, Sejarah dan Fungsinya, <https://www.tutorialpedia.net/apa-itu-mysql/>
- Johnson. Rod, a. a. (2014, Januari 23). *Spring Framework Documentation*. Dipetik Juni 16, 2019, dari <http://docs.spring.io/spring/docs/current/springframework-reference/pdf/spring-framework-reference.pdf>
- M. Ardi Yudha Prawir, 2017, Search Engine, dari <https://fryunfirst.blogspot.com/2013/05/search-engine.html>
- Nurhasanah, 2017, Pengelolaan Institutional Repository Di Perpustakaan Utsman Bin Affan Universitas Muslim Indonesia.
- Okkita Rizan, 2013, Bahasa Pemrograman Menggunakan JAVA Dengan Netbeans dan Ireport

- Pelangi, 10, Des, 2018 *NetBeans IDE*, dari <http://pelangiilmu2.blogspot.com/2018/12/assalamualaikum-wr.html>.
- Point star, 2019, kibbana, dari <https://www.pointstar.co.id/id/elastic-search-andlogging/kibana/>
- Pranawa, 19, Maret, 2019, Step by Step Belajar REST API Menggunakan Flask di Windows (Instalasi POSTMAN), <https://blog.pranawa.com/step-by-step-belajar-rest-api-menggunakan-flask-di-windows-instalasi-postman/>
- Rista Novitasari, 2015, Temu Kembali Informasi Menggunakan Elasticsearch Pada Unstructured Datatext Multidimensi,27)
- Rista Novitasari, 2015, Temu Kembali Informasi Menggunakan Elasticsearch Pada Unstructured Datatext Multidimensi,9)
- Rosa A.S, M.Shalahuddin, 2015, Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek, 43).
- Sigit Kurniawan, 2014, Javafx Scene Builder, <https://sigitkurniawanisme.wordpress.com/2015/01/02/javafx-scene-builder/>
- Tester, S. (2013, Maret). *Penjelasan dan fungsi xampp*. Retrieved September 6,2017, from <http://solylight.blogspot.co.id:http://solylight.blogspot.co.id/2013/03/penjelasan-dan-fungsi-xampp.html>
- Ttisnadi, 2019, Model-driven development, <http://trismadi169.blogspot.com/search/label/model-driven%20development>
- Uptodown, Membuat diagram-diagram rumit dengan cara tercepat dan termudah , <https://clickcharts-free-diagram-and-flowchartmaker.id.uptodown.com/windows>

LAMPIRAN

Lampiran 1 Form Wawancara

Narasumber : Ahmad Faozan Muntaha S.KOM

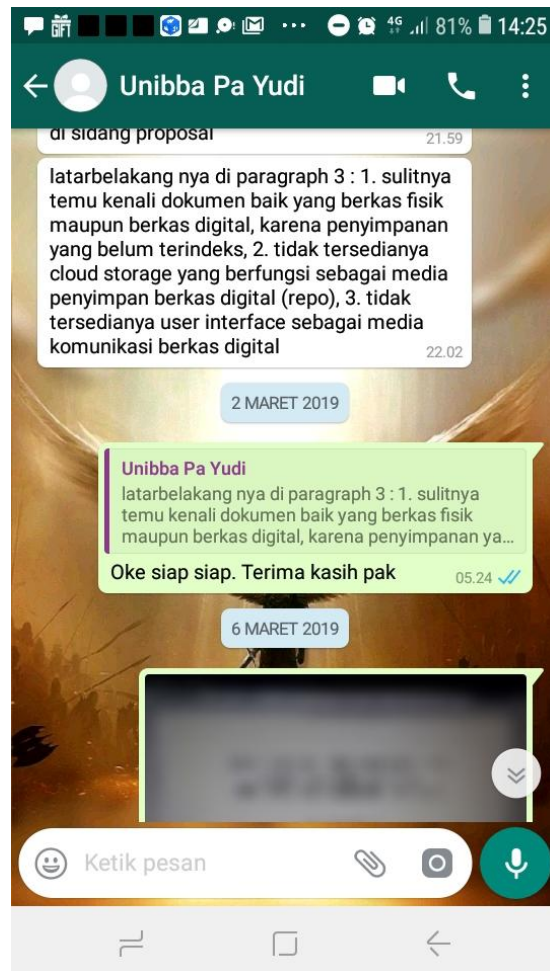
Hari / Tanggal : Senin/ 29 april 2019

Waktu : 15.00

Tempat : Ruang Tata Usaha

No	Pertanyaan	Jawaban	
		Ya	Tidak
1.	Apakah pengumpulan laporan skripsi sudah berjalan dengan tepat waktu?		✓
2.	Apakah pengumpulan laporan skripsi masih manual?	✓	
3.	Apakah penyimpanan laporan skripsi sudah tersusun rapih?	✓	
4.	Apakah ada aplikasi yang digunakan dalam proses pengarsipan dokumen skripsi?	✓	
5.	Apakah dengan menggunakan aplikasi tersebut sudah memudahkan dalam proses pengelolaan dokumen?		✓
6.	Apakah sudah pernah di buat aplikasi repository karya ilmiah sebelumnya?		✓
7.	Apakah dengan akan di terapkan nya aplikasi Repository karya ilmiah akan membantu pihak staff perpustakaan ?	✓	

Narasumber : Yudi Herdiana, S.T, M.T
Hari / Tanggal : Sabtu/ 02 Maret 2019
Waktu : 15.00
Prihal : Menanyakan masalah yang sering terjadi terkait proses pengelolaan dokumen di Fakultas Teknologi Informasi.



Lampiran 2 Listing Program

1. FLoginController.java

```

private void btn_login(ActionEvent event) {
String users = tfuser.getText().toString();
    String username = tfuser.getText();
    connection = ConnectionUtil.connectdb();
    try {
        Statement st = connection.createStatement();
        String sql = "SELECT * FROM tabel_user WHERE username = '"+tfuser.getText()+"' and password = '"+tfpassword.getText()+"'";
        ResultSet rs = st.executeQuery(sql);

        if(rs.next()){
            if(rs.getString("username").equals("admin")){
                user.setUsername(username);
                f_Logiin.getChildren().clear();
                Node [] nodes = new Node[15];
                for(int i = 0; i<1; i++){
                    try {
                        nodes[i] = (Node)FXMLLoader.load(getClass().getResource("/fxml/fPengelolaan_dokumen.fxml"));
                        f_Logiin.getChildren().add(nodes[i]);
                        Alert alrm = new Alert(Alert.AlertType.WARNING);
                        alrm = new Alert(Alert.AlertType.WARNING);
                        alrm.setTitle("pesan");
                        alrm.setHeaderText(null);
                        alrm.setContentText("Selamat Datang "+username);
                        alrm.showAndWait();
                    } catch (IOException ex) {
                        Logger.getLogger(Form_pengelolaan_dokumenController.class.getName()).log(Level.SEVERE, null,
ex);
                    }
                }
            }else{
                user.setUsername(username);
                f_Logiin.getChildren().clear();
                Node [] nodes = new Node[15];
                for(int i = 0; i<1; i++){
                    try {
                        nodes[i] = (Node)FXMLLoader.load(getClass().getResource("/fxml/f_mahasiswa.fxml"));
                        f_Logiin.getChildren().add(nodes[i]);
                        Alert alrm = new Alert(Alert.AlertType.WARNING);
                        alrm = new Alert(Alert.AlertType.WARNING);
                        alrm.setTitle("pesan");
                        alrm.setHeaderText(null);
                        alrm.setContentText("Selamat Datang "+username);
                        alrm.showAndWait();
                    }
                }
            }
        }
    }
}

```

```

    } catch (IOException ex) {
        Logger.getLogger(Form_pengelolaan_dokumenController.class.getName()).log(Level.SEVERE, null,
ex);
    } }
} else {
    Alert alert = new Alert(Alert.AlertType.INFORMATION);
    alert = new Alert(Alert.AlertType.INFORMATION);
    alert.setTitle("pesan");
    alert.setHeaderText(null);
    alert.setContentText("username atau password salah !");
    alert.showAndWait();
}
} catch (SQLException e) {
    e.printStackTrace();
}
}
}

```

2. Pane_tambahakun

```

private void mTambahAkun(){
    String user = tfuser1.getText();
    String NPM = tfnpm.getText();
    String nama = tfnama.getText();
    String email = tfemail.getText();
    String tlp = tftlp.getText();
    String pas = tfpasss.getText();
    String cat = tfcat.getText();
    //var username
    if((user=="")||(NPM=="")||(email=="")||(tlp=="")||(pas=="")||(cat=="")) {
        tdklengkap.setVisible(true);
        alret1.setVisible(false);
    } else {
        try {
            String sql = "INSERT INTO tabel_user VALUES
(""+tfnpm.getText()+""+", ""+tfnama.getText()+""+", ""+tfemail.getText()+""+", ""+tftlp.getText()+""+", ""+tfuser1.getText()+""+", ""+tfp
asss.getText()+""+", ""+tfcat.getText()+""+)";
            java.sql.Connection conn=(Connection)config.configDB();
            java.sql.PreparedStatement pst=conn.prepareStatement(sql);
            pst.execute();
            tdklengkap.setVisible(false);
            alret1.setVisible(false);

            Alert alert = new Alert(Alert.AlertType.INFORMATION);
            alert.setTitle("pesan");
            alert.setHeaderText(null);
            alert.setContentText("Akun "+user+" berhasil ditambahkan");
            alert.showAndWait();
            pane_tambahakun.setVisible(false);
            alret1.setVisible(false);

```

```

tfnpm.setText("");
    tfnama.setText("");
    tfemail.setText("");
    tftlp.setText("");
    tfuser1.setText("");
    tfpasss.setText("");
    tfcat.setText("");
} catch (Exception e) {
    Alert alert = new Alert(Alert.AlertType.WARNING);
    alert.setTitle("pesan");
    alert.setHeaderText(null);
    alert.setContentText("Username sudah terpakai!");
    alert.showAndWait();
    tdklengkap.setVisible(false);
    alret1.setVisible(true);
} } }
}

```

3. Pane Reset akun

```

private void btn_hapusakun(ActionEvent event) {
    String user = "";
    String sql = "";
    ResultSet res= null;

    user = txt_masukan_user.getText();
    try {
        java.sql.Connection conn=(Connection)config.configDB();
        java.sql.PreparedStatement pst;
        sql = "select * from tabel_user where username='"+user+"'";
        pst=conn.prepareStatement(sql);
        res = pst.executeQuery();
        if(!res.next()){
            Alert alert = new Alert(Alert.AlertType.WARNING);
            alert.setContentText("Username tidak ditemukan!!!");
            alert.setTitle("Gagal");
            alert.showAndWait();
        }else{
            sql = "delete from tabel_user where username='"+user+"'";
            pst=conn.prepareStatement(sql);
            pst.execute();
            Alert alert = new Alert(Alert.AlertType.INFORMATION);
            alert.setTitle("pesan");
            alert.setHeaderText(null);
            alert.setContentText("Akun " +user+" berhasil Reset !");
            alert.showAndWait();
        }
    }
}

```

```

pane_reset.setVisible(false);
    idinput.setDisable(false);
    idvbox.setDisable(false);
    txt_masukan_user.setText("");
}
} catch (Exception h) {

```

4. Postt.java

```

public class postt {
    public void postResponse(String iddoc, String author, String title, Integer year, String fulteks) {try {
        URL url = new URL("http://localhost:9200/skripsi/_doc/"+iddoc);
        HttpURLConnection conn = (HttpURLConnection) url.openConnection();
        conn.setDoOutput(true);
        conn.setRequestMethod("POST");
        conn.setRequestProperty("Content-Type", "application/json");
        String input =
            "{\"IDDoc\":\""+iddoc+"\", \"Author\":\""+author+"\", \"Title\":\""+title+"\", \"Year\":\""+year+"\", \"Fulltext\":\""+fulteks+"\"}";
        OutputStream os = conn.getOutputStream();
        os.write(input.getBytes());
        os.flush();
        if (conn.getResponseCode() != HttpURLConnection.HTTP_CREATED) {
            Alert alert = new Alert(Alert.AlertType.INFORMATION);
            alert = new Alert(Alert.AlertType.INFORMATION);
            alert.setTitle("Pesan");
            alert.setHeaderText(null);
            alert.setContentText("Data gagal disimpan \n Kode Error :"+ conn.getResponseCode());
            alert.show();
            throw new RuntimeException("Failed : HTTP error code : " + conn.getResponseCode())
            BufferedReader br = new BufferedReader(new InputStreamReader(
                (conn.getInputStream())));

            String output;
            System.out.println("Output from Server .... \n");
            while ((output = br.readLine()) != null) {
                System.out.println(output);
                Alert alert = new Alert(Alert.AlertType.INFORMATION);
                alert = new Alert(Alert.AlertType.INFORMATION);
                alert.setTitle("Pesan");
                alert.setHeaderText(null);
                alert.setContentText("Data berhasil disimpan");
                alert.showAndWait();
            }
            conn.disconnect();
        } catch
        (MalformedURLException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

```

5. Update.java

```

Public class updatee {
public void updateResponse(String iddoc, String author, String title, Integer year) {
try {
    URL url = new URL("http://localhost:9200/skripsi/_doc/"+iddoc);
    HttpURLConnection conn = (HttpURLConnection) url.openConnection();
    conn.setDoOutput(true);
    conn.setRequestMethod("PUT");
    conn.setRequestProperty("Content-Type", "application/json");

    String input =
        "{\"IDDoc\":\""+iddoc+"\", \"Author\":\""+author+"\", \"Title\":\""+title+"\", \"Year\":\""+year+"\"}";
    OutputStream os = conn.getOutputStream();
    os.write(input.getBytes());
    os.flush();

    BufferedReader br = new BufferedReader(new InputStreamReader(
        (conn.getInputStream())));

    String output;
    System.out.println("Output from Server .... \n");
    while ((output = br.readLine()) != null) {
    if (conn.getResponseCode() != HttpURLConnection.HTTP_CREATED) {
        Alert alm = new Alert(Alert.AlertType.INFORMATION);
        alm = new Alert(Alert.AlertType.INFORMATION);
        alm.setTitle("pesan");
        alm.setHeaderText(null);
        alm.setContentText("Dokumen berhasil disimpan");
        alm.showAndWait();

        throw new RuntimeException("data berhasil disimpan"+output);

    }
    System.out.println(conn.getResponseCode());
    Alert alr = new Alert(Alert.AlertType.INFORMATION);
    alr = new Alert(Alert.AlertType.INFORMATION);
    alr.setTitle("pesan");
    alr.setHeaderText(null);
    alr.setContentText("Dokumen gagal disimpan");
    alr.showAndWait();
    throw new RuntimeException("Failed : HTTP error code : "
        );
    }
    conn.disconnect();
} catch (MalformedURLException e) {
e.printStackTrace();
} catch (IOException e) {
e.printStackTrace();
}
} }

```

6. delete.java

```

public class deletee {
    public void deleteResponse(String id ) {
        try {
            com.sun.jersey.api.client.Client client = com.sun.jersey.api.client.Client.create();

            WebResource webResource = client.resource("http://localhost:9200/skripsi/_doc/"+id);
            ClientResponse response = webResource.accept("")
                .delete(ClientResponse.class);
            if (response.getStatus() != 200) {
                Alert alm = new Alert(Alert.AlertType.WARNING);
                alm = new Alert(Alert.AlertType.WARNING);
                alm.setTitle("pesan");
                alm.setHeaderText(null);
                alm.setContentText("maaf ID Dokumen tidak tersedia");
                alm.showAndWait();
            }
        }
    }
}

```

```

        throw new RuntimeException("Failed : HTTP error code : " +
            response.getStatus());
    }
    String output2 = response.getEntity(String.class);
    //System.out.println("\n=====Plain Text Response=====");
    System.out.println(output2);
    Alert alrm = new Alert(Alert.AlertType.INFORMATION);
    alrm = new Alert(Alert.AlertType.INFORMATION);
    alrm.setTitle("pesan");
    alrm.setHeaderText(null);
    alrm.setContentText("Dokumen dengan Id "+id+" berhasil hapus");
    alrm.showAndWait();
} catch (Exception e) {
    e.printStackTrace();
}
} //end of method
}

```

7. Lengkapi_dok.java

```

public class lengkapi_dok {
    public void lrResponse(String iddoc, String npm, String author,String title, String key,
        Integer year, String month, String stat, Integer tags, String full) {

        try {
            URL url = new URL("http://localhost:9200/skripsi/_doc/"+iddoc);
            HttpURLConnection conn = (HttpURLConnection) url.openConnection();
            conn.setDoOutput(true);
            conn.setRequestMethod("PUT");
            conn.setRequestProperty("Content-Type", "application/json");

            String input =
                "{\"IDDoc\":\""+iddoc+"\", \"NPM\":\""+npm+"\", \"Author\":\""+author+"\", \"Title\":\""+title+"\", \"Keywords\":\""+key+"\", \"Year\":\""+year+"\", \"Month\":\""+month+"\", \"Status\":\""+stat+"\", \"Tags\":\""+tags+"\", \"Fulltext\":\""+full+"\"}";
            OutputStream os = conn.getOutputStream();
            os.write(input.getBytes());
            os.flush();

            BufferedReader br = new BufferedReader(new InputStreamReader(
                (conn.getInputStream())));

            String output;
            System.out.println("Output from Server .... \n");
            while ((output = br.readLine()) != null) {
                if (conn.getResponseCode() != HttpURLConnection.HTTP_CREATED) {
                    Alert alrm = new Alert(Alert.AlertType.INFORMATION);
                    alrm = new Alert(Alert.AlertType.INFORMATION);
                    alrm.setTitle("pesan");
                    alrm.setHeaderText(null);
                    alrm.setContentText("Dokumen berhasil disimpan");
                    alrm.showAndWait();
                    throw new RuntimeException("data berhasil disimpan"+output);
                }
            }
            System.out.println(conn.getResponseCode());
            Alert alr = new Alert(Alert.AlertType.INFORMATION);
            alr = new Alert(Alert.AlertType.INFORMATION);
            alr.setTitle("pesan");
            alr.setHeaderText(null);
            alr.setContentText("Dokumen gagal disimpan");
            alr.showAndWait();
            throw new RuntimeException("Failed : HTTP error code : "
                );
        } catch (MalformedURLException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

```

8. Parse.java

```

public class Parse {
    @FXML
    private AnchorPane formpengelolaandokumen;
    @FXML
    private TextField tFullText;

    public static void main(final String[] args) throws IOException, TikaException, SAXException {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Form_pengelolaan_dokumenController().setVisible(true);
            }
        });
    }

    public String convertPdf(String fileName) throws FileNotFoundException, IOException,
    SAXException, TikaException{

        BodyContentHandler handler = new BodyContentHandler(-1);
        Metadata metadata = new Metadata();
        FileInputStream inputstream = new FileInputStream(new File(fileName));
        ParseContext pcontext = new ParseContext();

        //parsing the document using PDF parser
        AutoDetectParser parser = new AutoDetectParser ();
        parser.parse(inputstream, handler, metadata,pcontext);

        //getting the content of the document
        // System.out.println("Contents of the PDF :"+ handler.toString());

        //getting metadata of the document
        System.out.println("Metadata of the PDF:");
        String[] metadataNames = metadata.names();

        for(String name : metadataNames) {

            System.out.println(name+ " : " + metadata.get(name));
        }
        Alert alert = new Alert(Alert.AlertType.INFORMATION);
        alert.setTitle("pesan");
        alert.setHeaderText(null);
        alert.setContentText("Contents of the PDF :"+ handler.toString());
        alert.showAndWait();
        // final String full=("Contents of the PDF :"+ handler.toString());
        return handler.toString();
    }
}

```

9. ConnectionUtil.java

```

package com.mycompany.util;
import java.sql.*;
import javax.swing.*;
public class ConnectionUtil {

    public static Object getConnection() {
        throw new UnsupportedOperationException("Not supported yet."); //To change body
of generated methods, choose Tools | Templates.
    }
    Connection conn = null;
    public static Connection connectdb()
    {
        try{
            Class.forName("com.mysql.jdbc.Driver");
            Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost/storageing","root","");
            return conn;
        }
        catch(Exception e)
        {
            JOptionPane.showMessageDialog(null, e);
            return null;
        }
    }
}

```

10. User.java

```

public class User {
    private ObservableList file;
    private String author;
    private String title ;
    private Integer year;
    public ObservableList getFile() {
        return file; }
    public void setFile(ObservableList file) {
        this.file= file;
    }
    public String getAuthor() {
        return author;
    }
    public void setAuthor(String author) {
        this.author = author;
    }
    public String getTitle() {
        return title;
    }
    public void setTitle(String title) {
        this.title = title;
    }
    public Integer getYear() {
        return year ;
    }
    public void setYear(Integer year) {
        this.year = year;
    }
}

```


11. Mencari judul dokumen

```

private void btn_data_judul(ActionEvent event) throws MalformedURLException,
IOException {

    String value = txt_caritahun.getText();
    String url = "";
    URL urlObject = null;
    URLConnection con = null;
    String response = "";
    JSONArray hitsArray = null;
    JSONObject hits = null;
    JSONObject source = null;
    JSONObject json = null;

    if(value==null){
        Alert alert = new Alert(Alert.AlertType.WARNING);
        alert.setTitle("Pencarian");
        alert.setHeaderText(null);
        alert.setContentText("Mohon diisi tahun Terlebih Dahulu!");
        alert.showAndWait();
    } else {
        //MENCARI DATA DARI ELASTICSEARCH
        try {
            url = "http://localhost:9200/skripsi/_search?q=Year:"+value;
            urlObject = new URL(url);
            con = urlObject.openConnection();
            con.setRequestProperty("User-Agent", "Mozilla/5.0");
            buffer = new BufferedInputStream(con.getInputStream());
            while (buffer.available()>0) {
                response += (char)buffer.read();
            }
            buffer.close();
            json = new JSONObject(response);
            hits = json.getJSONObject("hits");
            hitsArray = hits.getJSONArray("hits");

            int jmlHits = hitsArray.length();

            if(jmlHits==0){
                Alert alert = new Alert(Alert.AlertType.ERROR);
                alert.setTitle("Pencarian");
                alert.setHeaderText(null);
                alert.setContentText("Data yang anda cari tidak cocok!!!");
                alert.showAndWait();
            } else {
                lv_judul.getItems().clear();
                for (int i=0; i<jmlHits; i++) {
                    JSONObject h = hitsArray.getJSONObject(i);
                    source = h.getJSONObject("_source");
                    String object = (source.getString("Title"));
                    lv_judul.getItems().addAll(object);
                }
            }
        } catch (Exception e) {

        }
    }
}

```

12. Pencarian berdasarkan ID Dokumen

```

private void btn_carilD(ActionEvent event) {

    String iddocx = txtcaridok.getText();
    BufferedInputStream buffer = null;
    String url = "";
    URL urlObject = null;
    URLConnection con = null;
    String response = "";
    JSONArray hitsArray = null;
    JSONObject hits = null;
    JSONObject source = null;
    JSONObject json = null;

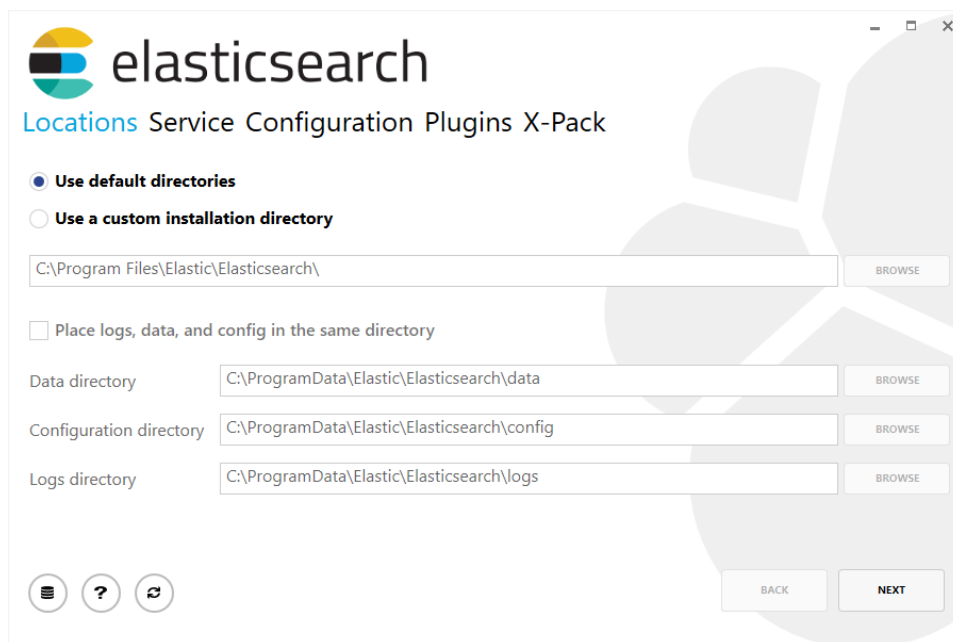
    if(iddocx==""){
        Alert alert = new Alert(Alert.AlertType.WARNING);
        alert.setTitle("Pencarian");
        alert.setHeaderText(null);
        alert.setContentText("Mohon di isi ID Dokumen! ");
        alert.showAndWait();
    } else {
        //MENCARI DATA DARI ELASTICSEARCH
        try {
            url = "http://localhost:9200/skripsi/_search?q=IDDoc:"+iddocx;
            urlObject = new URL(url);
            con = urlObject.openConnection();
            con.setRequestProperty("User-Agent", "Mozilla/5.0");
            buffer = new BufferedInputStream(con.getInputStream());
            while (buffer.available()>0) {
                response += (char)buffer.read();
            }
            buffer.close();
            json = new JSONObject(response);
            hits = json.getJSONObject("hits");
            hitsArray = hits.getJSONArray("hits");
            int jmlHits = hitsArray.length();

            if(jmlHits==0){
                Alert alert = new Alert(Alert.AlertType.ERROR);
                alert.setTitle("Pencarian");
                alert.setHeaderText(null);
                alert.setContentText("ID Dokumen yang anda cari tidak cocok!!!");
                alert.showAndWait();
            } else {
                for (int i=0; i<jmlHits; i++) {
                    JSONObject h = hitsArray.getJSONObject(i);
                    source = h.getJSONObject("_source");
                    String nama = (source.getString("Author"));
                    String judul = (source.getString("Title"));
                    int tahun;
                    tahun= (source.getInt ("Year"));
                    authorr.setText(nama);
                    titlee.setText(judul);
                    yearr.setText(Integer.toString(tahun));
                    hapusdok.setDisable(false);
                }
            }
        } catch (Exception e) {
        }
    }
}

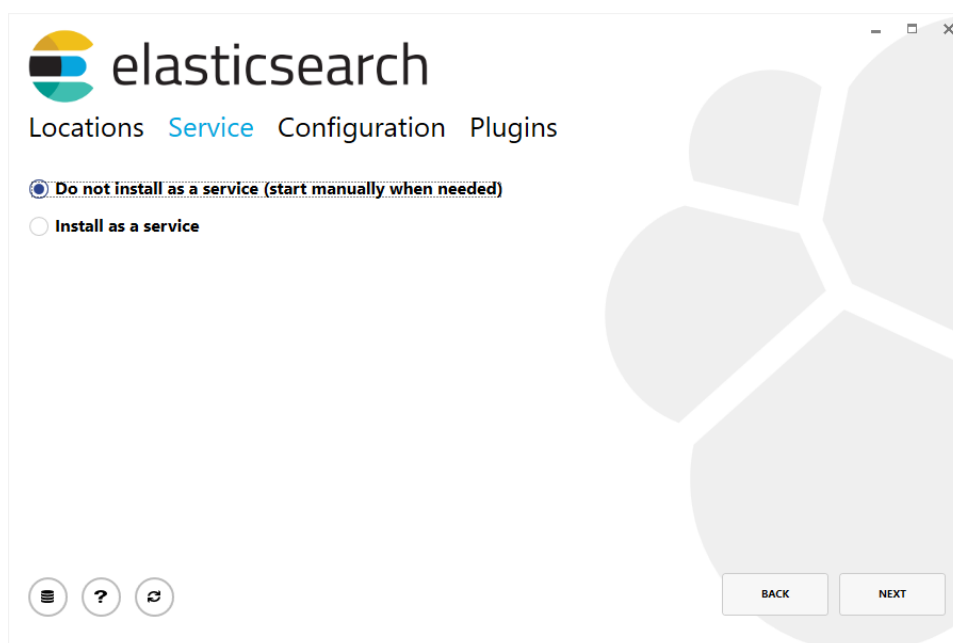
```

Lampiran 3 Instalasi Elasticsearch 6.5.4

1. Download Elasticsearch 6.5.4 MSI di:
<https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-6.5.4.msi>.
2. Klik dua kali file yang diunduh untuk memunculkan GUI. Di dalam layar pertama, pilih direktori penempatan:



3. Kemudian pilih “Do not install a service”. Untuk menyelaraskan dengan contoh tar, pilih untuk tidak menginstal sebagai layanan:



4. Untuk konfigurasi, cukup tinggalkan nilai default:

elasticsearch

Locations Service **Configuration** Plugins X-Pack

Identifiers

Cluster name:

Node name:

Roles

☒ Data ☒ Master ☒ Ingest

Memory

2 GB/15.78 GB

☐ Lock JVM memory

Network (Optional)

Network host:

HTTP port: Transport port:

Discovery (Optional)

Minimum master nodes not set

Unicast Hosts:

BACK NEXT

5. Sekali lagi, untuk menyelaraskan dengan contoh tar, hapus centang semua plugin untuk tidak menginstal plugins:

elasticsearch

Locations Service Configuration **Plugins X-Pack**

☐ **Ingest Attachment Processor**
The ingest attachment plugin lets Elasticsearch extract file attachments in common formats (such as PPT, XLS, and PDF) by using the Apache text and metadata extraction library Tika. You can use the ingest attachment plugin as a replacement for the mapper attachment plugin.

☐ **Ingest GeoIP Processor**
The GeoIP processor adds information about the geographical location of IP addresses, based on data from Geo-IP databases. This processor adds this information by default under the geoip field.

☐ **ICU Analysis**
The ICU Analysis plugin integrates the Lucene ICU module into Elasticsearch, adding extended Unicode support using the ICU libraries, including better analysis of Asian languages, Unicode normalization, Unicode-aware case folding, collation support, and transliteration.

☐ **Japanese (kuromoji) Analysis**
The Japanese (kuromoji) Analysis plugin integrates the Lucene kuromoji analysis module into Elasticsearch.

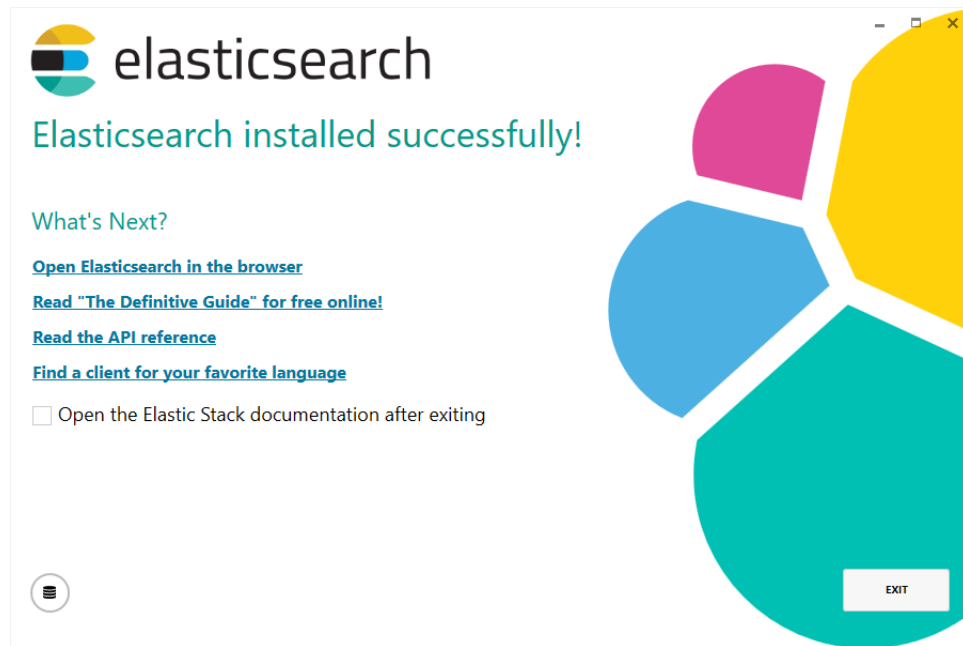
☐ **Phonetic Analysis**
The Phonetic Analysis plugin provides token filters which convert tokens to their phonetic representation using Soundex, Metaphone, and a variety of other algorithms.

☐ **Smart Chinese Analysis**
The Smart Chinese Analysis plugin integrates the Lucene Smart Chinese analysis module into Elasticsearch.

☐ **Custom Plugin Analysis**

PROXY BACK NEXT

6. Setelah mengklik tombol install, Elasticsearch akan diinstal:



7. Secara default, Elasticsearch akan diinstal di% PROGRAMFILES% \ Elastic \ Elasticsearch. Navigasi di sini dan masuk ke direktori bin sebagai berikut:

```
cd %PROGRAMFILES%\Elastic\Elasticsearch\bin
```

8. Berhasil menjalankan simpul. Jika semuanya berjalan baik dengan instalasi, Anda akan melihat banyak pesan yang terlihat seperti di bawah ini:

```
Elasticsearch 6.5.4
[2019-08-16T21:17:51,543][INFO ][o.e.p.PluginsService] [LAPTOP-IH21NM4U] loaded module [x-pack-rollup]
[2019-08-16T21:17:51,543][INFO ][o.e.p.PluginsService] [LAPTOP-IH21NM4U] loaded module [x-pack-security]
[2019-08-16T21:17:51,543][INFO ][o.e.p.PluginsService] [LAPTOP-IH21NM4U] loaded module [x-pack-sql]
[2019-08-16T21:17:51,543][INFO ][o.e.p.PluginsService] [LAPTOP-IH21NM4U] loaded module [x-pack-upgrade]
[2019-08-16T21:17:51,543][INFO ][o.e.p.PluginsService] [LAPTOP-IH21NM4U] loaded module [x-pack-watcher]
[2019-08-16T21:17:51,543][INFO ][o.e.p.PluginsService] [LAPTOP-IH21NM4U] no plugins loaded
[2019-08-16T21:18:05,677][INFO ][o.e.x.m.j.p.l.CppLogMessageHandler] [LAPTOP-IH21NM4U] [controller/16720] [Main.cc@109]
controller (64 bit): Version 6.5.4 (Build b616805ef32893) Copyright (c) 2018 Elasticsearch BV
[2019-08-16T21:18:10,959][INFO ][o.e.d.DiscoveryModule] [LAPTOP-IH21NM4U] using discovery type [zen] and host provid
ers [settings]
[2019-08-16T21:18:12,858][INFO ][o.e.n.Node] [LAPTOP-IH21NM4U] initialized
[2019-08-16T21:18:12,858][INFO ][o.e.n.Node] [LAPTOP-IH21NM4U] starting ...
[2019-08-16T21:18:14,827][INFO ][o.e.t.TransportService] [LAPTOP-IH21NM4U] publish_address {127.0.0.1:9300}, bound_ad
dresses {127.0.0.1:9300}, {[::]:9300}
[2019-08-16T21:18:19,040][INFO ][o.e.c.s.MasterService] [LAPTOP-IH21NM4U] zen-disco-elected-as-master ([0] nodes joi
ned), reason: new master {LAPTOP-IH21NM4U}{P7otjqXgTIOUicucr1x3g}{cm7YDs0yRYC6u4MNdewH2Q}{127.0.0.1}{127.0.0.1:9300}}{ml
.machine.memory=4109561856, xpack.installed=true, ml.max_open_jobs=20, ml.enabled=true}
[2019-08-16T21:18:19,056][INFO ][o.e.c.s.ClusterApplierService] [LAPTOP-IH21NM4U] new master {LAPTOP-IH21NM4U}{P7otjqXgT
IOUicucr1x3g}{cm7YDs0yRYC6u4MNdewH2Q}{127.0.0.1}{127.0.0.1:9300}}{ml.machine.memory=4109561856, xpack.installed=true, ml
.max_open_jobs=20, ml.enabled=true}, reason: apply cluster state (from master {master {LAPTOP-IH21NM4U}{P7otjqXgTIOUicuc
r1x3g}{cm7YDs0yRYC6u4MNdewH2Q}{127.0.0.1}{127.0.0.1:9300}}{ml.machine.memory=4109561856, xpack.installed=true, ml.max_op
en_jobs=20, ml.enabled=true} committed version [1] source [zen-disco-elected-as-master ([0] nodes joined)])
[2019-08-16T21:18:19,092][INFO ][o.e.h.n.KettyHttpClientTransport] [LAPTOP-IH21NM4U] publish_address {127.0.0.1:9200},
bound_addresses {127.0.0.1:9200}, {[::]:9200}
[2019-08-16T21:18:19,096][INFO ][o.e.n.Node] [LAPTOP-IH21NM4U] started
[2019-08-16T21:18:28,756][INFO ][o.e.c.s.ClusterSettings] [LAPTOP-IH21NM4U] updating [xpack.monitoring.collection.enab
led] from [false] to [true]
[2019-08-16T21:18:28,055][INFO ][o.e.l.licenseService] [LAPTOP-IH21NM4U] license [af3eb541-355a-4093-9700-63bd36754
bdf] mode [basic] - valid
[2019-08-16T21:18:28,695][INFO ][o.e.g.GatewayService] [LAPTOP-IH21NM4U] recovered [46] indices into cluster state
```

Lampiran 4 Kartu Bimbingan

PERBAIKAN PENELITIAN / SKRIPSI PEMBIMBING 1

TANGGAL Bimbingan	BAB, BAGIAN YANG HARUS DIPERBAIKI	PARAF PEMBIMBING
	Bab 2: V. pembekal (angasan) perniagaan sesuai dengan 9. Semua petunjuk harus tersedia	
	Bab 3: Labuan Kuala Perlis - mengabaikan	
	Bab 4: Kaji ulang hal yang benar-benar ditanyakan.	

PERBAIKAN PENELITIAN / SKRIPSI PEMBIMBING 2

TANGGAL Bimbingan	BAB, BAGIAN YANG HARUS DIPERBAIKI	PARAF PEMBIMBING
24/4 2019	perbaikan bab 7 ganti CAD dgn foto.	
9/5 2019	letak selatun perbaikan gambar implem. ditambail. gambar foto tembak.	
11/5 2019	perubahan teknik ElasticScan	
15/5 2019	buatkan alur diagram metode penelitian pd tahap investigasi	
13/6 2019	perbaiki perangkaan bahin metode penelitian	
17/6 2019	perbaiki bahin metode penelitian	
20/7 2019	perbaiki sequence diagram	
24/7 2019	sequence diagram	
6/8 2019	perbaiki Ekstensi Jsw dan file	

Lampiran 5**RIWAYAT HIDUP PENULIS**

Penulis bernama Jeri Rismawan, dilahirkan pada Tanggal 05 Mei 1995 di Bandung. Penulis adalah seorang anak dari pasangan suami istri yang bernama Bapak Parman dan Ibu Yayah Rokayah. Pada tahun 2007 penulis lulus Sekolah Dasar di SDN JATI, pada tahun 2010 penulis lulus Sekolah Menengah Pertama di SMP PASUNDAN 1 Banjaran, pada tahun 2013 penulis lulus sekolah di Sekolah Menengah Kejuruan di SMK PASUNDAN 2 Banjaran dan pada tahun 2015 penulis masuk di UNIVERSITAS BALE BANDUNG (UNIBBA) yang In Sya Alloh pada tahun ini mengantarkan penulis untuk mendapatkan gelar Sarjana Strata Satu.

Demikian Riwayat hidup penulis untuk sekedar diketahui.

Terima Kasih

Penulis