



*Context-Aware untuk Natural Language Processing Services menggunakan
Service Oriented Architecture*

Abdullah Aziz Sembada
18917102

*الجامعة الإسلامية
الاندونيسية*
Tesis diajukan sebagai syarat untuk meraih gelar Magister Komputer

*Konsentrasi Sistem Informasi Enterprise
Program Studi Informatika Program Magister
Fakultas Teknologi Industri
Universitas Islam Indonesia*

2023

Lembar Pengesahan Pembimbing

Context-Aware untuk Natural Language Processing Services menggunakan
Service Oriented Architecture

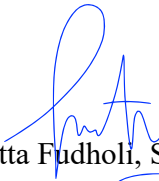
Abdullah Aziz Sembada

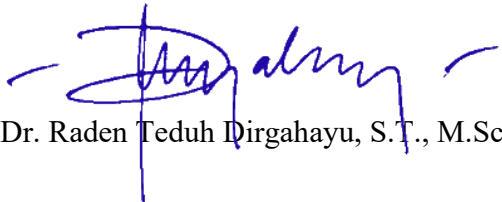
18917102



Pembimbing 1

Pembimbing 2


Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.


Dr. Raden Teduh Dirgahayu, S.T., M.Sc.

Lembar Pengesahan Penguji

**Context-Aware untuk Natural Language Processing Services menggunakan
Service Oriented Architecture**

Abdullah Aziz Sembada
189171102

Yogyakarta, 16 Januari 2023

Tim Penguji,

Dhomas Hatta Fudholi, ST., M.Eng, Ph.D

Ketua

Dr. R Teduh Dirgahayu, ST., M.Sc

Anggota I

Dr. Mukhammad Andri Setiawan, ST., M.Sc.

Anggota II

Mengetahui,

Ketua Program Studi Informatika Program Magister

Universitas Islam Indonesia

Irving Vitra Papatungan, S.T., M.Sc., Ph.D.

Abstrak

***Context-Aware* untuk Natural Language Processing Services menggunakan Service Oriented Architecture**

Berkembangnya era industri 4.0 dan *big data* membuat *Natural Language Processing* banyak dibutuhkan terutama saat melakukan *preprocessing* data. Dengan adanya *Natural Language Processing service* agar bisa mempermudah peneliti dalam melakukan penelitian, karena *NLP service* sudah menyediakan beberapa algoritma. Adapun langkah-langkah yang dilakukan untuk melakukan implementasi *Context-Aware* untuk *NLP Service* adalah pengumpulan data, *refactoring source code*, *containerization*, pemodelan *Context-Aware*, implementasi sistem, pengujian sistem.

Pengumpulan data pada penelitian ini adalah mencari *source code* dengan bahasa python untuk dilakukan implementasi. *Refactoring source code* dilakukan supaya *source code* yang didapat bisa di *Containerization* dengan baik. Tantangan pada tahap ini adalah penggunaan *library* yang telah usang, ini menghambat saat *Containerization* karena *library* yang telah usang tidak bisa dipasang lagi atau kode dasarnya tidak mendukung dengan *library* terbaru.

Pemodelan *Context-Aware* adalah menentukan jenis sebuah *file* termasuk jenis *categorical* atau *uncategorical*. Jika jumlah data unik pada fitur kurang atau sama dengan tiga (≤ 3) maka *file* tersebut dikategorikan *categorical* dan jika lebih dari tiga maka dikategorikan *uncategorical*. Kondisi ini dipilih karena semakin sedikit data unik pada fitur semakin mudah menentukan kategori pada setiap baris datanya.

Implementasi sistem pada tahap ini adalah melakukan integrasi pada semua *docker container* untuk bisa saling terhubung dan berkomunikasi. *SOA* juga sangat membantu dalam pengembangan terutama saat penambah aplikasi karena setiap aplikasi terisolasi oleh *container*.

Kata kunci

SOA, natural language processing, Context-Aware

Abstract

***Context-Aware* untuk Natural Language Processing Services menggunakan Service Oriented Architecture**

The development of the industrial era 4.0 and big data has made Natural Language Processing much needed, especially when preprocessing data. Having a Natural Language Processing service makes it easier for researchers to do research, because the NLP service already provides several algorithms. The steps taken to implement Context-Aware for NLP Services are data collection, source code refactoring, containerization, Context-Aware modeling, system implementation, system testing.

Collecting data in this study is looking for source code in Python for implementation. Refactoring the source code is done so that the source code obtained can be properly containerized. The challenge at this stage is using obsolete libraries, this hinders containerization because obsolete libraries can't be installed anymore or the base code doesn't support the latest libraries.

Context-Aware Modeling is to determine the type of a file including categorical or uncategorical type. If the number of unique data on a feature is less than or equal to three (≤ 3) then the file is categorized as categorical and if there are more than three then it is categorized as uncategorical. This condition was chosen because the less unique data the feature has, the easier it is to determine the category for each row of data.

System implementation at this stage is to integrate all Docker containers to be able to connect and communicate with each other. SOA is also very helpful in development, especially when adding applications because each application is isolated by a container.

Keywords

SOA, natural language processing, *Context-Aware*.

Pernyataan Keaslian Tulisan

Dengan ini saya menyatakan bahwa tesis ini merupakan tulisan asli dari penulis, dan tidak berisi material yang telah diterbitkan sebelumnya atau tulisan dari penulis lain terkecuali referensi atas material tersebut telah disebutkan dalam tesis. Apabila ada kontribusi dari penulis lain dalam tesis ini, maka penulis lain tersebut secara eksplisit telah disebutkan dalam tesis ini.

Dengan ini saya juga menyatakan bahwa segala kontribusi dari pihak lain terhadap tesis ini, termasuk bantuan analisis statistik, desain survei, analisis data, prosedur teknis yang bersifat signifikan, dan segala bentuk aktivitas penelitian yang dipergunakan atau dilaporkan dalam tesis ini telah secara eksplisit disebutkan dalam tesis ini.

Segala bentuk hak cipta yang terdapat dalam material dokumen tesis ini berada dalam kepemilikan pemilik hak cipta masing-masing. Apabila dibutuhkan, penulis juga telah mendapatkan izin dari pemilik hak cipta untuk menggunakan ulang materialnya dalam tesis ini.

Yogyakarta, 3 Januari 2023


Abdullah Aziz Sembada

Daftar Publikasi

Publikasi yang menjadi bagian dari tesis

Sembada, A. A., Fudholi, D. H., & Dirgahayu, R. T. (2022). *Context-Aware for Natural Language Processing Services using Microservices Architecture*. *Budapest International Research and Critics Institute (BIRCI-Journal)*, Vol 5, No 2 (2022), 16379–16389.
<https://doi.org/10.33258/birci.v5i2.5553>

| Kontributor | Jenis Kontribusi |
|-----------------------|---|
| Abdullah Aziz Sembada | Menulis dan mereview artikel |
| Dhomas Hatta Fudholi | Memberi ide dan saran Mereview artikel |
| Raden Teduh Dirgahayu | Memberi ide dan saran Mereview artikel |

Halaman Kontribusi

Penelitian ini tidak terlepas dari berbagai saran maupun bimbingan dari berbagai pihak, mulai dari pra penelitian, seminar proposal, seminar progres, hingga seminar pendadaran. Pihak-pihak tersebut, antara lain: Bapak Dthomas Hatta Fudholi, S.T., M.Eng., Ph.D., dan Dr. R. Teduh Dirgahayu, S.T., selaku dosen pembimbing dan teman – teman di stackoverflow.com.



Halaman Persembahan

Saya membuat Tesis ini teruntuk:

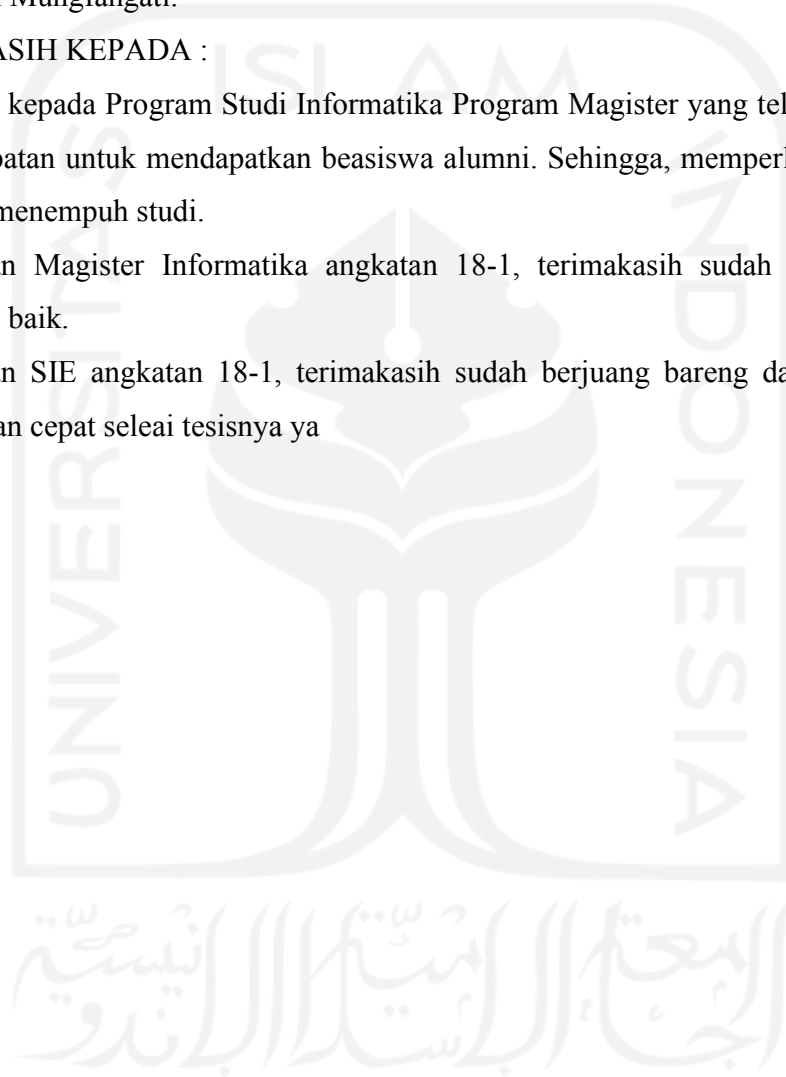
Ayah saya tercinta yang takhenti hentinya memberi dukungan Harun Nurudin dan Ibu saya tersayang Darsiti, Kakak – kakak yang selalu mendoakan saya saya Ika Trisna Apriliana dan Khusnul Mungfangati.

TERIMAKASIH KEPADA :

Terimakasih kepada Program Studi Informatika Program Magister yang telah memberikan saya kesmepatan untuk mendapatkan beasiswa alumni. Sehingga, memperlanja perjalanan saya dalam menempuh studi.

Teman-teman Magister Informatika angkatan 18-1, terimakasih sudah menjadi teman diskusi yang baik.

Teman-teman SIE angkatan 18-1, terimakasih sudah berjuang bareng dari awal kuliah, semoga kalian cepat selesai tesisnya ya



Kata Pengantar

Assalamu'alaikum Wr. Wb

Alhamdulillahirobbil'alamin,

Puji serta syukur senantiasa kita tujukan kehadiran Allah Subhanahu wa ta'ala, atas segala nikmat, karunia dan hidayah-Nya sehingga Tesis yang berjudul "*Context-Aware Natural Language Processing Services* menggunakan *Arsitektur Service Oriented Architecture*" ini dapat selesai dengan baik dan tepat waktu. Laporan Tesis yang telah disusun ini sebagai salah satu syarat untuk memperoleh gelar Master Strata-2 (S2) Informatika di Universitas Islam Indonesia.

Dalam pembuatan tugas akhir ini, peneliti menyadari bahwa telah ada banyak sekali dukungan, bimbingan, dorongan, inspirasi serta semangat dan doa yang selalu menyertai dalam pengerjaan tugas akhir ini. Sehingga pada kesempatan ini peneliti ingin mengucapkan terima kasih kepada:

1. Bapak Prof. Fathul Wahid, S.T., M.Sc., selaku Rektor Universitas Islam Indonesia
2. Bapak Hari Purnomo, Prof., Dr., Ir., M.T., selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia
3. Dr. Raden Teduh Dirgahayu, S.T., M.Sc. selaku Ketua Jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia dan dosen pembimbing dua saya yang telah memberikan banyak masukan dalam Tesis.
4. Irving Vitra Papatungan, S.T., M.Sc., Ph.D. selaku Ketua Program Studi Informatika Program Magister Fakultas Teknologi Industri Universitas Islam Indonesia.
5. Bapak Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D. selaku dosen pembimbing payung saya yang telah memberikan ilmu dan pemahaman kepada saya dalam menyelesaikan Tesis ini.
6. Segenap dosen Informatika UII yang telah mengajarkan banyak hal dan ilmu yang bermanfaat.
7. kedua Orang tua saya, Ayahanda Harun Nurudin dan Ibunda Darsiti yang sangat saya sayangi, yang sudah memberikan segalanya kepada saya, yang selalu memberikan dukungan dan doa setiap saat.
8. Kedua Kakak saya Ika Trisna Apriliana dan Khusnul Mungfangati yang juga selalu mendoakan setiap saat.

9. Serta semua orang-orang hebat yang ada di sekeliling saya, yang telah mendukung dan membantu dalam menyelesaikan tugas akhir ini.

Sebagai seorang yang masih perlu banyak belajar, peneliti menyadari bahwa masih ada banyak kekurangan dalam penyusunan tugas akhir ini. Oleh karena itu, peneliti mengharapkan saran dan kritik yang membangun untuk bekal di masa mendatang. Mohon maaf atas keterbatasan peneliti. Semoga apa yang telah dihasilkan dapat memberikan manfaat untuk semua pembaca.

Wassalamu'alaikum Wr. Wb.

Yogyakarta, 3 Januari 2023



Daftar Isi

| | |
|---|----------|
| Halaman Judul | i |
| Lembar Pengesahan Pembimbing | ii |
| Lembar Pengesahan Penguji..... | iii |
| Abstrak | iv |
| Abstract..... | v |
| Pernyataan Keaslian Tulisan | vi |
| Daftar Publikasi | vii |
| Halaman Kontribusi..... | viii |
| Halaman Persembahan | ix |
| Kata Pengantar..... | x |
| Daftar Isi | xii |
| Daftar Tabel..... | xvi |
| Daftar Gambar | xviii |
| Glosarium | xxi |
| BAB 1 Pendahuluan | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Rumusan Masalah..... | 3 |
| 1.3 Tujuan Penelitian | 3 |
| 1.4 Batasan Masalah | 3 |
| 1.5 Manfaat Penelitian | 3 |
| 1.6 Sistematika Penulisan | 3 |
| BAB 2 Tinjauan Pustaka | 5 |
| 2.1 Service Oriented Architecture | 5 |
| 2.2 Natural Language Processing (NLP) | 5 |
| 2.3 <i>Context-Aware</i> | 6 |

| | | |
|----------------------------------|---|----|
| 2.4 | Penelitian Terkait | 6 |
| BAB 3 Metodologi | | 8 |
| 3.1 | Pengumpulan Data | 8 |
| 3.2 | <i>Refactoring Source Code</i> | 8 |
| 3.3 | <i>Containerization</i> | 9 |
| 3.4 | Pemodelan <i>Context-Aware</i> | 9 |
| 3.5 | Rancangan Sistem | 10 |
| 3.5.1 | Perancangan Fungsionalitas | 11 |
| 3.5.2 | Perancangan Perilaku Sistem | 12 |
| 3.5.3 | Perancangan Antarmuka Sistem | 28 |
| 3.6 | Pengujian Sistem | 33 |
| 3.6.1 | Skenario Pengujian <i>Form Login</i> | 33 |
| 3.6.2 | Skenario Pengujian Halaman <i>Token</i> | 34 |
| 3.6.3 | Skenario Pengujian Aplikasi <i>Text Preprocessing</i> | 35 |
| 3.6.4 | Skenario Pengujian Aplikasi Filtering Konten Dewasa | 35 |
| 3.6.5 | Skenario Pengujian Aplikasi Sentimen Analisis | 36 |
| 3.6.6 | Skenario Pengujian Aplikasi K-means | 37 |
| 3.6.7 | Skenario Pengujian Aplikasi LSTM | 37 |
| 3.6.8 | Skenario Pengujian Aplikasi Naïve Bayes | 38 |
| BAB 4 Hasil dan Pembahasan | | 40 |
| 4.1 | Hasil Pengumpulan Data | 40 |
| 4.2 | <i>Hasil Refactoring Source Code</i> | 40 |
| 4.2.1 | <i>Adult content classification on Indonesian tweets using LSTM neural network</i> 41 | |
| 4.2.2 | Sentimen Analisis | 41 |
| 4.2.3 | <i>Text Preprocessing dan Long Short-Term Memory</i> | 42 |
| 4.2.4 | K-means | 42 |

| | | |
|----------------------------------|--|----|
| 4.3 | Hasil <i>Containerization</i> | 43 |
| 4.3.1 | <i>Containerization Backend Portal</i> | 44 |
| 4.3.2 | <i>Containerization Frontend Portal</i> | 45 |
| 4.3.3 | <i>Containerization Text Preprocessing</i> | 47 |
| 4.3.4 | <i>Containerization Filter Konten Dewasa</i> | 48 |
| 4.3.5 | <i>Containerization Sistem Analisis</i> | 48 |
| 4.3.6 | <i>Containerization LSTM</i> | 48 |
| 4.3.7 | <i>Containerization Naïve Bayes</i> | 48 |
| 4.3.8 | <i>Containerization K-means</i> | 49 |
| 4.3.9 | <i>Containerization Context-Aware</i> | 51 |
| 4.4 | Pemodelan Context-Aware | 51 |
| 4.5 | Hasil Implementasi Sistem | 53 |
| 4.5.1 | Akses <i>Portal</i> | 53 |
| 4.5.2 | Akses API | 58 |
| 4.5.3 | Service Oriented Architecture | 64 |
| 4.6 | Hasil Pengujian Sistem | 65 |
| 4.6.1 | Hasil Skenario Pengujian <i>Login</i> | 65 |
| 4.6.2 | Hasil Skenario Pengujian <i>Token</i> | 66 |
| 4.6.3 | Hasil Skenario Pengujian <i>Text Preprocessing</i> | 66 |
| 4.6.4 | Hasil Skenario Pengujian Filter Konten Dewasa | 67 |
| 4.6.5 | Hasil Skenario Pengujian Sentimen Analisis | 69 |
| 4.6.6 | Hasil Skenario Pengujian K-means | 70 |
| 4.6.7 | Hasil Skenario Pengujian LSTM | 71 |
| 4.6.8 | Hasil Skenario Pengujian Naïve Bayes | 72 |
| BAB 5 Kesimpulan dan Saran | | 73 |
| 5.1 | Kesimpulan | 73 |
| 5.2 | Saran | 74 |



Daftar Tabel

| | |
|---|----|
| Tabel 2.1 Penelitian Terkait SOA..... | 7 |
| Tabel 2.2 Penelitian Terkait Natural Language Processing | 7 |
| Tabel 3.1 Skenario Pengujian <i>Form Login</i> | 34 |
| Tabel 3.2 Skenario Pengujian Halaman <i>Token</i> | 34 |
| Tabel 3.3 Skenario Pengujian Aplikasi <i>Text Preprocessing (PORTAL)</i> | 35 |
| Tabel 3.4 Skenario Pengujian Aplikasi <i>Text Preprocessing (API)</i> | 35 |
| Tabel 3.5 Skenario Pengujian Aplikasi Filter Konten Dewasa (<i>Portal</i>)..... | 35 |
| Tabel 3.6 Skenario Pengujian Aplikasi Filtering Konten Dewasa (API)..... | 36 |
| Tabel 3.7 Skenario Pengujian Aplikasi Sentimen Analisis (<i>Portal</i>) | 36 |
| Tabel 3.8 Skenario Pengujian Aplikasi K-means (<i>Portal</i>) | 37 |
| Tabel 3.9 Skenario Pengujian Aplikasi LSTM (<i>Portal</i>) | 37 |
| Tabel 3.10 Skenario Pengujian Aplikasi LSTM (API)..... | 38 |
| Tabel 3.11 Skenario Pengujian Aplikasi Naïve Bayes (<i>Portal</i>) | 38 |
| Tabel 3.12 Skenario Pengujian Aplikasi Naïve Bayes (API)..... | 39 |
| Tabel 4.1 Daftar <i>Source Code</i> | 40 |
| Tabel 4.2 <i>Input</i> Parameter API <i>Text Preprocessing</i> | 59 |
| Tabel 4.3 Hasil Request API <i>Text Preprocessing</i> | 60 |
| Tabel 4.4 <i>Input</i> Parameter API Filter Konten Dewasa..... | 61 |
| Tabel 4.5 Hasil Request API Filter Konten Dewasa | 61 |
| Tabel 4.6 <i>Input</i> Parameter API Naïve Bayes..... | 62 |
| Tabel 4.7 Hasil <i>Request</i> API Naïve Bayes | 62 |
| Tabel 4.8 <i>Input</i> Parameter API LSTM Model..... | 63 |
| Tabel 4.9 Hasil <i>Request</i> API LSTM Model | 64 |
| Tabel 4.10 Hasil Pengujian Fitur <i>Login</i> | 65 |
| Tabel 4.11 Hasil Pengujian Fitur <i>Token</i> | 66 |
| Tabel 4.12 Hasil Pengujian Fitur <i>Text Preprocessing</i> | 66 |
| Tabel 4.13 Hasil Pengujian Fitur API <i>Text Preprocessing</i> | 67 |
| Tabel 4.14 Hasil Pengujian Filter Konten Dewasa..... | 68 |
| Tabel 4.15 Hasil Pengujian API Filter Konten Dewasa | 68 |
| Tabel 4.16 Hasil Pengujian Sentimen Analisis | 69 |
| Tabel 4.17 Hasil Pengujian K-means | 70 |
| Tabel 4.18 Hasil Pengujian LSTM..... | 71 |

Tabel 4.19 Hasil Pengujian API LSTM..... 71
Tabel 4.20 Hasil Pengujian Fitur Naïve Bayes..... 72
Tabel 4.21 Hasil Pengujian Fitur API Naïve Bayes 72



Daftar Gambar

| | |
|---|----|
| Gambar 3.1 Langkah – Langkah Penelitian | 8 |
| Gambar 3.2 Langkah-langkah <i>Containerization</i> | 9 |
| Gambar 3.3 Contoh File <i>Input</i> | 10 |
| Gambar 3.4 Proses <i>Context-Aware</i> | 10 |
| Gambar 3.5 Diagram <i>Use Case NLP Service</i> | 11 |
| Gambar 3.6 Diagram Aktivitas Daftar | 13 |
| Gambar 3.7 Diagram Aktivitas Login | 14 |
| Gambar 3.8 Diagram Aktivitas Buat <i>Token</i> | 15 |
| Gambar 3.9 Diagram Aktivitas <i>Text Preprocessing</i> | 16 |
| Gambar 3.10 Diagram Aktivitas <i>Text Preprocessing API</i> | 17 |
| Gambar 3.11 Diagram Aktivitas Filter Konten Dewasa..... | 18 |
| Gambar 3.12 Diagram Aktivitas Filter Konten Dewasa API..... | 18 |
| Gambar 3.13 Sentimen Analisis | 19 |
| Gambar 3.14 Diagram Aktivitas Sentimen Analisis API..... | 20 |
| Gambar 3.15 Diagram Aktivitas K-means | 21 |
| Gambar 3.16 Diagram Aktivitas K-means API..... | 22 |
| Gambar 3.17 Diagram Aktivitas LSTM API..... | 23 |
| Gambar 3.18 Diagram Aktivitas LSTM..... | 24 |
| Gambar 3.19 Diagram Aktivitas Naïve Bayes | 25 |
| Gambar 3.20 Diagram Aktivitas Naïve Bayes API..... | 26 |
| Gambar 3.21 Diagram Aktivitas <i>Context-Aware</i> | 27 |
| Gambar 3.22 Diagram Aktivitas <i>Manage User</i> | 28 |
| Gambar 3.23 Rancangan Antarmuka Daftar | 29 |
| Gambar 3.24 Rancangan Antarmuka Login..... | 29 |
| Gambar 3.25 Rancangan Antarmuka Buat <i>Token</i> | 30 |
| Gambar 3.26 Rancangan Antarmuka <i>Context-Aware</i> | 30 |
| Gambar 3.27 Rancangan Antarmuka <i>Text Preprocessing</i> | 31 |
| Gambar 3.28 Rancangan antarmuka Filter Konten Dewasa..... | 31 |
| Gambar 3.29 Rancangan Antarmuka Sentimen Analisis | 32 |
| Gambar 3.30 Rancangan Antarmuka K-means | 32 |
| Gambar 3.31 Rancangan Antarmuka LSTM..... | 33 |
| Gambar 3.32 Rancangan Antarmuka Naïve Bayes | 33 |

| | |
|--|----|
| Gambar 4.1 <i>Recode koneksi database</i> | 41 |
| Gambar 4.2 <i>Recode Adult Content Classification On Indonesian Tweets Using Lstm Neural Network</i> | 41 |
| Gambar 4.3 <i>Source Code K-means</i> | 42 |
| Gambar 4.4 <i>Kode Algoritma Sentimen Analisis</i> | 43 |
| Gambar 4.5 <i>Daftar Container</i> | 44 |
| Gambar 4.6 <i>Docker-compose Backend Portal</i> | 45 |
| Gambar 4.7 <i>Dockerfile Frontend Portal</i> | 46 |
| Gambar 4.8 <i>Docker-compose Frontend Portal</i> | 46 |
| Gambar 4.9 <i>Dockerfile Text Preprocessing</i> | 47 |
| Gambar 4.10 <i>Docker-Compose Text Preprocessing</i> | 47 |
| Gambar 4.11 <i>Docker-compose Filter Konten Dewasa</i> | 48 |
| Gambar 4.12 <i>Docker-compose Sistem Analisis</i> | 49 |
| Gambar 4.13 <i>Docker-compose LSTM</i> | 49 |
| Gambar 4.14 <i>Docker-compose Naïve Bayes</i> | 50 |
| Gambar 4.15 <i>Docker-compose K-means</i> | 50 |
| Gambar 4.16 <i>Docker-compose Context-Aware</i> | 51 |
| Gambar 4.17 <i>Contoh Dataset 1</i> | 51 |
| Gambar 4.18 <i>Contoh Dataset 2</i> | 52 |
| Gambar 4.19 <i>Hasil Context-Aware</i> | 52 |
| Gambar 4.20 <i>Hasil Analisis Algoritma Supervised</i> | 53 |
| Gambar 4.21 <i>Halaman Daftar</i> | 54 |
| Gambar 4.22 <i>Halaman Login</i> | 54 |
| Gambar 4.23 <i>Halaman Token</i> | 55 |
| Gambar 4.24 <i>Halaman Text preprocessing</i> | 55 |
| Gambar 4.25 <i>Halaman Filter Konten Dewasa</i> | 56 |
| Gambar 4.26 <i>Halaman Sentimen Analisis</i> | 56 |
| Gambar 4.27 <i>Halaman K-means</i> | 57 |
| Gambar 4.28 <i>Gambar Halaman LSTM</i> | 57 |
| Gambar 4.29 <i>Halaman Naïve Bayes</i> | 58 |
| Gambar 4.30 <i>Akses API Aplikasi Text Preprocessing</i> | 60 |
| Gambar 4.31 <i>Akses API Aplikasi Filter Konten Dewasa</i> | 61 |
| Gambar 4.32 <i>Akses API aplikasi Naïve Bayes</i> | 62 |

Gambar 4.33 Akses API aplikasi LSTM Model 63
Gambar 4.34 Akses API aplikasi LSTM..... 64



Glosarium

| | |
|------|-------------------------------------|
| NLP | - Natural Language Processing |
| AI | - Artificial Intelligence |
| UI | - User Interface |
| LSTM | - Long Short Term Memory Network |
| API | - Application Programming Interface |
| SOA | - Service Oriented Architecture |



BAB 1

Pendahuluan

1.1 Latar Belakang

Service oriented architecture (SOA) hadir sebagai kerangka desain untuk mewujudkan pengembangan sistem yang cepat dan murah, sistem yang mudah dimodifikasi (Komoda, n.d.). SOA menyediakan satu set aturan dasar untuk mendefinisikan konsep dan mengelola perubahan pengembangan sistem dan integrasi (Ouda et al., 2010). SOA pada dasarnya adalah kumpulan layanan yang memberikan fungsionalitas yang dapat digunakan dan digunakan kembali saat membangun atau mengintegrasikan aplikasi (Parveen & Tilley, 2008).

(Sudjana et al., 2019) memilih SOA karena memudahkan saat melakukan pembaruan untuk meningkatkan aplikasi jalan.com. (Muhardany et al., 2020) berhasil mengembangkan *e-commerce* menggunakan SOA dengan hasil waktu *loading* data menjadi lebih cepat. (Papulovskaya et al., 2019) menyimpulkan bahwa SOA dapat memberikan skalabilitas, fleksibilitas, kompatibilitas, ketahanan dan keamanan di semua tingkat aplikasi IoT.

Natural Language Processing adalah cabang dari ilmu Artificial Intelligence (AI) supaya komputer bisa memahami bahasa alami (Trofin et al., 2019). *Natural Language Processing* semakin dibutuhkan seiring dengan berkembangnya data teks setiap harinya (Lee, 2019). Berkembangnya data teks membuat penelitian dengan metode *Natural Language Processing* semakin dibutuhkan, namun untuk melakukan penelitian menggunakan *Natural Language Processing* harus bisa pemrograman tentu ini menjadi hambatan tersendiri terutama bagi yang tidak memiliki latar belakang teknologi informasi.

Dalam hal ini sebenarnya banyak perusahaan sudah menyediakan layanan *Natural Language Processing service*. Seperti CloudFactory dan prosa.ai, namun layanan yang ditawarkan tidak melihatkan algoritma yang digunakan atau tidak bisa mengubah algoritma yang digunakan.

(Siddiqui & Babar, 2020) mengusulkan *Natural Language Processing* untuk sistem yang mengklasifikasikan opini tentang produk dan layanan dari perspektif dan tweet konsumen untuk rekomendasi. Sistem ini juga berguna untuk semua produk dan layanan sehingga memudahkan pelanggan. (Eryiğit, 2015) mengembangkan ITU Turkish NLP *Web*

Service, sistem yang menyediakan alat penelitian yang canggih, disebutkan juga karena memberikan kemudahan dalam melakukan pemrosesan bahasa alami, sistem ini mampu menarik banyak pengguna dari berbagai Universitas di Turki.

Context-Aware adalah sistem yang menggunakan konteks untuk memberikan informasi atau layanan yang relevan kepada pengguna (Perera et al., 2014). *Context-Aware* bisa diterapkan pada sistem cerdas atau aplikasi pintar. *Context-Aware* membuat aplikasi mampu untuk merasakan dan bereaksi terhadap lingkungannya. Dengan teknologi ini memungkinkan system untuk beradaptasi dan beroperasi pada aturan-aturan yang telah ditetapkan. Hal ini membuat system dapat bekerja sesuai keinginan pengguna tanpa melakukan perubahan aturan (Perdana & Ashari, 2019).

Context-Aware diperlukan untuk mengatasi masalah banyaknya metode *Natural Language Processing*, *Context-Aware* membantu memilih metode *Natural Language Processing* yang tepat. *Context-Aware* didefinisikan sebagai kemampuan sistem atau komponen sistem untuk mengumpulkan informasi tentang lingkungannya pada waktu tertentu dan menyesuaikan perilaku yang sesuai. *Context-Aware* menggunakan perangkat lunak dan perangkat keras untuk secara otomatis mengumpulkan dan menganalisis data untuk memandu tanggapan aplikasi (Wigmore, 2016).

Dengan *Context-Aware* sistem mampu membaca file yang diinput oleh pengguna dan bisa mengerti jenis data dari file tersebut apakah termasuk *categorical* jika file mempunyai data unik kurang atau sama dengan tiga (≤ 3) atau *uncategorical* jika file mempunyai data unik lebih dari tiga (> 3).

(Kousiouris et al., 2019) mengembangkan sistem untuk mendeteksi percakapan penting pada telepon seluler, disebutkan juga sistem mampu menunjukkan informasi tentang konteks percakapan dengan menggunakan pendekatan berbasis konten murni. (Voita et al., 2018) mengembangkan sistem *neural machine translation* (NMT), merupakan mesin penerjemah yang dilengkapi *Context-Aware* sehingga NMT mampu mendeteksi kata ganti yang ambigu.

Menurut (Flores-Rodriguez & Aguilar-Alonso, 2020) SOA sangat membantu untuk pengembangan perangkat lunak dengan lebih efisien dan lincah, terlebih lagi untuk perusahaan yang memiliki layanan cloud. Maka pada penelitian ini akan melakukan implementasi SOA untuk sistem *Natural language processing services* diharapkan dapat memudahkan pengembangan sistem. Belum ada penelitian yang melakukan implementasi *Context-Aware* dan SOA dengan studi kasus *Natural Language Processing Services* maka

kontribusi dari penelitian ini adalah fitur *Context-Aware* untuk memberikan rekomendasi pengolahan data menggunakan algoritma *clustering* atau *classification*.

1.2 Rumusan Masalah

Dengan kelebihan yang diberikan oleh SOA dan semakin dibutuhkannya *Natural Language Processing*, terdapat pertanyaan penelitian yaitu bagaimana merancang, membangun, mengimplementasi dan mengevaluasi SOA untuk *Natural Language Processing Service* dengan ditambahkan fitur *Context-Aware*.

1.3 Tujuan Penelitian

Tujuan dari penelitian adalah untuk merancang, membangun dan mengimplementasikan SOA untuk *Natural Language Processing Service* dengan ditambahkan fitur *Context-Aware*.

1.4 Batasan Masalah

Di dalam melaksanakan kegiatan penelitian ini ada beberapa batasan masalah, yaitu:

1. Penelitian ini hanya mengintegrasikan aplikasi-aplikasi yang menggunakan Bahasa pemrograman Python.
2. Penelitian ini hanya menggunakan keamanan komunikasi secara default dari *framework* yang digunakan.
3. Pengguna hanya mengakses fitur saja tidak bisa mengakses *source code* yang ada. ataupun debugging pada aplikasi-aplikasi yang ada.
4. *Context-Aware* untuk mengenali data untuk diolah dengan algoritma *clustering* atau algoritma *classification*.

1.5 Manfaat Penelitian

Di dalam melaksanakan penelitian ini terdapat beberapa manfaat yaitu :

1. Untuk ilmu pengetahuan bidang informatika, *Natural Language Processing Service* dapat memudahkan pengolahan data *text*.
2. Untuk Peneliti tentang *natural language processing*, sistem ini dapat memudahkan karena aplikasinya bisa langsung digunakan tanpa harus melakukan pemrograman.

1.6 Sistematika Penulisan

Sistematika penulisan penelitian disusun sebagai berikut :

Bab 1 Pendahuluan

Pada bab ini membahas tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan dalam penulisan penelitian.

Bab 2 Tinjauan Pustaka

Pada bab ini membahas tentang literatur mengenai SOA, *natural language processing* dan *Context-Aware* dengan beberapa sumber pencarian data dan jurnal. Penelitian terdahulu yang terkait dengan penelitian yang dilakukan dan beberapa landasan teori yang menjadi rujukan pada penelitian.

Bab 3 Metodologi Penelitian

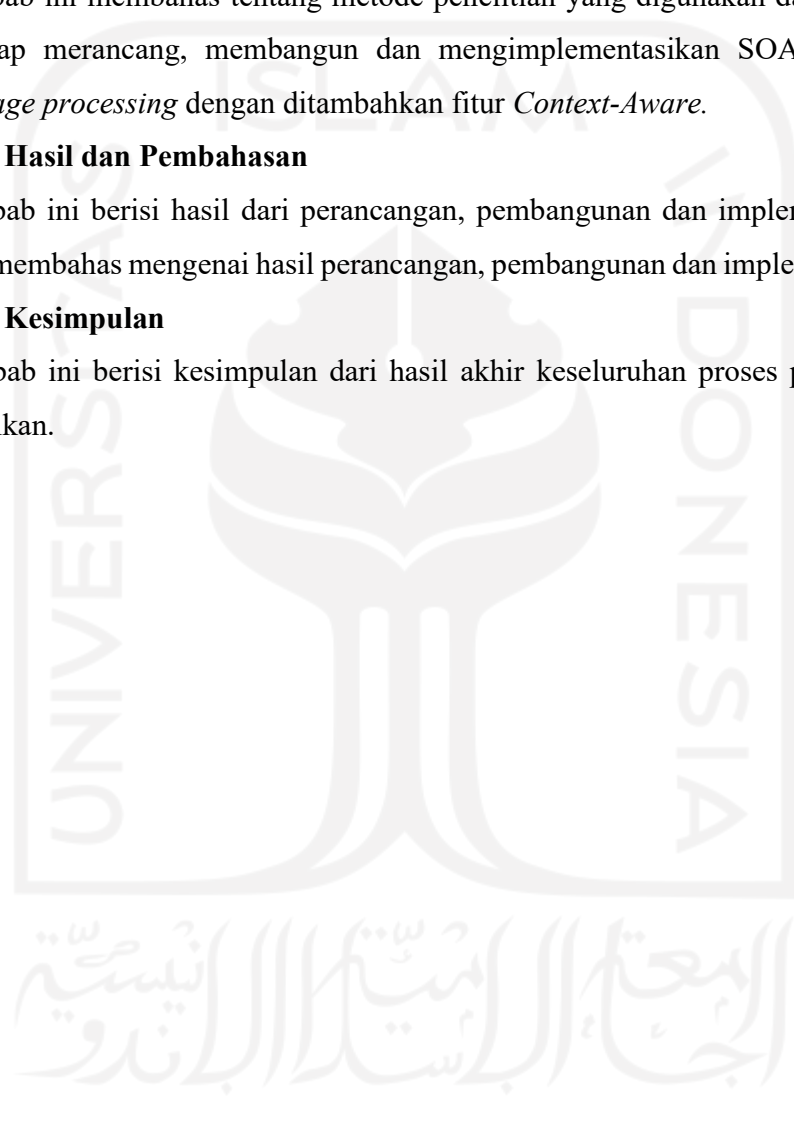
Pada bab ini membahas tentang metode penelitian yang digunakan dalam penelitian terhadap merancang, membangun dan mengimplementasikan SOA pada *natural language processing* dengan ditambahkan fitur *Context-Aware*.

Bab 4 Hasil dan Pembahasan

Pada bab ini berisi hasil dari perancangan, pembangunan dan implementasi sistem. Serta membahas mengenai hasil perancangan, pembangunan dan implementasi sistem.

Bab 5 Kesimpulan

Pada bab ini berisi kesimpulan dari hasil akhir keseluruhan proses penelitian yang dihasilkan.



BAB 2

Tinjauan Pustaka

Penelitian ini dimulai dengan studi literatur mengenai Service oriented architecture (SOA), *natural language processing* dan *Context-Aware*. Didukung dengan beberapa sumber pencarian data dan jurnal, penelitian terdahulu terkait dengan penelitian yang dilakukan dan beberapa landasan teori yang menjadi rujukan pada penelitian adalah sebagai berikut.

2.1 Service Oriented Architecture

SOA dirancang untuk memungkinkan pengembang dapat mengatasi banyak tantangan yang didistribusikan komputasi perusahaan, termasuk integrasi aplikasi, transaksi kebijakan manajemen dan keamanan, memungkinkan berbagai platform, protokol, dan meningkatkan akses ke banyak perangkat (Monteiro & Silva, 2015).

(Sudjana et al., 2019) menggunakan SOA sebagai arsitektur dari aplikasi jalan.com karena memudahkan saat melakukan pembaruan aplikasi. (Muhardany et al., 2020) berhasil waktu *loading* data menjadi lebih cepat pada *e-commerce* dengan menggunakan SOA. (Papulovskaya et al., 2019) menyebutkan bahwa dengan SOA bisa memberikan skalabilitas, fleksibilitas, kompatibilitas, ketahanan dan keamanan.

2.2 Natural Language Processing (NLP)

Natural language processing (NLP) merupakan metode yang semakin dibutuhkan seiring dengan berkembangnya data teks setiap harinya (Lee, 2019). konten utama penelitian teknologi NLP adalah analisis semantik, analisis tata bahasa, dan pemahaman teks (Wang et al., 2019). Secara umum relasi analisis semantik dipahami sebagai relasi dari konsep dalam sistem konseptual domain (Osipov et al., 2012). Analisis tata bahasa adalah komponen utama dari banyak aplikasi NLP. termasuk pemeriksa tata bahasa, berbasis aturan terjemahan mesin, dan pengenalan suara sistem (Santaholma, 2007). Pemahaman teks adalah NLP bisa menggunakan kecerdasan buatan untuk mengambil input, memprosesnya, dan memahaminya dengan cara yang dapat dipahami komputer.

(Siddiqui & Babar, 2020) Menyimpulkan NLP bisa diterapkan untuk sistem klasifikasi opini tentang produk dan layanan dari tweet konsumen untuk dijadikan rekomendasi, sistem ini dapat memudahkan pelanggan. (Eryiğit, 2015) mengambangkan ITU *Turkish NLP Web Service*, sistem ini mampu menarik banyak pengguna karena memberikan kemudahan dalam melakukan pemrosesan bahasa alami.

Menggunakan SOA penelitian ini akan membangun *natural language processing service* dengan algoritma–algoritma *natural language processing* sebagai *service*-nya.

2.3 *Context-Aware*

Context aware adalah kemampuan untuk memahami situasi pengguna dalam segala kondisi dan beradaptasi dengan semua konsekuensi dari perilaku sistem (Dewabharata et al., 2013). Teknologi ini membuat sistem mampu untuk mendeteksi dan bereaksi terhadap lingkungannya. Dengan teknologi ini memungkinkan sistem untuk beradaptasi dan beroperasi pada aturan-aturan yang telah ditetapkan. Hal ini membuat sistem bekerja sesuai dengan keinginan pengguna tanpa mengubah aturan (Perdana & Ashari, 2019).

Sistem bisa merasakan lingkungannya dengan menggunakan sensor atau algoritma tertentu. Contohnya pada Aplikasi Gmaps pada *handphone* sistem bisa menentukan posisi dengan menggunakan sensor GPS, Ketika pengguna membuka aplikasi, sistem bisa menunjukkan titik lokasi secara langsung. Sedangkan untuk sistem bisa bereaksi dengan lingkungannya seperti pada fitur *directions* pada Gmaps, di mana sistem bisa memberitahukan jalan tercepat ke tujuan dengan menghindari kemacetan/hambatan yang ada di jalan.

Context-Aware pada penelitian ini adalah sistem mampu mengenali file *input* (xlsx/csv) dengan menentukan label dan fitur pada file tersebut dan memberikan kategori pada file termasuk *supervised* atau *unsupervised*. Reaksi dari sistem adalah menentukan algoritma yang paling tepat digunakan untuk file tersebut. Jika kategori file adalah *unsupervised* maka sistem akan memberikan daftar algoritma sementara jika algoritma *supervised* selain sistem akan memberikan daftar algoritma sistem juga akan menampilkan hasil dari komputasi setiap algoritma *supervised*.

2.4 Penelitian Terkait

Kontribusi pada penelitian ini adalah implementasi *Context-Aware* dan SOA untuk *Natural Language Processing Services*. Menjelaskan kontribusi di atas dan posisi dari rencana penelitian ini, *state of the art* penelitian sebelumnya adalah belum ada penelitian yang melakukan implementasi *Context-Aware* dan SOA dengan studi kasus *Natural Language Processing Services*.

Context-Aware pada penelitian ini adalah sistem mampu mengkategorikan file *input* dalam kategori *supervised* atau *unsupervised* dengan membaca jumlah data unik pada setiap *file*. Ulasan dari penelitian tentang SOA di jelaskan pada Tabel 2.1 dan ulasan penelitian tentang NLP pada Tabel 2.2

Tabel 2.1 Penelitian Terkait SOA

| No. | Keywords | Ulasan Kritis | Pustaka |
|-----|---|---|-----------------------------|
| 1 | <i>Transportation, One Stop Travel Solution, Travel Information System, Service Oriented Architecture</i> | Dalam penelitian ini disimpulkan bahwa SOA memudahkan saat melakukan pembaruan dan perbaikan aplikasi jalan.com. | (Sudjana et al., 2019) |
| 2 | <i>E-Commerce, Web Services, SOMA, Service Oriented Architecture</i> | Dalam penelitian ini disebutkan bahwa berhasil mengembangkan e-commerce menggunakan SOA dengan hasil waktu loading data menjadi lebih cepat. | (Muhardany et al., 2020) |
| 3 | <i>scalability, microcontrollers, service-oriented systems engineering, message-oriented middleware, client-server systems, Internet of Things.</i> | Dalam penelitian ini disebutkan SOA dapat memberikan skalabilitas, fleksibilitas, kompatibilitas, ketahanan dan keamanan di semua tingkat aplikasi IoT. | (Papulovskaya et al., 2019) |

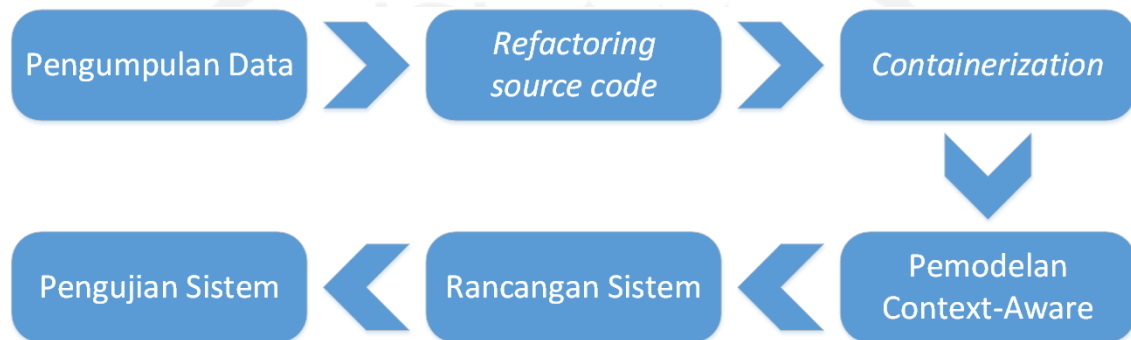
Tabel 2.2 Penelitian Terkait Natural Language Processing

| No. | Keywords | Ulasan Kritis | Pustaka |
|-----|--|--|--------------------------|
| 1 | NLP | (Eryigit, 2015) mengambangkan ITU <i>Turkish NLP Web Service</i> , sistem yang canggih, karena memberikan kemudahan dalam melakukan pemrosesan bahasa alami. | (Eryigit, 2015) |
| 2 | NLTK, AlchemyAPI, recommendation system, Tweepy. | (Siddiqui & Babar, 2020) mengembangkan sistem klasifikasi opini tentang produk dan layanan dari perspektif dan cuitan konsumen untuk dijadikan rekomendasi. | (Siddiqui & Babar, 2020) |

BAB 3

Metodologi

Penulis melewati beberapa tahapan dalam melakukan penelitian ini. Tahapan penelitian tersebut adalah pengumpulan data, *refactoring source code*, *Containerization*, pemodelan *Context-Aware* dan Pengujian Sistem. Gambaran tahapan penelitian ini bisa dilihat pada Gambar 3.1.



Gambar 3.1 Langkah – Langkah Penelitian

3.1 Pengumpulan Data

Data yang digunakan sebagai dataset diambil dari <https://www.kaggle.com/datasets> data yang diambil hanya data .csv, xls atau xlsx. Kaggle dipilih karena menyediakan banyak dataset secara gratis (Quaranta et al., 2021).

Source code yang digunakan adalah *source code* tentang *Natural Language Processing* yang ditulis menggunakan bahasa pemrograman Python. Python dipilih karena merupakan solusi pemrograman lengkap, bahasa yang mudah dipelajari, fleksibilitasnya juga membuat Python menjadi bahasa yang cepat berkembang dan banyak tersedia *library* (Nagpal & Gabrani, 2019). Terutama *library data analytics* yang membuat Python merupakan bahasa yang direkomendasikan digunakan untuk analisis data (Nagpal & Gabrani, 2019). Python juga lebih sederhana jika dibandingkan bahasa pemrograman lain seperti Ruby, Perl, Java, C++, dll (Prashant et al., 2019).

3.2 Refactoring Source Code

Refactoring diperlukan untuk menyesuaikan *source code* dengan *Natural Language Processing Services* sehingga *source code* bisa berjalan dengan baik. *Refactoring* dilakukan dengan mengambil kode algoritmanya, serta melakukan penyesuaian *input* dan *output*. Kemudian kode algoritma tersebut diterapkan pada Django *Rest Framework*. Django dipilih

karena terbukti *powerful* dalam penggunaan kembali kode dan ekstensibilitas fungsional dan perkembangan yang cepat (Yu & Yang, 2019).

3.3 *Containerization*

Containerization telah mencapai popularitasnya saat ini karena diadopsi oleh SOA sebagai dasar untuk membangun aplikasi (van den Berg et al., 2017). Arsitektur berbasis SOA adalah salah satu fungsi bisnis yang disediakan melalui sejumlah layanan yang saling terhubung. Dengan *Containerization*, aplikasi dapat dikemas ke dalam wadah yang dapat dengan mudah digunakan pada banyak layanan yang berbeda (van den Berg et al., 2017).

Docker adalah platform *containerization* yang sangat populer. *Containerization* adalah proses mengisolasi aplikasi menjadi *container* sehingga aplikasi menjadi independent (Muddinagiri et al., 2020). *Container* adalah tempat aplikasi dan semua dependensinya, library, dan binari lainnya dan file konfigurasi yang diperlukan untuk menjalankannya, dibundel menjadi satu paket (Modak et al., 2018). Dockerfile berfungsi untuk mendefinisikan root dari sistem file *image* kontainer, berbagai informasi seperti *environment variables*, perubahan yang akan dilakukan pada *image base*, penambahan beberapa aplikasi dan *port* serta menentukan perintah yang harus dijalankan setelah kontainer dimulai (Santoro et al., 2018). *Containerization* dapat membantu mempercepat *restart* setelah pengembangan atau pemulihan aplikasi (Abdollahi Vayghan et al., 2018).

Containerization diperlukan agar *source code* bisa berjalan di *docker container*. *Containerization* dilakukan dengan tiga langkah. Langkah – langkah tersebut seperti pada Gambar 3.2. Langkah yang pertama adalah pembuatan Dockerfile/docker-compose. Langkah yang kedua adalah *build* *docker image* dan langkah yang ketiga adalah menjalankan *container*.



Gambar 3.2 Langkah-langkah *Containerization*

3.4 *Pemodelan Context-Aware*

Untuk memberikan rekomendasi analisis, sistem membaca file excel atau csv yang diunggah pengguna. Jika jumlah kolom pada file lebih dari dua, maka kolom dengan jumlah data unik paling sedikit akan digunakan sebagai label dan kolom yang mempunyai jumlah unik paling banyak akan digunakan sebagai fitur. Jika jumlah data unik pada fitur kurang atau sama dengan tiga (≤ 3) maka file tersebut dikategorikan *categorical* dan jika lebih dari tiga maka dikategorikan *uncategorical*.

Kondisi tersebut dipilih karena semakin sedikit data unik pada fitur, semakin mudah menentukan kategori pada setiap baris datanya. Jika file input seperti Gambar 3.3 maka sistem akan mengenali file tersebut sebagai file *categorical* karena jumlah data unik pada kolom fitur adalah tiga yaitu 1,2 dan 6. Setelah kategori file diketahui, sistem memberikan rekomendasi algoritma yang sesuai dengan kategorinya Misalnya jika ketegori *categorical* sistem merekomendasikan untuk menggunakan algoritma *naive bayes*.

| country | child_mor | exports | health | imports | income | inflation | life_exp | total_fer | gdp |
|------------------|-----------|---------|--------|---------|--------|-----------|----------|-----------|-------|
| Alghaniastan | 30.2 | 10 | 7.38 | 44.7 | 3300 | 22.4 | 60.1 | 2 | 333 |
| Albania | 16.6 | 28 | 6.55 | 48.1 | 19100 | 1.44 | 76.8 | 1 | 4090 |
| Algeria | 27.3 | 38.4 | 4.17 | 31.1 | 18700 | 20.9 | 75.8 | 2 | 4460 |
| Angola | 119 | 62.3 | 2.85 | 42.5 | 3300 | 22.4 | 60.1 | 6 | 3530 |
| Antigua and Barb | 10.3 | 45.5 | 6.03 | 58.9 | 19100 | 1.44 | 76.8 | 2 | 12200 |
| Argentina | 3.1 | 16 | 18700 | 20.9 | 75.8 | 2 | 10300 | | |
| Armenia | 4.4 | 45.3 | 73.8 | 1 | 3220 | | | | |
| Australia | 7.3 | 20.9 | 82.1 | 1 | 51900 | | | | |
| Austria | 4.3 | 51.3 | 11 | 47.8 | 80.6 | 1 | 46900 | | |
| Azerbaijan | 39.2 | 54.3 | 5.88 | 20.7 | 16000 | 13.8 | 69.1 | 1 | 5840 |
| Bahamas | 13.8 | 35 | 7.89 | 43.7 | 22900 | -0.393 | 73.8 | 1 | 28000 |
| Bahrain | 8.6 | 69.5 | 4.97 | 50.9 | 41100 | 7.44 | 76.7 | 2 | 20700 |
| Bangladesh | 49.4 | 16 | 3.52 | 21.8 | 2440 | 7.14 | 70.4 | 2 | 758 |
| Barbados | 14.2 | 39.5 | 7.97 | 48.7 | 15300 | 0.321 | 76.7 | 1 | 16000 |
| Belarus | 5.5 | 51.4 | 5.61 | 64.5 | 16200 | 15.1 | 70.4 | 1 | 6030 |
| Belgium | 4.5 | 76.4 | 10.7 | 74.7 | 41100 | 1.88 | 80.1 | 1 | 44400 |
| Belize | 18.8 | 58.2 | 5.2 | 57.5 | 7880 | 1.14 | 71.1 | 2 | 4340 |
| Benin | 111 | 23.8 | 4.1 | 37.2 | 1820 | 0.885 | 61.9 | 2 | 758 |
| Bhutan | 42.7 | 42.5 | 5.2 | 70.7 | 6420 | 5.99 | 72.1 | 2 | 2180 |

Gambar 3.3 Contoh File Input

Gambar 3.4 merupakan alur proses *Context-Aware* yang terdiri 3 tahap. Pertama pengguna mengunggah file csv/xlsx, kemudian aplikasi mengubah file menjadi JSON dan dikirim ke aplikasi *backend Context-Aware*. JSON dipilih karena format pertukaran data yang ringan, mudah dibaca dan ditulis, ini berbanding terbalik dengan XML (Lin et al., 2012). Kedua aplikasi *Context-Aware* menentukan jenis file tersebut termasuk *categorical* atau *uncategorical*. Ketiga aplikasi *Context-Aware* menampilkan jenis file apakah *categorical* dan *uncategorical*.



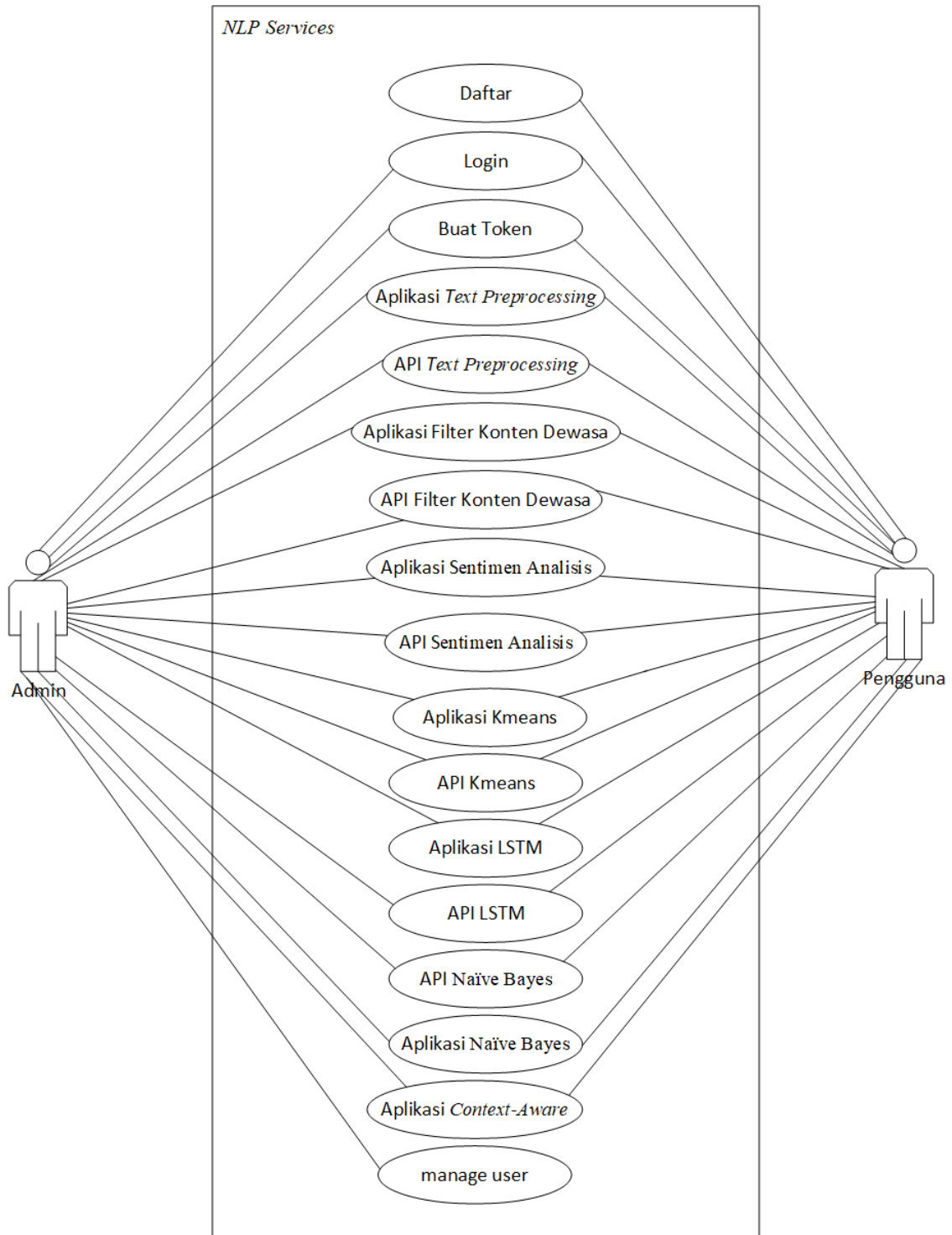
Gambar 3.4 Proses *Context-Aware*

3.5 Rancangan Sistem

Rancangan sistem bertujuan untuk menganalisis kebutuhan yang ada pada sistem yang meliputi perangkat lunak (*software*), pengguna (*user*) serta hasil analisis terhadap sistem.

3.5.1 Perancangan Fungsionalitas

Perancangan fungsionalitas pada penelitian ini menggunakan diagram *use case*. Diagram *use case* menggambarkan fungsionalitas sistem. *Use case* merepresentasikan sebuah interaksi antar aktor (manusia) dengan sistem. Gambar 3.5 adalah *use case natural language processing services* menggambarkan dua actor admin dan pengguna.



Gambar 3.5 Diagram *Use Case* NLP Service

Dari *use case* yang sudah dimodelkan dapat dilihat proses yang bisa dilakukan admin dan pengguna. Admin dapat melakukan *login*, buat *token*, Akses API, Akses aplikasi dan manajemen user sedangkan pengguna bisa daftar, *login*, buat *token*, Akses API, Akses aplikasi.

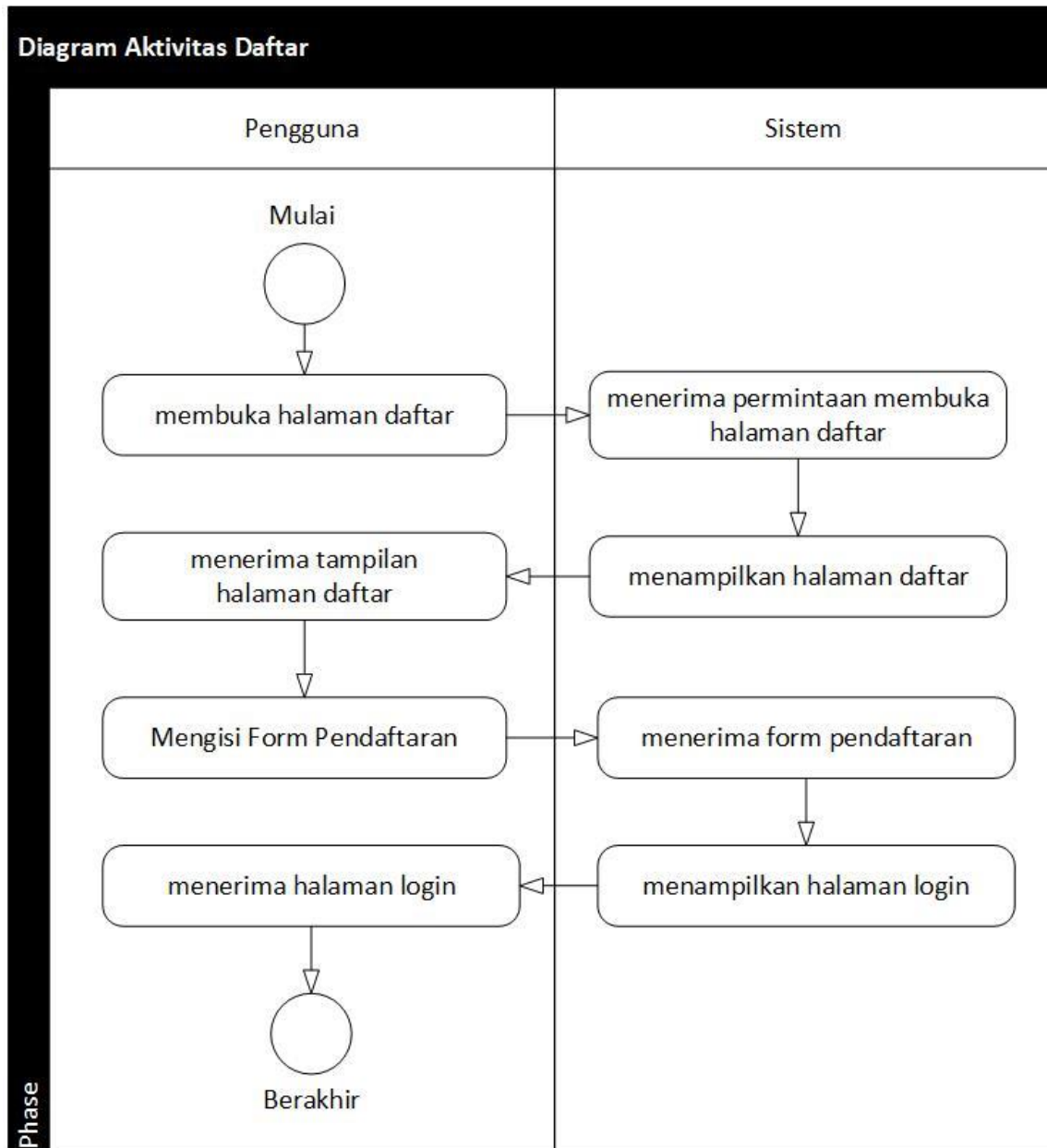
Akses API adalah pengguna mengakses langsung *endpoint* API NLP *service* tanpa melalui aplikasi *frontend* NLP *service*. Akses Aplikasi adalah pengguna mengakses NLP *service* melalui aplikasi *frontend* NLP *service*

3.5.2 Perancangan Perilaku Sistem

Diagram aktivitas (*activity diagram*) digunakan untuk merancang perilaku sistem. Diagram aktivitas merupakan diagram yang menggambarkan seluruh aktivitas dalam sebuah *use case*. Berikut merupakan diagram aktivitas dari *natural language processing services* :

Diagram Aktivitas Daftar

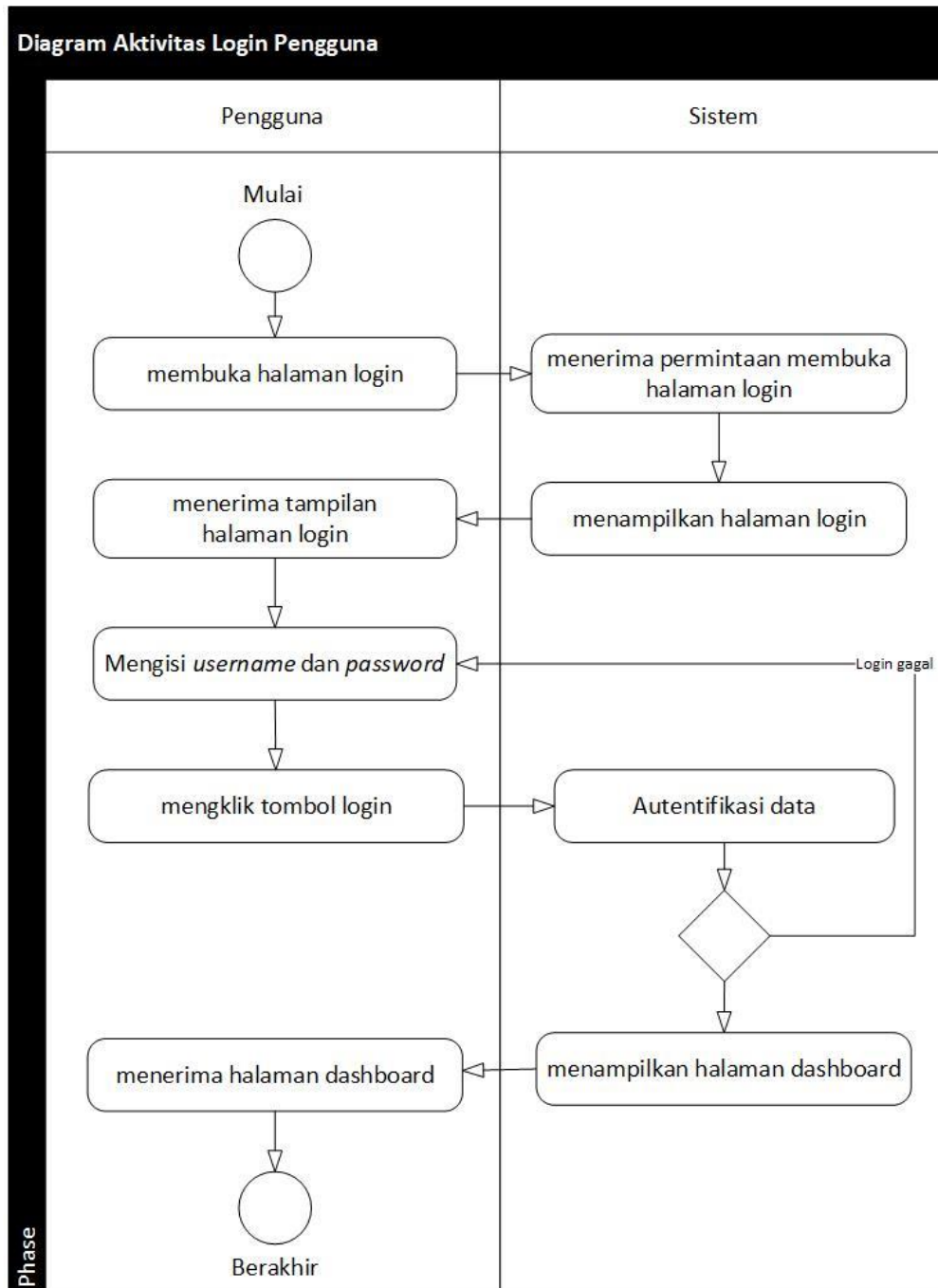
Diagram ini menggambarkan aktivitas pengguna untuk mendaftar ke *natural language processing services*. Gambar 3.6 adalah Skenario dari aktivitas daftar, dimulai dari pengguna membuka atau mengakses url *natural language processing services*, kemudian mengklik daftar. Pengguna harus mengisi form pendaftaran dengan isian NIM, Nama, email, Kata sandi dan Ulang kata sandi.



Gambar 3.6 Diagram Aktivitas Daftar

Diagram Aktivitas Login

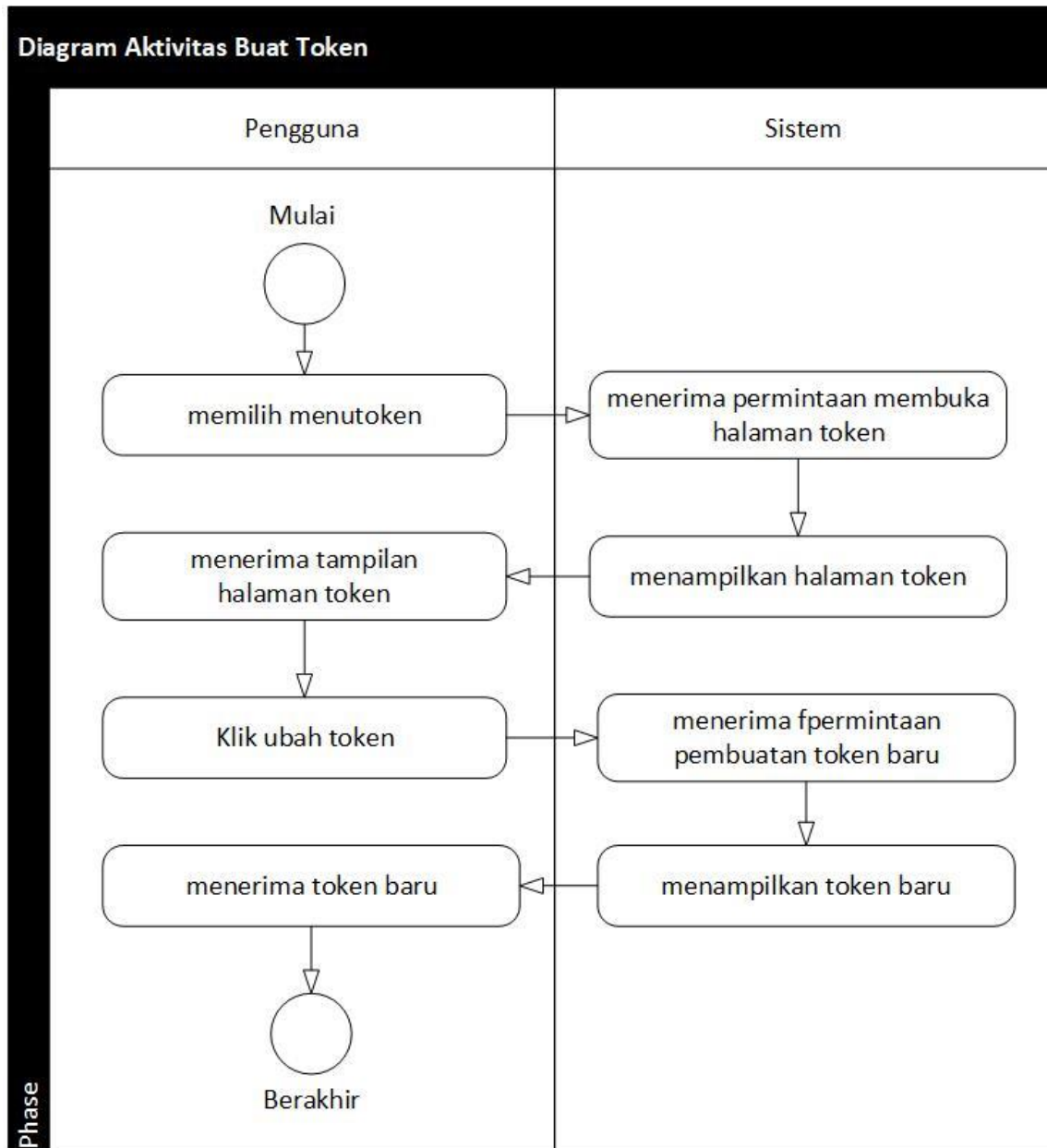
Diagram aktivitas *login* digunakan pengguna untuk masuk ke dalam sistem *natural language processing services*. Gambar 3.7 adalah skenario aktivitas *login*, dimulai dengan pengguna membuka halaman *login*, mengisi NIM dan *password*, menekan tombol *login*, dan sistem akan menampilkan halaman *dashboard*.



Gambar 3.7 Diagram Aktivitas *Login*

Diagram Aktivitas Buat *Token*

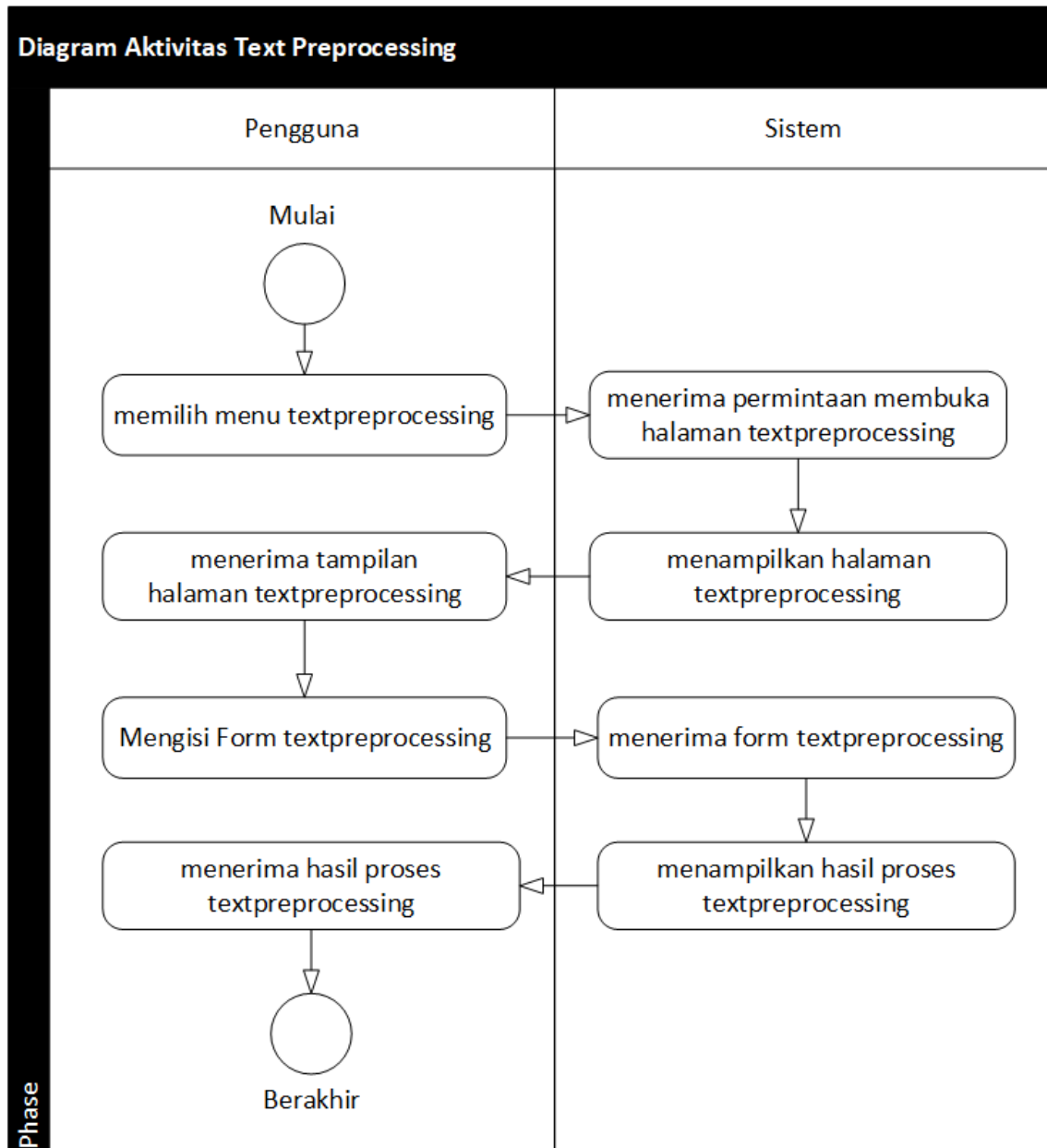
Diagram aktivitas buat *token* digunakan pengguna untuk mendapatkan *access_token* dan *access_token_secret*. Gambar 3.8 adalah skenario aktivitas buat *token* dimulai dengan pengguna membuka halaman *token*, menekan tombol ubah *login*, dan sistem akan menampilkan *token* baru.



Gambar 3.8 Diagram Aktivitas Buat *Token*

Diagram Aktivitas *Text Preprocessing*

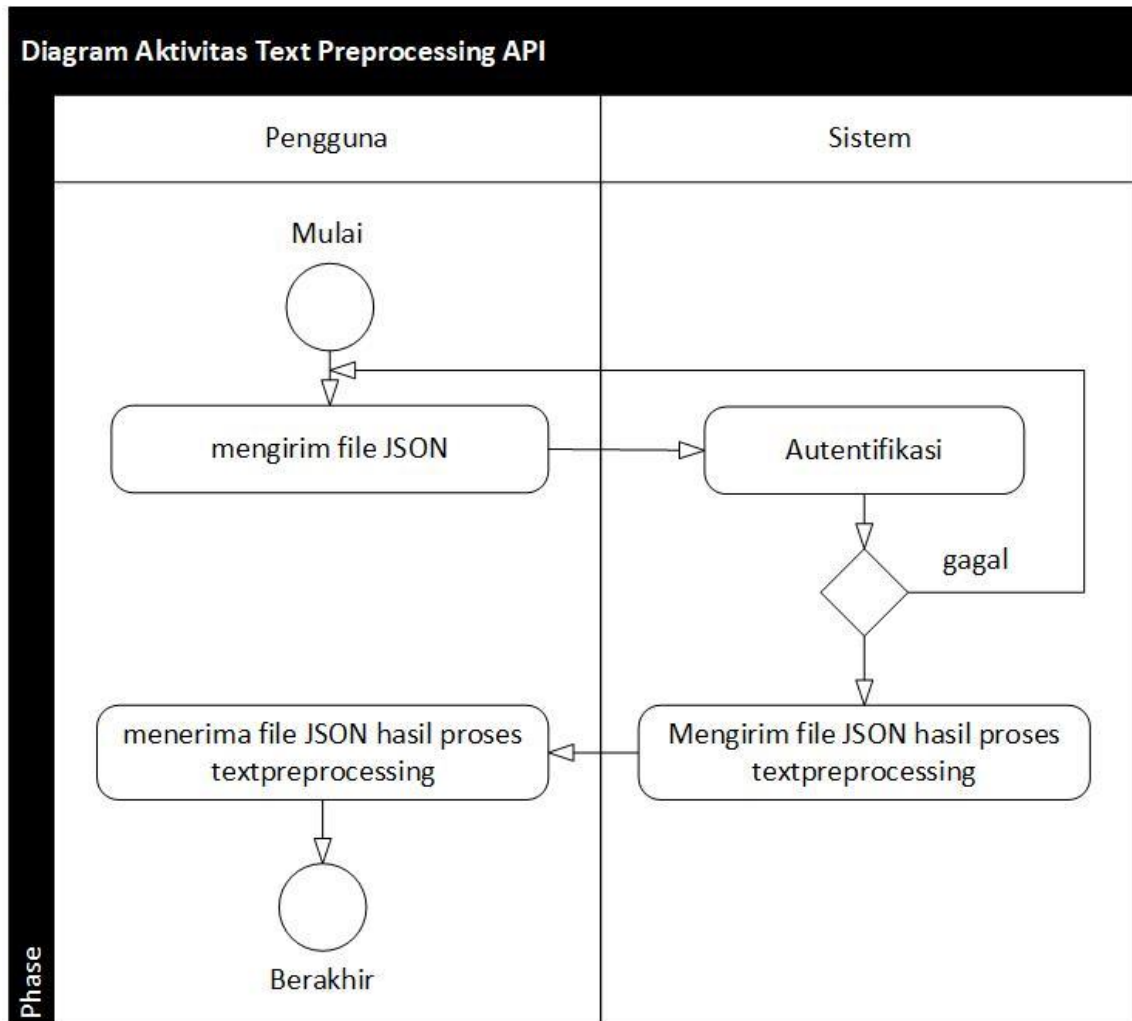
Diagram aktivitas *text preprocessing* digunakan pengguna untuk menjalankan aplikasi *text preprocessing*. Gambar 3.9 adalah skenario aktivitas *text preprocessing* dimulai dengan pengguna membuka halaman *text preprocessing*, memasukkan *text* dan memilih fitur *preprocessing* kemudian sistem akan menampilkan hasil *preprocessing*.



Gambar 3.9 Diagram Aktivitas *Text Preprocessing*

Diagram Aktivitas *Text Preprocessing* API

Diagram aktivitas *text preprocessing* API digunakan pengguna untuk mengakses aplikasi *text preprocessing* melalui API. Gambar 3.10 adalah skenario *text preprocessing* API dimulai dengan pengguna mengakses dan mengirim *file* JSON ke *endpoint text preprocessing*, kemudian sistem akan mengirimkan *file* JSON hasil *preprocessing*.



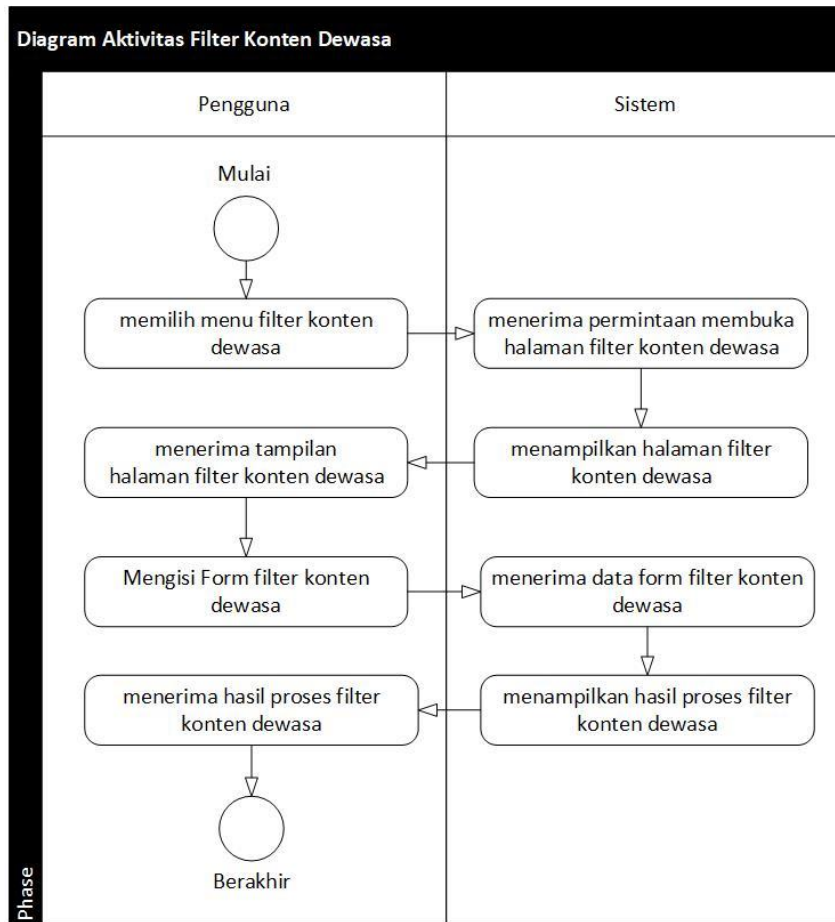
Gambar 3.10 Diagram Aktivitas *Text Preprocessing API*

Diagram Aktivitas Filter Konten Dewasa

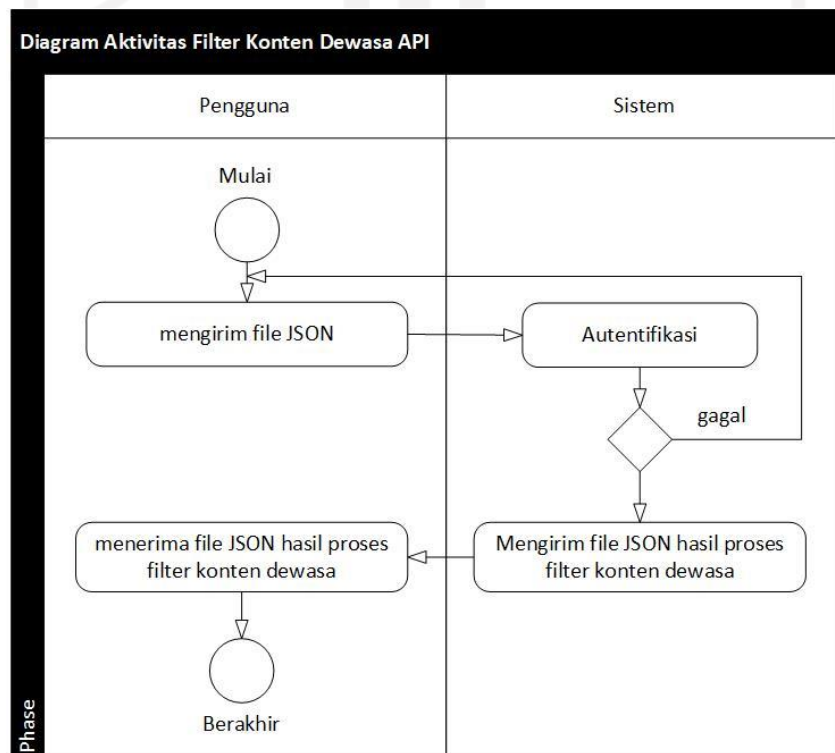
Diagram aktivitas filter konten dewasa digunakan pengguna untuk menjalankan aplikasi filter konten dewasa. Gambar 3.11 adalah skenario filter konten dewasa dimulai dengan pengguna membuka halaman filter konten dewasa, dan memasukkan teks kemudian sistem akan menampilkan hasil analisis.

Diagram Aktivitas Filter Konten Dewasa API

Diagram aktivitas filter konten dewasa API digunakan pengguna untuk mengakses aplikasi filter konten dewasa melalui API. Gambar 3.12 adalah skenario filter konten dewasa API dimulai dengan pengguna mengirim *file JSON* ke *endpoint* filter konten dewasa, kemudian sistem akan mengirimkan *file JSON* hasil analisis.



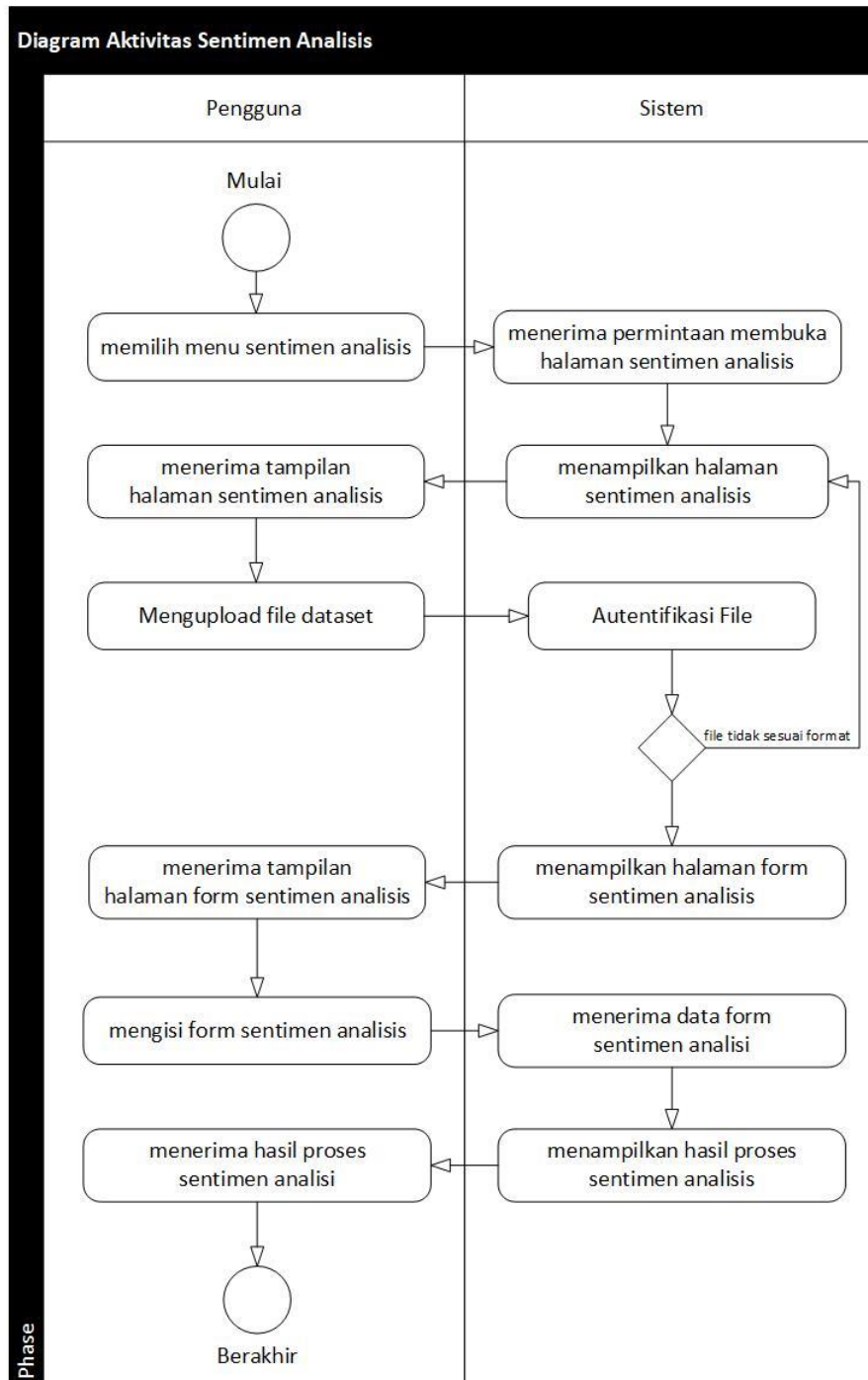
Gambar 3.11 Diagram Aktivitas Filter Konten Dewasa



Gambar 3.12 Diagram Aktivitas Filter Konten Dewasa API

Diagram Aktivitas Sentimen Analisis

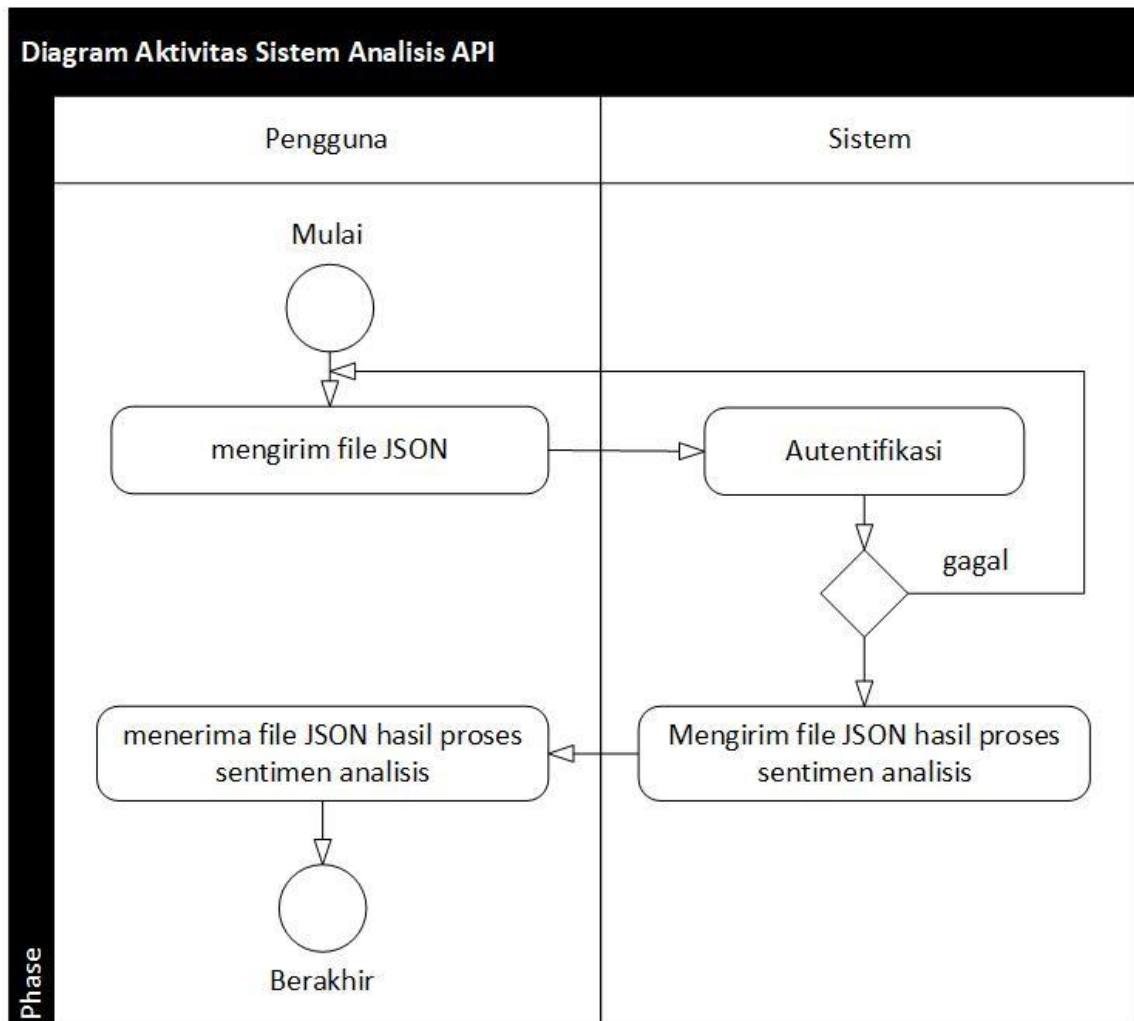
Diagram aktivitas sentimen analisis digunakan pengguna untuk menjalankan aplikasi sentimen analisis. Gambar 3.13 adalah skenario sentimen analisis dimulai dengan pengguna membuka halaman sentimen analisis, mengunggah *dataset* dan memasukkan *text* kemudian sistem akan menampilkan hasil analisis.



Gambar 3.13 Sentimen Analisis

Diagram Aktivitas Sentimen Analisis API

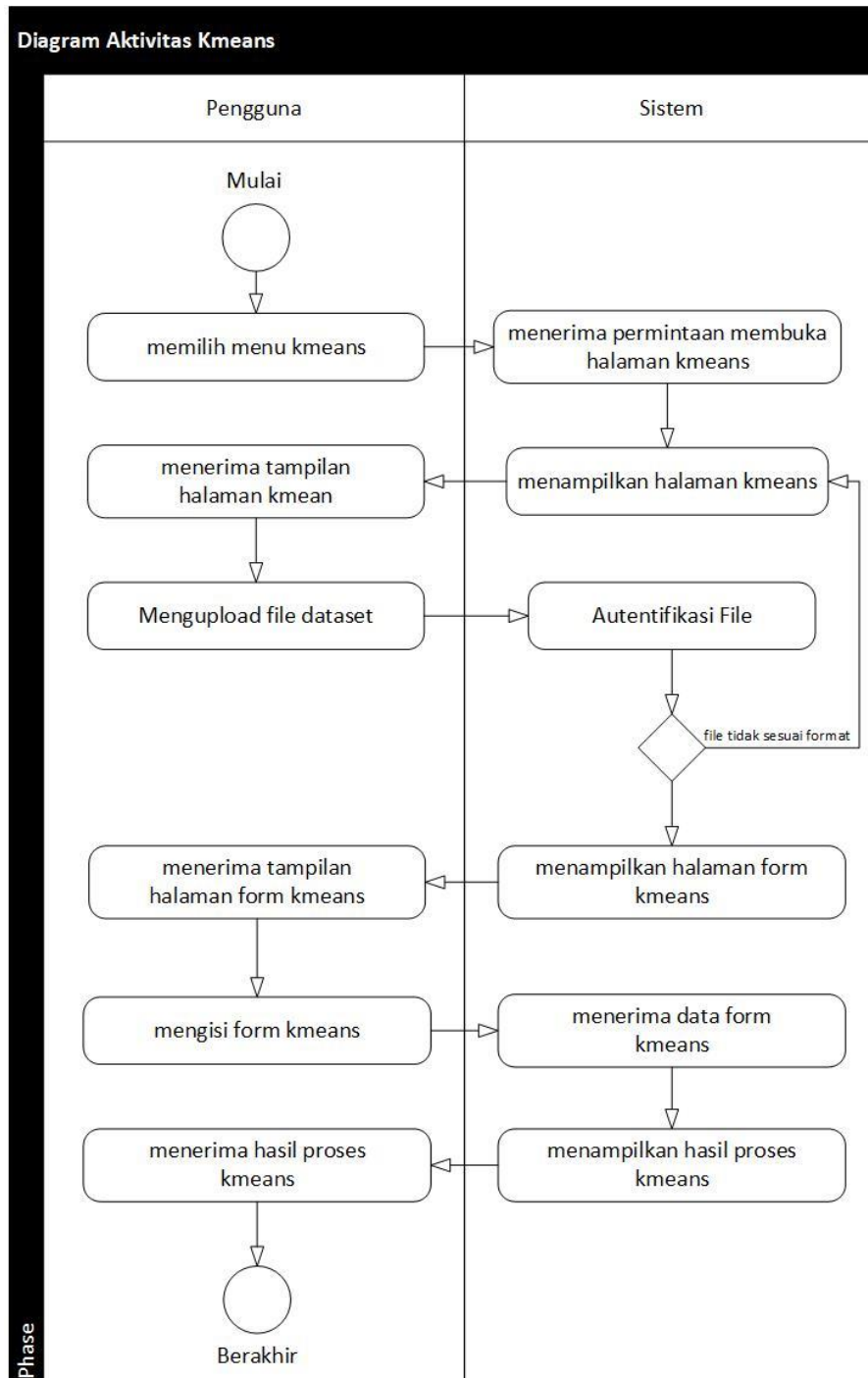
Diagram aktivitas sentimen analisis API digunakan pengguna untuk mengakses aplikasi sentimen analisis melalui API. Gambar 3.14 adalah skenario aktivitas sentimen analisis API dimulai dengan pengguna mengirim *file* JSON ke *endpoint* sentimen analisis, kemudian sistem akan mengirimkan *file* JSON hasil analisis.



Gambar 3.14 Diagram Aktivitas Sentimen Analisis API

Diagram Aktivitas K-means

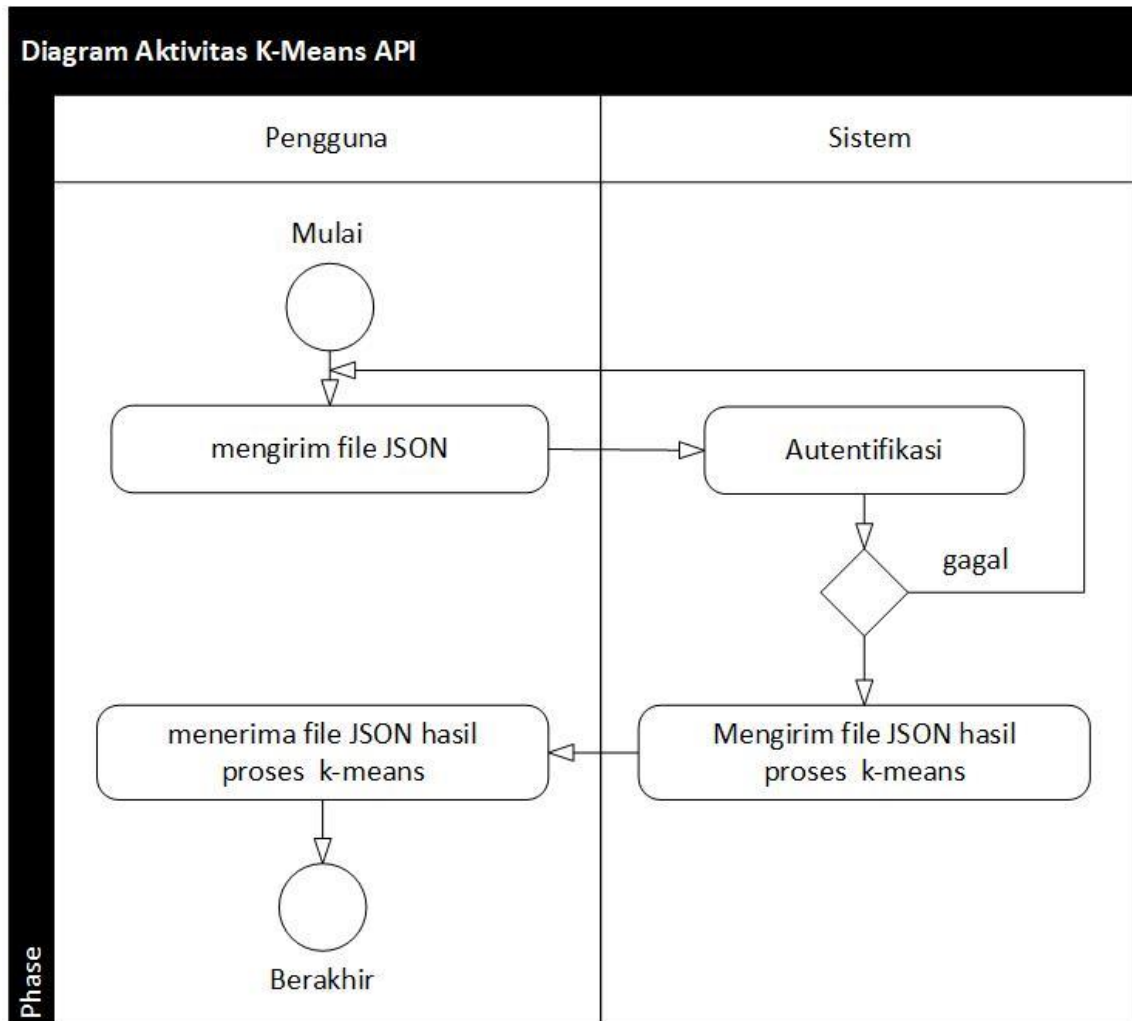
Diagram aktivitas k-means digunakan pengguna untuk menjalankan aplikasi k-means. Gambar 3.15 adalah skenario aktivitas aplikasi k-means dimulai dengan pengguna membuka halaman k-means, mengunggah *dataset* dan memasukkan *text* kemudian sistem akan menampilkan hasil analisis.



Gambar 3.15 Diagram Aktivitas K-means

Diagram Aktivitas K-means API

Diagram aktivitas k-means API digunakan pengguna untuk mengakses aplikasi k-means melalui API. Gambar 3.16 adalah skenario aplikasi k-means API dimulai dengan pengguna mengirim file JSON ke *endpoint* k-means, kemudian sistem akan mengirimkan *file* JSON hasil analisis.



Gambar 3.16 Diagram Aktivitas K-means API

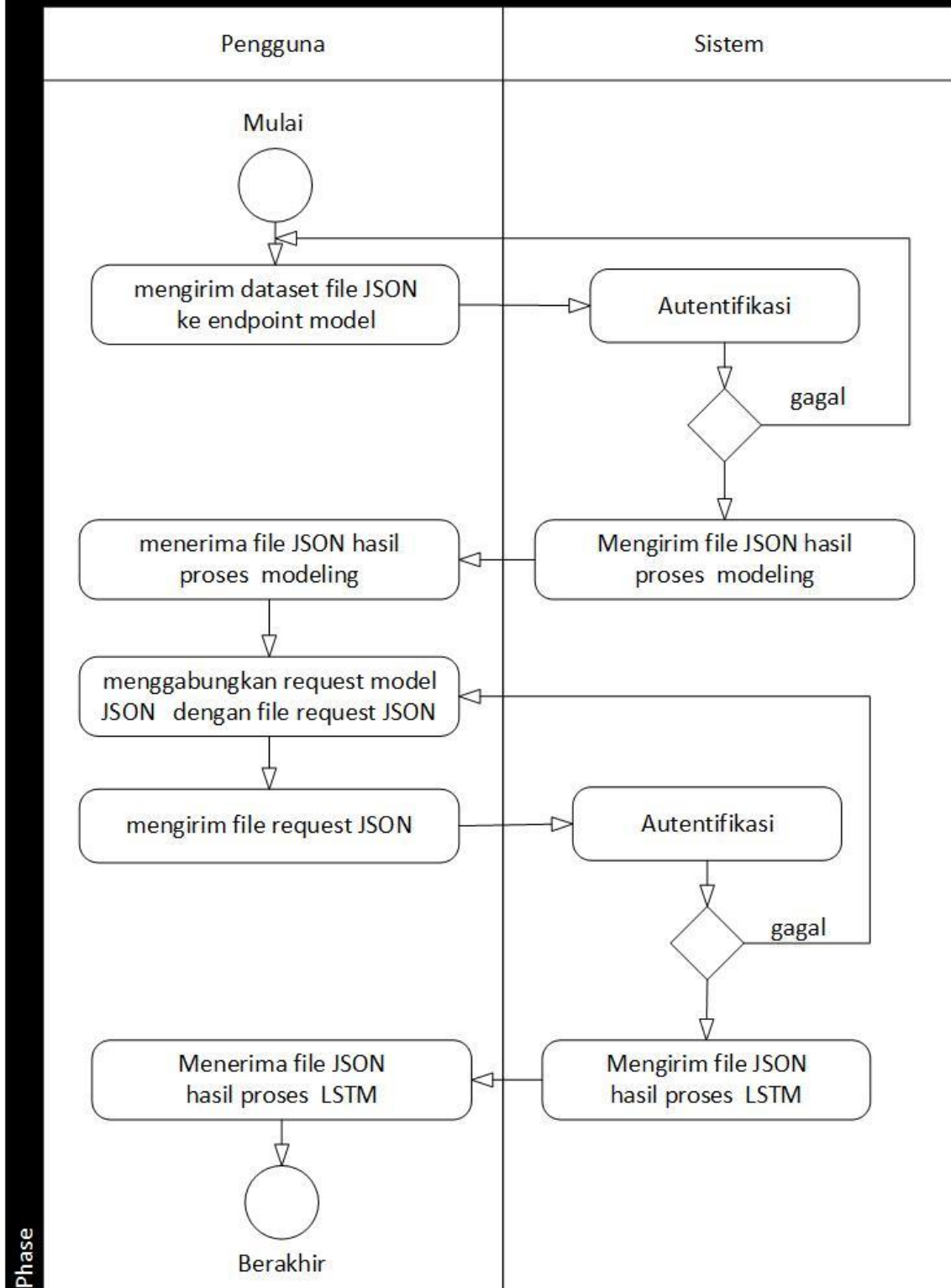
Diagram Aktivitas LSTM

Diagram aktivitas LSTM digunakan pengguna untuk menjalankan aplikasi LSTM. Gambar 3.18 adalah Skenario aktivitas aplikasi LSTM dimulai dengan pengguna membuka halaman k-means, mengunggah dataset dan memasukkan *text* kemudian sistem akan menampilkan hasil analisis.

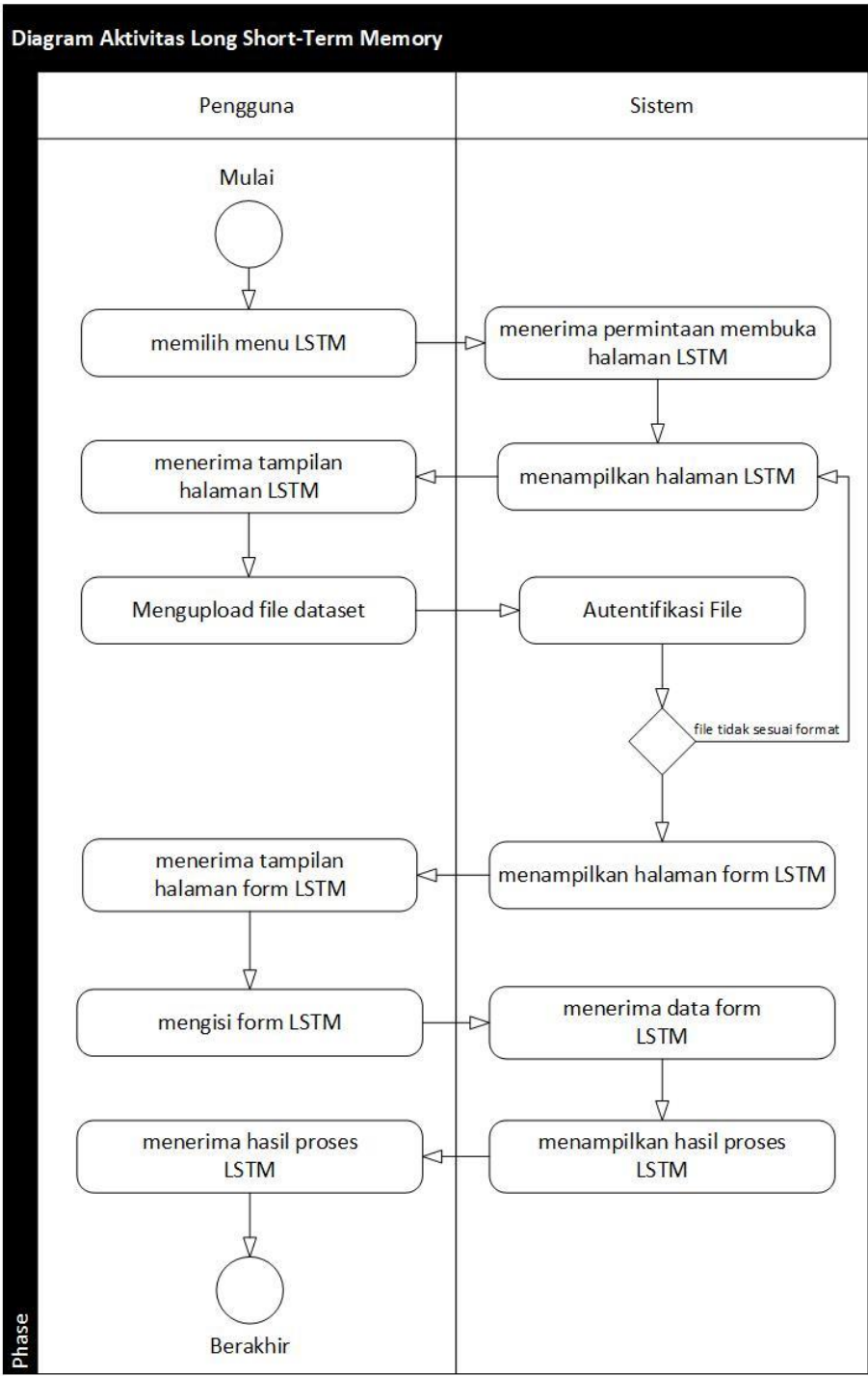
Diagram Aktivitas LSTM API

Diagram aktivitas LSTM API digunakan pengguna untuk mengakses aplikasi LSTM melalui API. Gambar 3.17 adalah skenario aktivitas aplikasi LSTM API dimulai dengan pengguna mengirim *file* JSON ke *endpoint* LSTM, kemudian sistem akan mengirimkan *file* JSON hasil analisis.

Diagram Aktivitas LSTM API



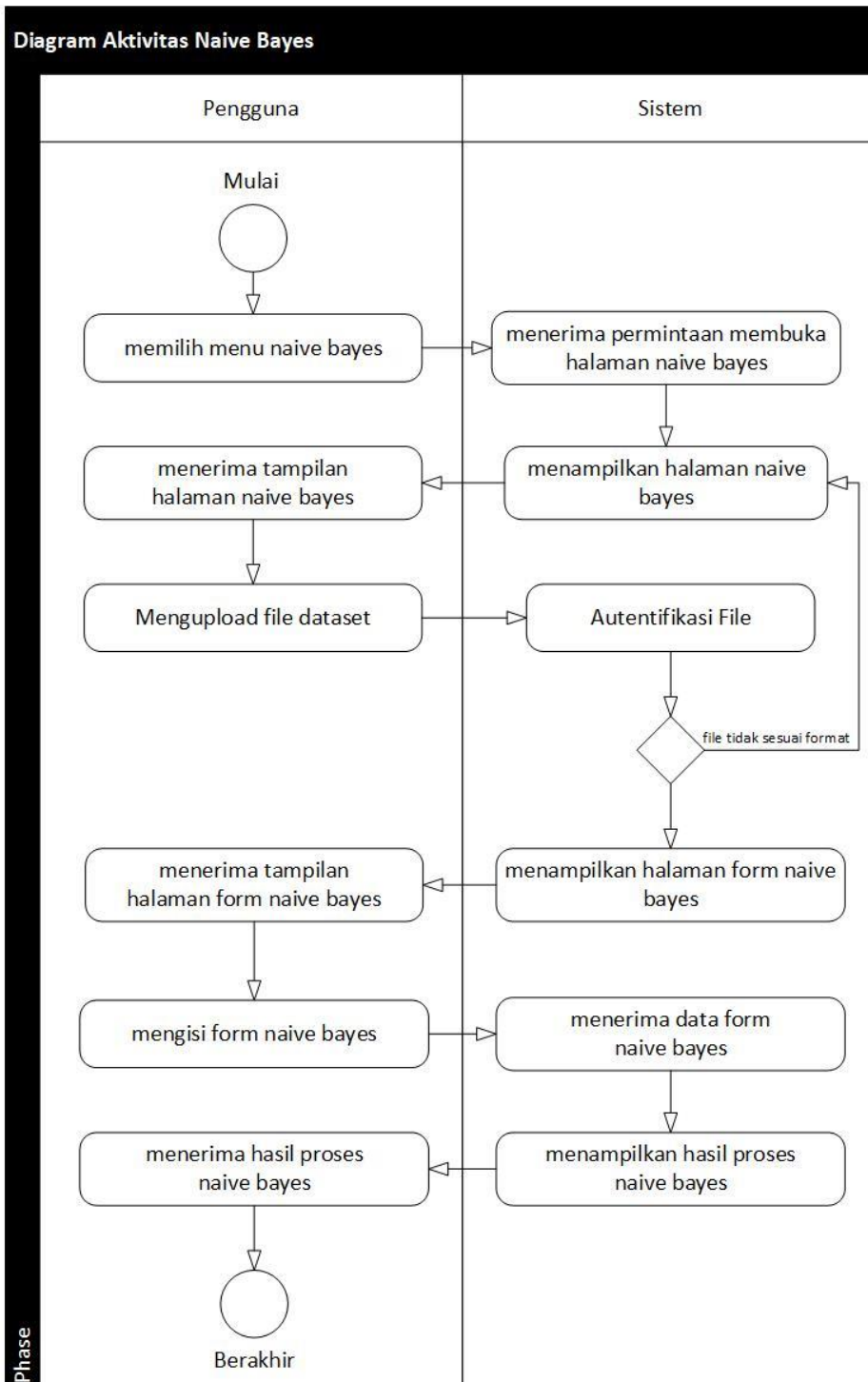
Gambar 3.17 Diagram Aktivitas LSTM API



Gambar 3.18 Diagram Aktivitas LSTM

Diagram Aktivitas Naïve Bayes

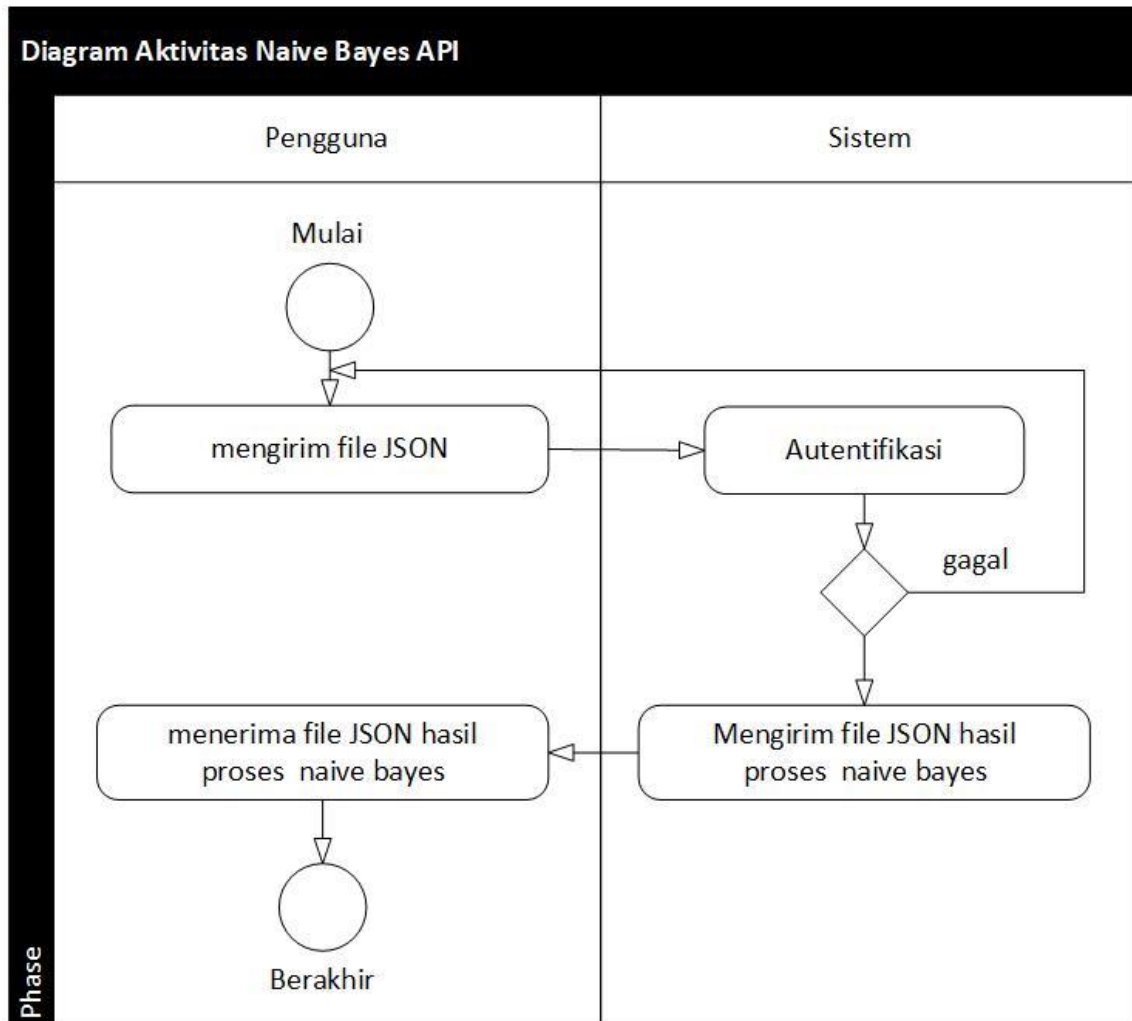
Diagram aktivitas Naïve Bayes digunakan pengguna untuk menjalankan aplikasi Naïve Bayes. Gambar 3.19 adalah skenario aktivitas aplikasi Naïve Bayes dimulai dengan pengguna membuka halaman Naïve Bayes, mengunggah *dataset* dan memasukkan *text* kemudian sistem akan menampilkan hasil analisis.



Gambar 3.19 Diagram Aktivitas *Naive Bayes*

Diagram Aktivitas *Naive Bayes* API

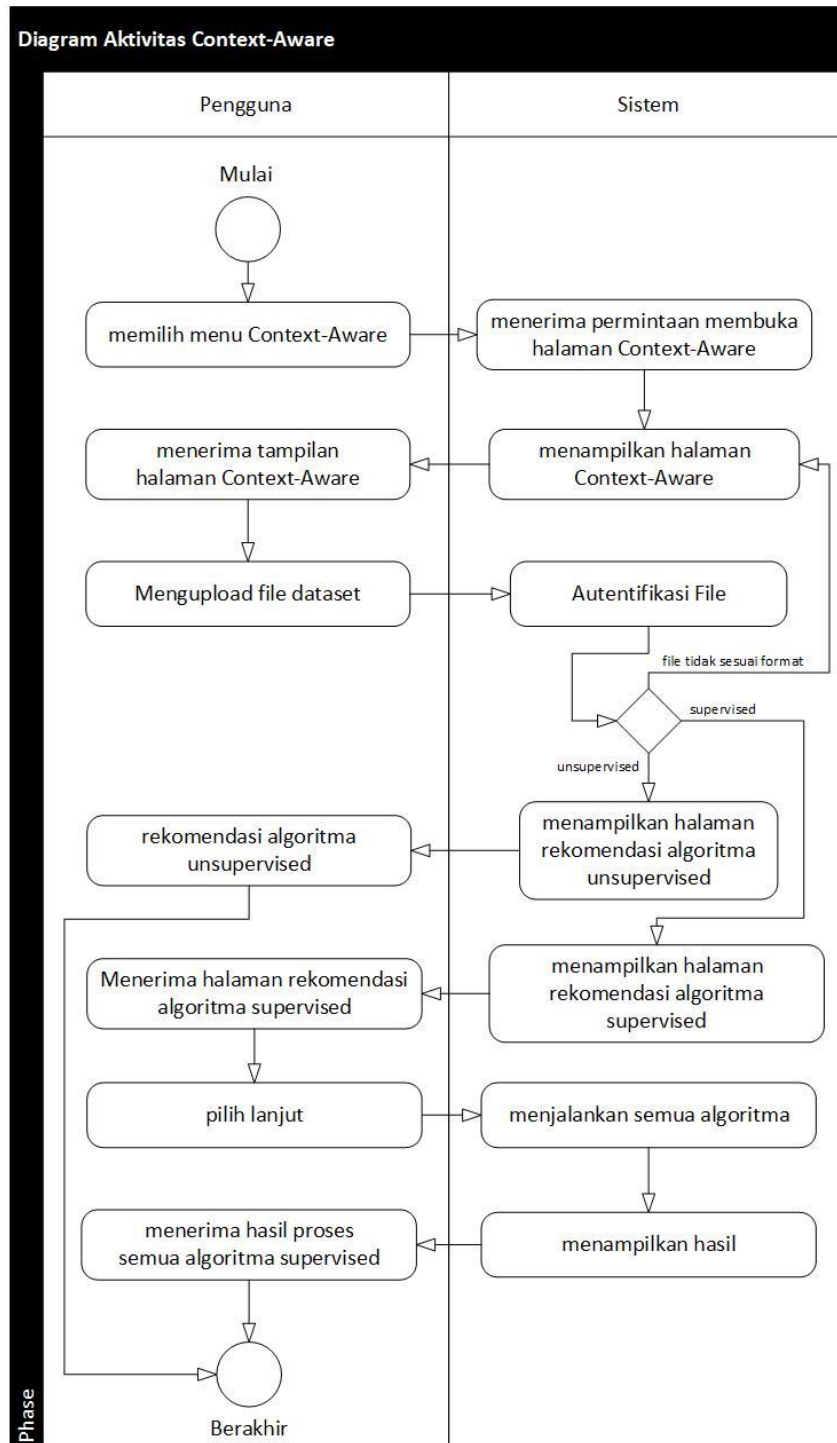
Diagram aktivitas *naive bayes* API digunakan pengguna untuk mengakses aplikasi *naive bayes* melalui API. Gambar 3.20 adalah skenario aplikasi *naive bayes* API dimulai dengan pengguna mengirim *file* JSON ke *endpoint* *naive bayes*, kemudian sistem akan mengirimkan *file* JSON hasil analisis.



Gambar 3.20 Diagram Aktivitas *Naive Bayes* API

Diagram Aktivitas *Context-Aware*

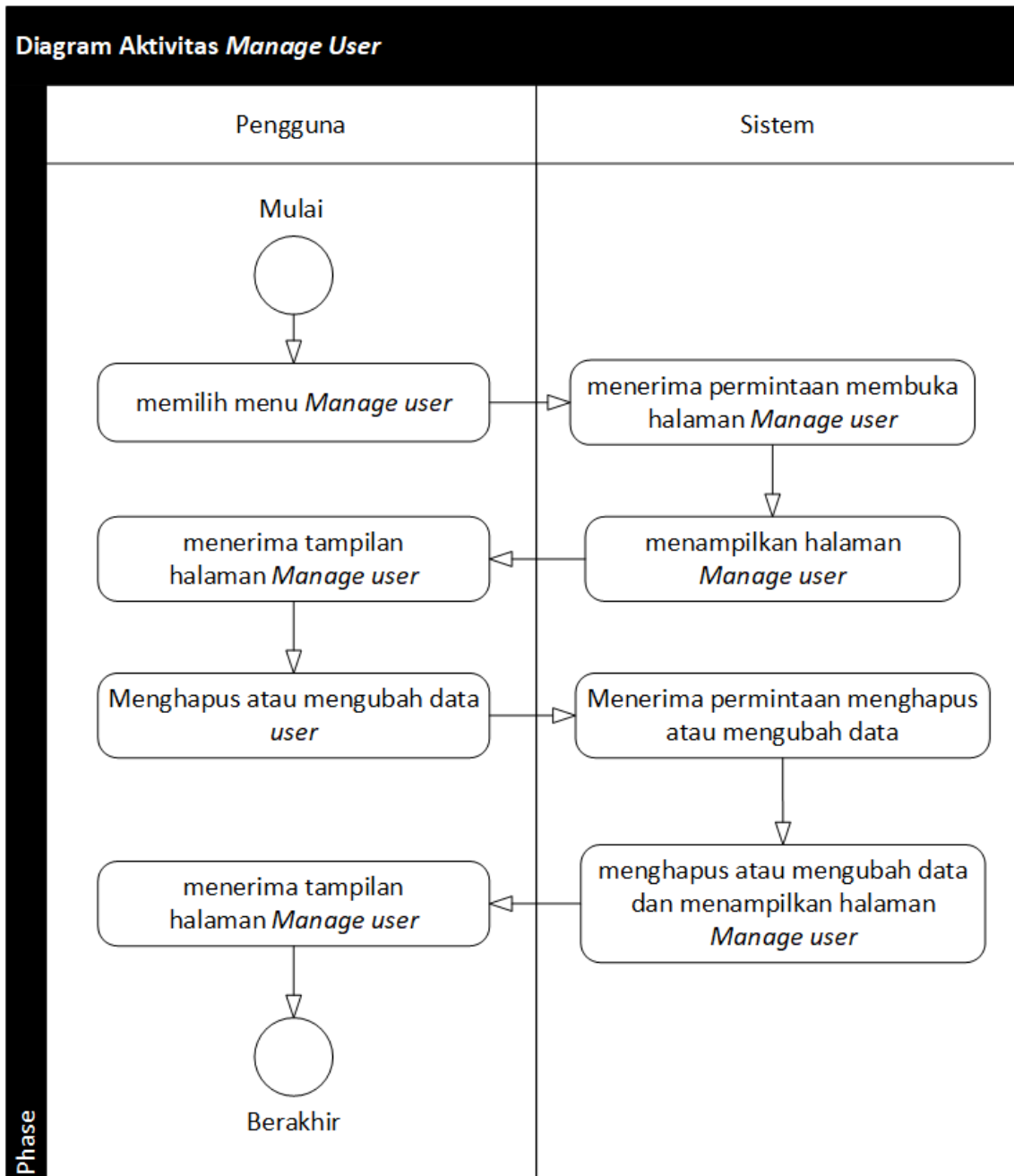
Diagram aktivitas *Context-Aware* digunakan pengguna untuk menjalankan aplikasi *Context-Aware*. Gambar 3.21 adalah skenario aktivitas aplikasi *Context-Aware* dimulai dengan pengguna membuka halaman *Context-Aware*, mengunggah *dataset*, dan sistem akan menampilkan rekomendasi algoritma. Jika *dataset* termasuk *file supervised*, pengguna dapat melanjutkan memproses *dataset* ke semua algoritma *supervised* yang tersedia.



Gambar 3.21 Diagram Aktivitas *Context-Aware*

Diagram Aktivitas *Manage user*

Diagram aktivitas *Manage User* digunakan pengguna untuk menjalankan aplikasi *Context-Aware*. Gambar 3.22 adalah skenario aktivitas aplikasi *Manage user* dimulai dengan pengguna membuka halaman *Manage user*, mengubah atau menghapus data pengguna, dan aplikasi akan mengubah atau menghapus data.



Gambar 3.22 Diagram Aktivitas *Manage User*

3.5.3 Perancangan Antarmuka Sistem

Tahap ini untuk menggambarkan tampilan antarmuka sistem. Beberapa rancangan antarmuka yang digunakan dalam sistem ini sebagai berikut:

Perancangan Antarmuka Daftar

Gambar 3.23 adalah rancangan halaman yang berfungsi untuk mendaftar ke *natural language processing service*. Data yang diperlukan untuk mendaftar adalah NIM, nama, email, *password*, ulangi *password*.

The image shows a registration form with the following elements:

- Title:** Daftar
- Subtitle:** Buat Akun Kamu
- Fields:**
 - NIM
 - Nama
 - Email
 - Password
 - Ulangi Password
- Button:** Buat Akun

Gambar 3.23 Rancangan Antarmuka Daftar

Perancangan Antarmuka *Login*

Gambar 3.24 adalah rancangan halaman *login* yang berfungsi untuk masuk ke *natural language processing service*. Data yang digunakan untuk *login* adalah NIM dan *password*

The image shows a login form with the following elements:

- Header:** LOGO
- Title:** Login
- Link:** [Forgot password?](#)
- Fields:**
 - NIM
 - Password
- Checkboxes:**
 - Show password
 - Keep me signed in.
- Buttons:**
 - Password
 - New Account
 - Buat Akun Baru
- Footer:** Sign Up for LOGO account

Gambar 3.24 Rancangan Antarmuka *Login*

Perancangan Antarmuka Buat *Token*

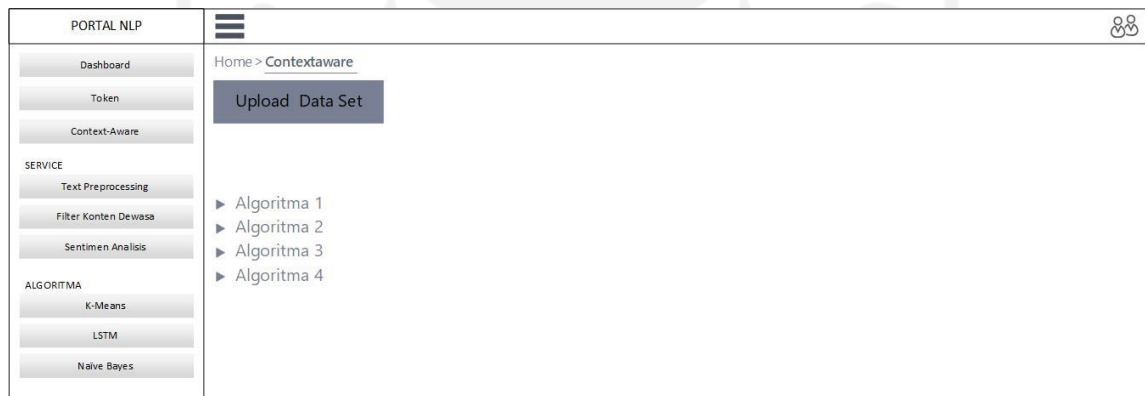
Gambar 3.25 adalah rancangan halaman *token* yang digunakan untuk mendapatkan atau mengubah *token*, pada halaman ini terdapat dua *field* untuk menampilkan *access_token* dan *access_token_secret* yang digunakan untuk mengakses API. Pengguna cukup menekan tombol ubah *token* untuk mendapatkan *token* baru atau mengubah *token*.



Gambar 3.25 Rancangan Antarmuka Buat *Token*

Perancangan Antarmuka *Context-Aware*

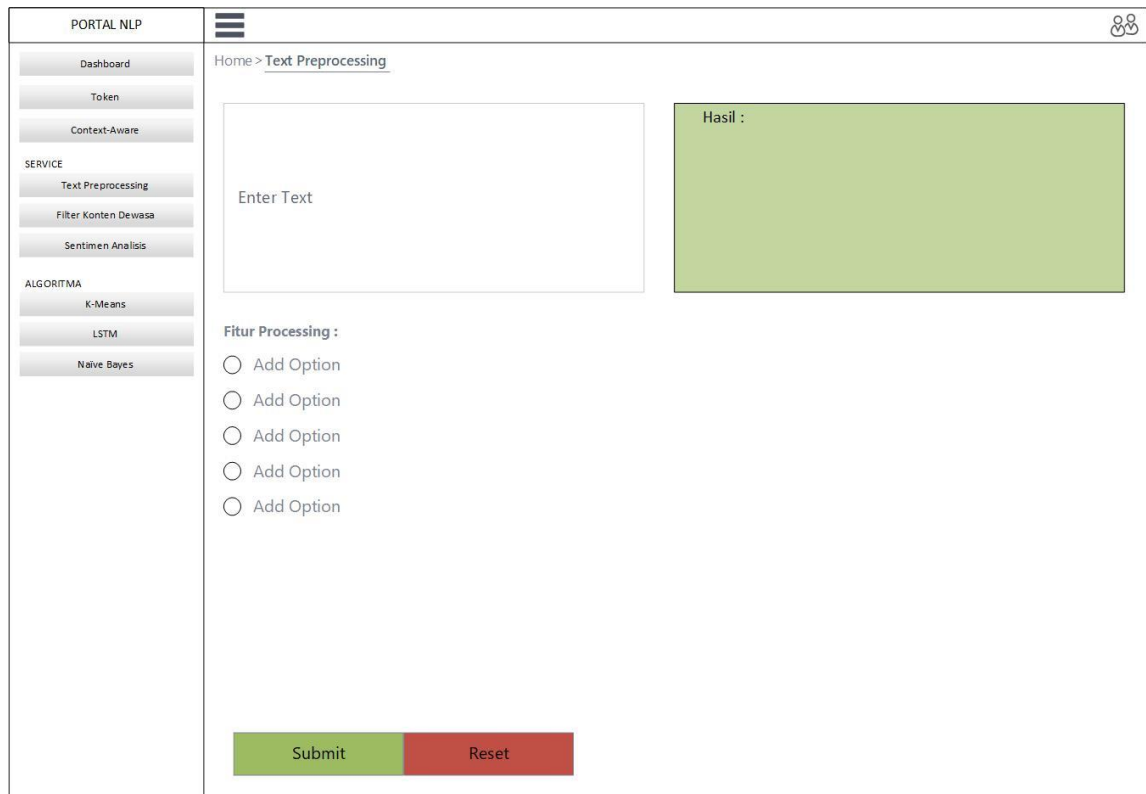
Halaman *Context-Aware* berfungsi untuk mengakses fitur *Context-Aware*. Gambar 3.26 adalah rancangan antarmuka *Context-Aware*, terdapat *button* untuk mengunggah *dataset* dan setelah *dataset* di analisis akan muncul daftar algoritma.



Gambar 3.26 Rancangan Antarmuka *Context-Aware*

Perancangan Antarmuka *Text Preprocessing*

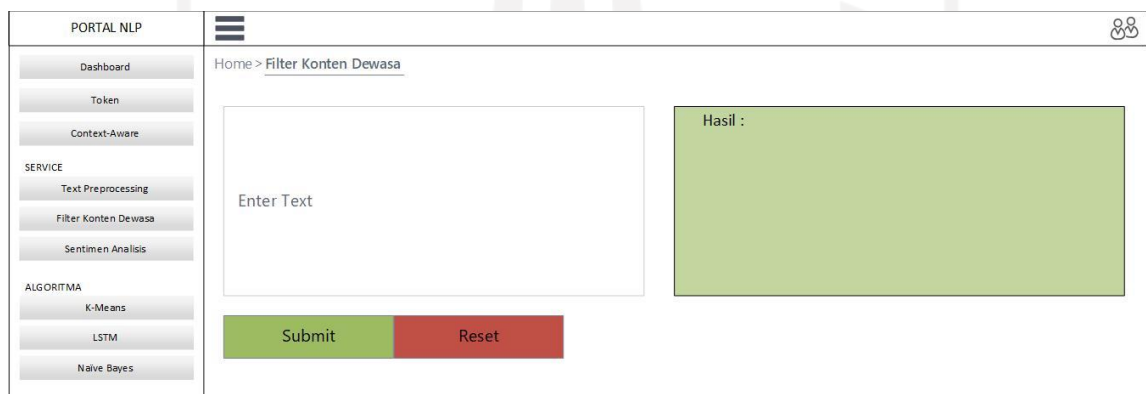
Halaman *text preprocessing* berfungsi untuk mengakses aplikasi *text preprocessing*. Gambar 3.27 adalah rancangan antarmuka *text preprocessing*, terdapat *field input text* untuk di *preprocessing* dan *field* untuk menampilkan hasil *preprocessing* serta. Terdapat juga banyak *option* yang dipilih sesuai kebutuhan *user*.



Gambar 3.27 Rancangan Antarmuka *Text Preprocessing*

Perancangan Antarmuka Filter Konten Dewasa

Halaman filter konten dewasa berfungsi untuk mengakses aplikasi filter konten dewasa. Gambar 3.28 adalah rancangan antarmuka filter konten dewasa, terdapat *field input text* dan *field* untuk menampilkan hasil.



Gambar 3.28 Rancangan antarmuka Filter Konten Dewasa

Perancangan Antarmuka Sentimen Analisis

Halaman sentimen analisis berfungsi untuk mengakses aplikasi sentimen analisis. Gambar 3.29 adalah rancangan antarmuka sentimen analisis terdapat *field input text* dan *field* untuk menampilkan hasil.

| | |
|----------------------|--------------------------|
| PORTAL NLP | Home > Sentimen Analisis |
| Dashboard | Upload Data Set |
| Token | |
| Context-Aware | |
| SERVICE | Enter Text |
| Text Preprocessing | |
| Filter Konten Dewasa | |
| Sentimen Analisis | Hasil : |
| ALGORITMA | |
| K-Means | |
| LSTM | |
| Naive Bayes | Submit Reset |

Gambar 3.29 Rancangan Antarmuka Sentimen Analisis

Perancangan Antarmuka K-means

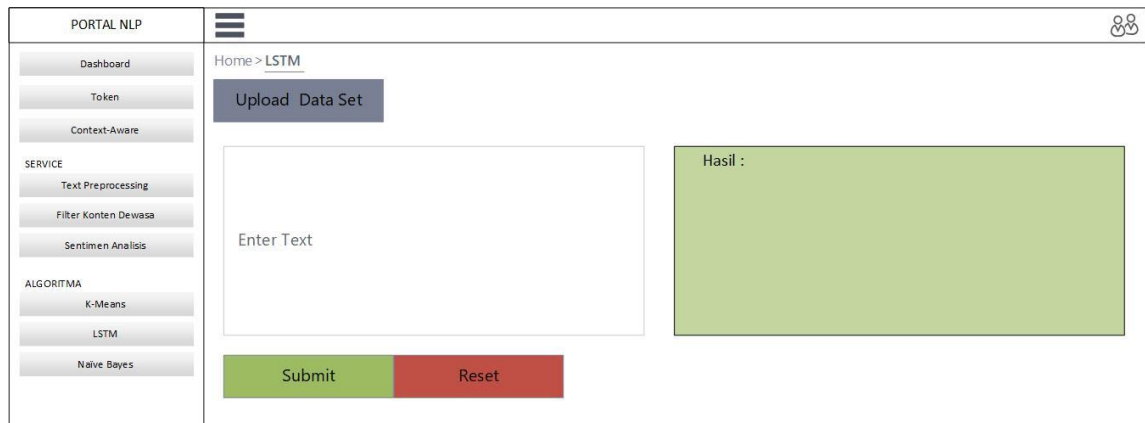
Halaman K-means berfungsi untuk mengakses aplikasi K-means. Gambar 3.30 adalah rancangan antarmuka K-means, terdapat *button* untuk mengunggah *dataset* dan dua *field* yaitu *N cluster* untuk menentukan jumlah *cluster*, *random state* digunakan untuk menentukan jumlah *random state*.

| | |
|----------------------|----------------------|
| PORTAL NLP | Home > K-Means |
| Dashboard | Upload Data Set |
| Token | |
| Context-Aware | |
| SERVICE | N Cluster : Value |
| Text Preprocessing | |
| Filter Konten Dewasa | Random State : Value |
| Sentimen Analisis | |
| ALGORITMA | |
| K-Means | |
| LSTM | |
| Naive Bayes | Submit Reset |

Gambar 3.30 Rancangan Antarmuka K-means

Perancangan Antarmuka LSTM

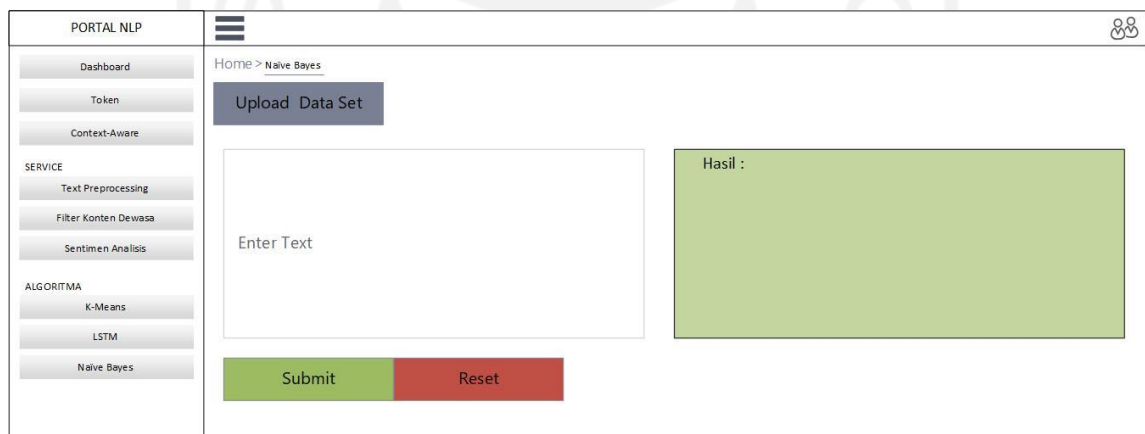
Halaman LSTM berfungsi untuk mengakses aplikasi LSTM. Gambar 3.31 adalah rancangan antarmuka LSTM, terdapat *button* untuk mengunggah *dataset*, terdapat *field input text* dan *field* untuk menampilkan hasil.



Gambar 3.31 Rancangan Antarmuka LSTM

Perancangan Antarmuka Naïve Bayes

Halaman *Naïve Bayes* berfungsi untuk mengakses aplikasi *Naïve Bayes*. Gambar 3.32 adalah rancangan antarmuka *Naïve Bayes*, terdapat *button* untuk mengunggah *dataset*, *field input text* dan *field* untuk menampilkan hasil.



Gambar 3.32 Rancangan Antarmuka *Naïve Bayes*

3.6 Pengujian Sistem

Pengujian dilakukan untuk mengetahui kesalahan-kesalahan yang terdapat pada sistem yang telah dibuat dengan melakukan pengujian menggunakan metode *black-box*. *Black-box* adalah salah satu jenis pengujian perangkat lunak yang melakukan pengujian terhadap fungsionalitas perangkat lunak (Hamza & Hammad, 2019).

Pengujian ini menggunakan beberapa Skenario sebagai berikut :

3.6.1 Skenario Pengujian *Form Login*

Tabel 3.1 adalah skenario pengujian *form login*, terdapat 5 skenario untuk pengujian sistem pada *form login*.

Tabel 3.1 Skenario Pengujian *Form Login*

| Skenario Pengujian | Hasil yang diharapkan |
|---|--|
| <i>Username</i> dan <i>Password</i> tidak diisi kemudian klik tombol <i>Login</i> | Sistem akan menolak dan menampilkan pesan “ <i>username</i> dan <i>Password</i> salah” |
| Mengetikkan <i>Username</i> , dan <i>password</i> tidak diisi atau kosong kemudian klik tombol <i>Login</i> | Sistem akan menolak dan menampilkan pesan “ <i>username</i> dan <i>Password</i> salah” |
| Mengetikkan <i>Password</i> , dan <i>username</i> tidak diisi atau kosong kemudian klik tombol <i>Login</i> | Sistem akan menolak dan menampilkan pesan “ <i>username</i> dan <i>Password</i> salah” |
| Mengetikkan <i>Username</i> dan/atau <i>password</i> tidak sesuai, kemudian klik tombol <i>Login</i> | Sistem akan menolak dan menampilkan pesan “ <i>username</i> dan <i>Password</i> salah” |
| Mengetikkan <i>Username</i> dan <i>password</i> (diisi), kemudian klik tombol <i>Login</i> | Sistem menerima akses <i>login</i> dan kemudian menampilkan halaman utama |

3.6.2 Skenario Pengujian Halaman *Token*

Tabel 3.2 adalah skenario pengujian halaman *token*, terdapat 2 skenario untuk pengujian sistem pada halaman *token*.

Tabel 3.2 Skenario Pengujian Halaman *Token*

| Skenario Pengujian | Hasil yang diharapkan |
|-----------------------------|---|
| User klik menu <i>Token</i> | Sistem menampilkan halaman <i>token</i> yang berisi <i>access_token</i> dan <i>access_token_secret</i> |
| User Klik Ubah <i>Token</i> | Sistem menerima permintaan perubahan dan melakukan perubahan pada <i>access_token</i> dan <i>access_token_secret</i> dan menampilkan pesan “Berhasil Mengubah <i>Token</i> !” |

3.6.3 Skenario Pengujian Aplikasi *Text Preprocessing*

Pengujian aplikasi ini dibagi menjadi dua yaitu pengujian pada *portal* dan pengujian pada API. Pengujian *Portal* seperti pada Tabel 3.3 menggunakan empat skenario pengujian dan Pengujian API seperti pada Tabel 3.4 menggunakan tiga skenario pengujian.

Tabel 3.3 Skenario Pengujian Aplikasi *Text Preprocessing (PORTAL)*

| Skenario Pengujian | Hasil yang diharapkan |
|--|--|
| User klik menu <i>Text Preprocessing</i> | Sistem menampilkan halaman <i>Text Preprocessing</i> yang <i>form</i> berisi <i>Text Preprocessing</i> |
| User langsung klik tombol <i>submit</i> tanpa mengisi data apa pun | Sistem menolak permintaan dan menampilkan pesan “Fitur harus diisi !” |
| User hanya mengisi beberapa data saja | Sistem menolak permintaan dan menampilkan pesan “Fitur harus diisi !” |
| User mengisi semua data | Sistem menerima permintaan dan menampilkan pesan “Data berhasil diPreprocessing” |

Tabel 3.4 Skenario Pengujian Aplikasi *Text Preprocessing (API)*

| Skenario Pengujian | Hasil yang diharapkan |
|--|---|
| User mengirimkan data json sesuai dengan format | Sistem akan mengirimkan <i>file</i> JSON hasil proses <i>text preprocessing</i> |
| User tidak mengirimkan data json sesuai dengan format | Sistem akan menolak |
| <i>access_token</i> dan <i>access_token_secret</i> tidak benar | Sistem akan menolak |

3.6.4 Skenario Pengujian Aplikasi *Filtering Konten Dewasa*

Pengujian aplikasi ini dibagi menjadi dua yaitu pengujian pada *portal* dan pengujian pada API. Pengujian *Portal* seperti pada Tabel 3.5 menggunakan tiga skenario pengujian dan Pengujian API seperti pada Tabel 3.6 menggunakan tiga skenario pengujian.

Tabel 3.5 Skenario Pengujian Aplikasi Filter Konten Dewasa (*Portal*)

| Skenario Pengujian | Hasil yang diharapkan |
|--|--|
| User klik menu Filter Konten Dewasa | Sistem menampilkan halaman Filter Konten Dewasa yang berisi <i>form Text Preprocessing</i> |
| User langsung klik tombol <i>submit</i> tanpa mengisi data apa pun | Sistem menolak permintaan dan menampilkan pesan “Fitur harus diisi !” |
| User mengisi semua data | Sistem menerima permintaan dan menampilkan pesan “Data berhasil di analisis” |

Tabel 3.6 Skenario Pengujian Aplikasi Filter Konten Dewasa (API)

| Skenario Pengujian | Hasil yang diharapkan |
|--|----------------------------|
| User tidak mengirimkan data json sesuai dengan format | Sistem akan menolak |
| <i>access_token</i> dan <i>access_token_secret</i> tidak benar | Sistem akan menolak |
| User mengirim data sesuai format | Sistem menerima permintaan |

3.6.5 Skenario Pengujian Aplikasi Sentimen Analisis

Tabel 3.7 merupakan skenario yang akan dilakukan saat melakukan pengujian terhadap aplikasi sentimen analisis, dengan menggunakan tiga skenario pengujian.

Tabel 3.7 Skenario Pengujian Aplikasi Sentimen Analisis (Portal)

| Skenario Pengujian | Hasil yang diharapkan |
|--|---|
| Pengguna klik menu Sentimen Analisis | Sistem menampilkan halaman Sentimen Analisis yang berisi untuk <i>upload file dataset</i> |
| Pengguna mengunggah <i>file dataset</i> yang sesuai format. Pengguna mengisi dan mengirim <i>text</i> yang akan di proses oleh fitur sistem analisis | Sistem menerima permintaan dan menampilkan analisis menggunakan fitur sentimen analisis |
| Pengguna mengunggah <i>file dataset</i> yang tidak sesuai format. Pengguna mengisi dan mengirim <i>text</i> yang akan di proses oleh fitur sistem analisis | Sistem tidak akan menerima dan akan menampilkan pesan <i>error</i> . |

3.6.6 Skenario Pengujian Aplikasi K-means

Tabel 3.8 merupakan skenario yang akan dilakukan saat melakukan pengujian terhadap aplikasi K-Means, dengan menggunakan tiga skenario pengujian.

Tabel 3.8 Skenario Pengujian Aplikasi K-means (*Portal*)

| Skenario Pengujian | Hasil yang diharapkan |
|---|---|
| Pengguna klik menu K-means | Sistem menampilkan halaman K-means yang berisi untuk <i>upload file dataset</i> |
| Pengguna mengunggah <i>file dataset</i> yang sesuai format. Pengguna mengisi dan mengirim jumlah <i>cluster</i> dan <i>random state</i> yang akan di proses oleh fitur K-means | Sistem menerima permintaan dan menampilkan analisis menggunakan fitur K-means |
| Pengguna mengunggah <i>file dataset</i> yang tidak sesuai format. Pengguna mengisi dan mengirim jumlah <i>cluster</i> dan <i>random state</i> yang akan di proses oleh fitur K-means | Sistem tidak akan menerima dan akan menampilkan pesan <i>error</i> . |

3.6.7 Skenario Pengujian Aplikasi LSTM

Pengujian aplikasi ini dibagi menjadi dua yaitu pengujian pada *portal* dan pengujian pada API. Pengujian *Portal* seperti pada Tabel 3.9 menggunakan empat skenario pengujian dan Pengujian API seperti pada Tabel 3.10 menggunakan tiga skenario pengujian.

Tabel 3.9 Skenario Pengujian Aplikasi LSTM (*Portal*)

| Skenario Pengujian | Hasil yang diharapkan |
|---|--|
| Pengguna klik menu LSTM | Sistem menampilkan halaman LSTM yang berisi untuk <i>upload file dataset</i> |
| Pengguna mengunggah file <i>dataset</i> yang sesuai format. | Sistem akan membuat model dari <i>dataset</i> |
| Pengguna mengunggah file <i>dataset</i> yang tidak sesuai format. | Sistem akan menampilkan pesan <i>error</i> |
| Pengguna mengisi dan mengirim <i>text</i> yang akan di proses oleh LSTM | Sistem akan menampilkan hasil proses LSTM |

Tabel 3.10 Skenario Pengujian Aplikasi LSTM (API)

| Skenario Pengujian | Hasil yang diharapkan |
|--|------------------------------|
| <i>User</i> tidak mengirimkan data json sesuai dengan format | Sistem akan menolak |
| <i>access_token</i> dan <i>access_token_secret</i> tidak benar | Sistem akan menolak |
| <i>User</i> mengirim data sesuai format | Sistem menerima permintaan |

3.6.8 Skenario Pengujian Aplikasi Naïve Bayes

Pengujian aplikasi ini dibagi menjadi dua yaitu pengujian pada *portal* dan pengujian pada API. Pengujian *Portal* seperti pada Tabel 3.11 menggunakan dua skenario pengujian dan Pengujian API seperti pada Tabel 3.12 menggunakan tiga skenario pengujian.

Tabel 3.11 Skenario Pengujian Aplikasi Naïve Bayes (*Portal*)

| Skenario Pengujian | Hasil yang diharapkan |
|---|---|
| Pengguna klik menu Naïve Bayes | Sistem menampilkan halaman Naïve Bayes yang berisi untuk <i>upload file dataset</i> |
| Pengguna mengunggah <i>file dataset</i> yang sesuai format. | Sistem akan menampilkan hasil proses Naïve Bayes dari <i>dataset</i> |

Tabel 3.12 Skenario Pengujian Aplikasi Naïve Bayes (API)

| Skenario Pengujian | Hasil yang diharapkan |
|--|------------------------------|
| <i>User</i> tidak mengirimkan data json sesuai dengan format | Sistem akan menolak |
| <i>access_token</i> dan <i>access_token_secret</i> tidak benar | Sistem akan menolak |
| <i>User</i> mengirim data sesuai format | Sistem menerima permintaan |



BAB 4

Hasil dan Pembahasan

4.1 Hasil Pengumpulan Data

Penelitian ini menggunakan *source code* yang diperoleh dengan mengumpulkan hasil penelitian tentang *Natural Language Processing* dan *source code* yang diunduh dari github. Tabel 4.1 adalah aplikasi dan algoritma yang sudah dipasang pada *Natural Language Processing Services*.

Tabel 4.1 Daftar *Source Code*

| No | Nama | Sumber |
|----|---|-----------------------------|
| 1 | Adult content classification on Indonesian tweets using LSTM neural network | (Hidayatullah et al., 2019) |
| 2 | Sentimen Analysis | (Hidayatullah, 2019) |
| 3 | Text Preprocessing | (Sembada, 2020b) |
| 4 | Long Short-Term Memory | (Sembada, 2020a) |
| 5 | Naïve Bayes | (Hidayatullah, 2021) |
| 6 | K-means | (Vanderplas, 2018) |

Workflow dari (Hidayatullah et al., 2019) dan (Sembada, 2020a) adalah *user* memberikan *input* berupa teks lewat *frontend* atau API dan *backend* akan mengirimkan hasil analisis.

Source code (Hidayatullah, 2019), (Hidayatullah, 2021) dan (Vanderplas, 2018) menggunakan bahasa pemrograman python yang ditulis menggunakan google colab. Dengan *Input* file excel serta *output*-nya berupa teks.

Source code (Sembada, 2020b) sudah menggunakan Django *rest framework* serta sudah bisa diakses melalui API. Input dan output berupa file JSON.

4.2 Hasil Refactoring Source Code

Refactoring dilakukan dengan cara *recode source code* ke *framework* Django. *Refactoring* yang dilakukan adalah sebagai berikut :

4.2.1 *Adult content classification on Indonesian tweets using LSTM neural network*

Aplikasi ini menggunakan 2 bahasa pemrograman yaitu PHP untuk *frontend* dan *python* untuk *backend*, untuk implementasi hanya menggunakan program *backend*. *Recode* dilakukan untuk mengubah pengaturan koneksi *database* seperti pada Gambar 4.1 di mana *host* awal adalah *localhost* diganti *mariadb* dan *file detectwidget.py* dilakukan perubahan seperti Gambar 4.2 yaitu *variable post_data* yang awalnya di berikan nilai melalui excel (.xlsx) diganti menggunakan JSON.

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'namadb',
        'USER': 'root',
        'PASSWORD': '*****',
        'HOST': 'localhost',
        'HOST': 'mariadb',
        'PORT': '3306',
    }
}
```

Gambar 4.1 *Recode koneksi database*

```
def post(self, request, *args, **kwargs):
    post_data = pd.read_excel('nama_file_dibaca.xlsx')
    post_data = json.loads(request.body)
    if post_data["access_token"] == "" or
    post_data["access_token"]==None:
        post_data["access_token"] = "Belum Terdaftar"
    if post_data["access_token_secret"] == "" or
    post_data["access_token_secret"]==None:
        post_data["access_token_secret"] = "Belum Terdaftar"
```

Gambar 4.2 *Recode Adult Content Classification On Indonesian Tweets Using Lstm Neural Network*

4.2.2 *Sentimen Analisis*

Kode yang digunakan hanya kode algoritma dari keseluruhan *source code*, code yang digunakan dari *source code* ini bisa dilihat pada Gambar 4.4 lalu ditulis ulang pada *controller* Django *rest framework*. Kemudian melakukan *setting database* seperti pada Gambar 4.1, autentikasi *token* seperti Gambar 4.2.

4.2.3 *Text Preprocessing* dan *Long Short-Term Memory*

Dua *source code* ini sudah siap diimplementasikan pada *natural language processing services* karena kedua *source code* ini dibuat menggunakan Django *rest framework*. *Recode* yang dilakukan hanya pada bagian *setting* koneksi *database* seperti Gambar 4.1 dan *autentikasi token* seperti Gambar 4.2.

4.2.4 **K-means**

Source code (Hidayatullah, 2019) hanya diambil kode algoritmanya, kode yang di ambil seperti pada Gambar 4.3. Kemudian di-*recode* pada *controller* Django *rest framework*. Kemudian melakukan *setting database* seperti pada Gambar 4.1 dan *autentikasi token* seperti Gambar 4.2.

```
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=4)
kmeans.fit(X)
y_kmeans = kmeans.predict(X)
centers = kmeans.cluster_centers_
```

Gambar 4.3 *Source Code* K-means

```

tfidf = TfidfVectorizer(sublinear_tf=True, min_df=5,
norm='l2', encoding='latin-1', ngram_range=(1, 2))
features = tfidf.fit_transform(df.konten).toarray()
labels = df.label
features.shape
X_train, X_test, y_train, y_test =
train_test_split(df['konten'], df['label'], random_state =
0)
count_vect = CountVectorizer()
X_train_counts = count_vect.fit_transform(X_train)
tfidf_transformer = TfidfTransformer()
X_train_tfidf =
tfidf_transformer.fit_transform(X_train_counts)
clf = MultinomialNB().fit(X_train_tfidf, y_train)
models = [
    RandomForestClassifier(n_estimators=200, max_depth=3,
random_state=0),
    LinearSVC(),
    MultinomialNB(),
    LogisticRegression(random_state=0),
]
CV = 5
cv_df = pd.DataFrame(index=range(CV * len(models)))
entries = []
for model in models:
    model_name = model.__class__.__name__
    accuracies = cross_val_score(model, features, labels,
scoring='accuracy', cv=CV)
    for fold_idx, accuracy in enumerate(accuracies):
        entries.append((model_name, fold_idx, accuracy))
cv_df = pd.DataFrame(entries, columns=['model_name',
'fold_idx', 'accuracy'])

```

Gambar 4.4 Kode Algoritma Sentimen Analisis

4.3 Hasil Containerization

Seperti pada Gambar 3.2 ada 3 langkah-langkah yang harus dilakukan untuk melakukan *containerization* adalah membuat *dockerfile*, *build* *docker image* dan menjalankan *container*. Gambar 4.5 adalah daftar *container* hasil dari proses *containerization* yang terdapat 12 *container*.

| <input type="checkbox"/> | Name | State Filter | Quick Actions |
|--------------------------|-------------------------------|-----------------|------------------|
| <input type="checkbox"/> | context-aware_app_1 | running | |
| <input type="checkbox"/> | sistem-analisis-service_app_1 | running | |
| <input type="checkbox"/> | kmeans_app_1 | running | |
| <input type="checkbox"/> | naive-bayes_app_1 | running | |
| <input type="checkbox"/> | lstm-service_app_1 | running | |
| <input type="checkbox"/> | filterkontendewasa_app_1 | running | |
| <input type="checkbox"/> | portal | running | |
| <input type="checkbox"/> | admin-service_php_1 | running | |
| <input type="checkbox"/> | admin-service_web_1 | running | |
| <input type="checkbox"/> | textpreprocessing_app_1 | running | |
| <input type="checkbox"/> | textpreprocessing_mariadb_1 | running | |
| <input type="checkbox"/> | portainer | running | |

Gambar 4.5 Daftar Container

Proses *containerization* pada setiap *container* adalah sebagai berikut :

4.3.1 *Containerization Backend Portal*

Aplikasi ini menggunakan *framework* CodeIgniter 3.1.11. Gambar 4.6 adalah konfigurasi docker-compose aplikasi *backend portal*, di mana *image* yang di gunakan adalah *nginx:latest*, *port* di *setting* ke 8020 dan *php* yang digunakan adalah 7.4. Pembuatan docker *image* menggunakan perintah *docker-compose up -d*.

```

version: "2"

services:
  web:
    image: nginx:latest
    ports:
      - 8020:80
    volumes:
      - ./code:/var/www/html
      - ./docker/nginx/logs:/var/log/nginx
      -
      ./docker/nginx/sites/default:/etc/nginx/conf.d/default.conf
      - ./docker/nginx/snippets:/etc/nginx/snippets
      -
      ./docker/nginx/fastcgi.conf:/etc/nginx/fastcgi.conf
    networks:
      - nlp-service-network

  php:
    image: php:7.4-fpm
    volumes:
      - ./code:/var/www/html
    networks:
      - nlp-service-network

networks:
  nlp-service-network:
    external:
      name: nlp-service-network

```

Gambar 4.6 Docker-compose *Backend Portal*

4.3.2 Containerization *Frontend Portal*

Aplikasi ini menggunakan *framework* Angular dan untuk membuat *container frontend portal*. Gambar 4.7 adalah konfigurasi dockerfile dan Gambar 4.8 adalah konfigurasi docker-compose, di mana *image* yang digunakan adalah angular-nginx-docker dan *port* di *setting* ke 80. Pembuatan docker *image* menggunakan perintah docker-compose *up -d*.

```

FROM alpine:3.12

ARG     timezone=Asia/Jakarta
ENV     TERM xterm

ENV     LANG en_US.UTF-8
ENV     LC_ALL en_US.UTF-8
ENV     TZ $timezone

RUN     apk update && apk upgrade \
        && apk add --no-cache \
        nginx \
        bash \
        tzdata

# Change timezone
RUN     echo $timezone > /etc/timezone \
        && cp /usr/share/zoneinfo/$timezone /etc/localtime

# Clean file
RUN     rm -rf /var/cache/apk/*

COPY ./dist/. /usr/share/nginx/html
COPY nginx.conf /etc/nginx/
ADD   conf.d /etc/nginx/conf.d

VOLUME ["/var/cache/nginx"]
EXPOSE 80 443

CMD ["nginx"]

```

Gambar 4.7 Dockerfile *Frontend Portal*

```

version: "3.7"

services:
  portal:
    container_name: portal
    image: angular-nginx-docker
    build:
      context: .
      dockerfile: Dockerfile
    volumes:
      - ./dist:/usr/share/nginx/html
    environment:
      NODE_ENV: production
    ports:
      - 80:80
    networks:
      - nlp-service-network
networks:
  nlp-service-network:
    external:
      name: nlp-service-network

```

Gambar 4.8 Docker-compose *Frontend Portal*

4.3.3 Containerization Text Preprocessing

Aplikasi ini menggunakan *django rest framework* dan untuk membuat *container text preprocessing*. Gambar 4.9 adalah konfigurasi *dockerfile* dan Gambar 4.10 adalah konfigurasi *docker-compose*, di mana *port* diatur ke 8010 dan terdapat konfigurasi *database* menggunakan *mariadb*. Pembuatan *docker image* menggunakan perintah *docker-compose up -d*.

```
FROM python:3.6
ENV PYTHONUNBUFFERED 1
RUN mkdir /code
WORKDIR /code
COPY requirements.txt /code/

RUN pip install -r requirements.txt
COPY . /code/
```

Gambar 4.9 Dockerfile Text Preprocessing

```
version: "3.7"
services:
  app:
    build: .
    command: python manage.py runserver 0.0.0.0:8010
    volumes:
      - ../code
      - ../docker/nltk:/root/nltk_data
    ports:
      - "8010:8010"
    depends_on:
      - mariadb
    links:
      - mariadb:mariadb
    networks:
      - nlp-service-network

  mariadb:
    image: mariadb:10.1.40
    restart: always
    ports:
      - "3306:3306"
    volumes:
      - dbdata:/var/lib/mysql
    environment:
      - MYSQL_DATABASE=django
      - MYSQL_ROOT_PASSWORD=rahasia
    networks:
      - nlp-service-network

volumes:
  dbdata:
    driver: local

networks:
  nlp-service-network:
    external:
      name: nlp-service-network
```

Gambar 4.10 Docker-Compose Text Preprocessing

4.3.4 *Containerization* Filter Konten Dewasa

Aplikasi ini menggunakan django *rest framework*. Gambar 4.11 adalah konfigurasi docker-compose, di mana *port* diatur ke 8030. Pembuatan docker *image* menggunakan perintah docker-compose *up -d*.

```
version: "3.7"

services:
  app:
    build: .
    command: python manage.py runserver 0.0.0.0:8030
    volumes:
      - ./code
      - ./docker/nltk:/root/nltk_data
    ports:
      - "8030:8030"
    external_links:
      - mariadb:mariadb
    networks:
      - nlp-service-network

networks:
  nlp-service-network:
    external:
      name: nlp-service-network
```

Gambar 4.11 Docker-compose Filter Konten Dewasa

4.3.5 *Containerization* Sistem Analisis

Aplikasi ini menggunakan django *rest framework*. Gambar 4.12 adalah konfigurasi docker-compose, di mana *port* diatur ke 8002. Pembuatan docker *image* menggunakan perintah docker-compose *up -d*.

4.3.6 *Containerization* LSTM

Aplikasi ini menggunakan django *rest framework*. Gambar 4.13 adalah konfigurasi docker-compose, di mana *port* diatur ke 8001. Pembuatan docker *image* menggunakan perintah docker-compose *up -d*.

4.3.7 *Containerization* Naïve Bayes

Aplikasi ini menggunakan django *rest framework*. Gambar 4.14 adalah konfigurasi docker-compose, di mana *port* diatur ke 8004. Pembuatan docker *image* menggunakan perintah docker-compose *up -d*.

```

version: "3.7"

services:
  app:
    build: .
    command: python manage.py runserver 0.0.0.0:8002
    volumes:
      - ./code
    ports:
      - "8002:8002"
    external_links:
      - mariadb:mariadb
    networks:
      - nlp-service-network

networks:
  nlp-service-network:
    external:
      name: nlp-service-network

```

Gambar 4.12 Docker-compose Sistem Analisis

```

version: "3.7"

services:
  app:
    build: .
    command: python manage.py runserver 0.0.0.0:8001
    volumes:
      - ./code
      - ./docker/nltk:/root/nltk_data
    ports:
      - "8001:8001"
    external_links:
      - mariadb:mariadb
    networks:
      - nlp-service-network

networks:
  nlp-service-network:
    external:
      name: nlp-service-network

```

Gambar 4.13 Docker-compose LSTM

4.3.8 Containerization K-means

Aplikasi ini menggunakan *django rest framework*. Gambar 4.15 adalah konfigurasi docker-compose, di mana *port* diatur ke 8005. Pembuatan docker *image* menggunakan perintah docker-compose *up -d*.

```

version: "3.7"

services:
  app:
    build: .
    command: python manage.py runserver 0.0.0.0:8004
    volumes:
      - ../code
    ports:
      - "8004:8004"
    external_links:
      - mariadb:mariadb
    networks:
      - nlp-service-network

networks:
  nlp-service-network:
    external:
      name: nlp-service-network

```

Gambar 4.14 Docker-compose Naïve Bayes

```

version: "3.7"

services:
  app:
    build: .
    command: python manage.py runserver 0.0.0.0:8005
    volumes:
      - ../code
    ports:
      - "8005:8005"
    external_links:
      - mariadb:mariadb
    networks:
      - nlp-service-network

networks:
  nlp-service-network:
    external:
      name: nlp-service-network

```

Gambar 4.15 Docker-compose K-means

4.3.9 Containerization Context-Aware

Aplikasi ini menggunakan *django rest framework*. Gambar 4.16 adalah konfigurasi *docker-compose*, di mana *port* diatur ke 8003. Pembuatan *docker image* menggunakan perintah *docker-compose up -d*.

```
version: "3.7"

services:
  app:
    build: .
    command: python manage.py runserver 0.0.0.0:8003
    volumes:
      - ./code
    ports:
      - "8003:8003"
    external_links:
      - mariadb:mariadb
    networks:
      - nlp-service-network

networks:
  nlp-service-network:
    external:
      name: nlp-service-network
```

Gambar 4.16 Docker-compose *Context-Aware*

4.4 Pemodelan Context-Aware

Aplikasi *Context-Aware* dapat mengenali file apakah *supervised* atau *unsupervised*, dengan menghitung keunikan data pada setiap kolom. Seperti yang dijelaskan pada BAB III. Dataset yang digunakan pada penelitian adalah 53 file csv, aplikasi *Context-Aware* akan menganalisis 53 dataset dan menentukan apakah termasuk file *supervised* atau *unsupervised*. Gambar 4.17 dan Gambar 4.18 adalah contoh dataset yang digunakan. Setiap dataset mempunyai jumlah kolom dan jumlah baris yang berbeda-beda. Dataset yang digunakan adalah full teks dan ada yang gabungan angka dengan teks.

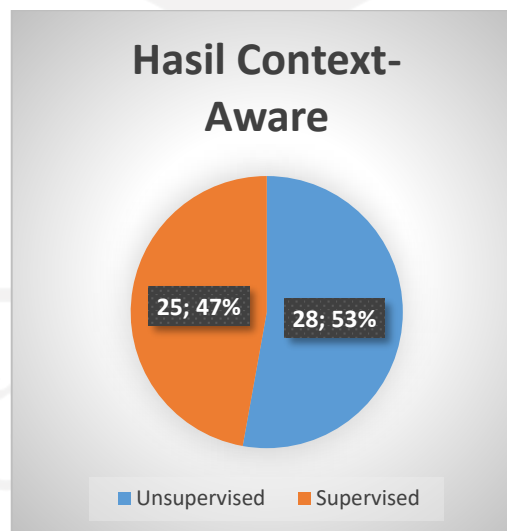
| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | |
|----|-------------------------|--|--------------|--------------------------------------|-------------------|---------------------|--------|--------|--------|--------|--------|---------------|---------------|---------------|---------------|---------------|---------------|------------|---------|-----------------|---|---|
| 1 | Name,Award | Criteria,# | Internal ID | Internal Name | Internal Category | Num of Tiers | Tier 1 | Tier 2 | Tier 3 | Tier 4 | Tier 5 | Reward Tier 1 | Reward Tier 2 | Reward Tier 3 | Reward Tier 4 | Reward Tier 5 | Reward Tier 6 | Sequential | Version | Unique Entry ID | | |
| 2 | (island name) | Miles I | Awarded when | Nook Miles program is first unlocked | 1,53 | Immigrate to Island | Event | 1,1 | 0,0 | 0,500 | 500 | 1000 | 2000 | 3000 | 0 | Yes | 1 | 0 | 0 | 0 | 0 | |
| 3 | Angling for Perfection! | Number of total fish caught | 2,0 | Catch Fish | Fish | 5 | 10 | 100 | 500 | 2000 | 5000 | 300 | 500 | 1000 | 2000 | 3000 | 0 | Yes | 1 | 0 | 0 | |
| 4 | Island Ichthyologist | Number of unique fish caught | 3,3 | Fill Fish List | Fish | 5 | 10 | 20 | 40 | 60 | 80 | 300 | 500 | 1000 | 2000 | 3000 | 0 | Yes | 1 | 0 | 0 | |
| 5 | Island Togetherness | Number of days in which you've talked to every one of your villagers | 4,68 | Greet All Villager | Communication | 5,1 | 10 | 20 | 30 | 50 | 300 | 500 | 1000 | 2000 | 3000 | 0 | Yes | 1 | 0 | 0 | | |
| 6 | You've Got the Bug | Number of total bugs caught | 5,1 | Catch Insect | Insect | 5 | 10 | 100 | 500 | 2000 | 5000 | 300 | 500 | 1000 | 2000 | 3000 | 0 | Yes | 1 | 0 | 0 | |
| 7 | Bugs Don't Bug Me | Number of unique bugs caught | 6,4 | Fill Insect List | Insect | 5 | 10 | 20 | 40 | 60 | 80 | 300 | 500 | 1000 | 2000 | 3000 | 0 | Yes | 1 | 0 | 0 | |
| 8 | Have a Nice DIY! | Number of DIY recipes collected | 7,55 | Fill Recipe List | DIY | 5 | 10 | 50 | 100 | 150 | 200 | 300 | 500 | 1000 | 2000 | 3000 | 0 | Yes | 1 | 0 | 0 | |
| 9 | DIY Tools | Number of tools you've crafted | 8,72 | DIY Tool | DIY | 5 | 5 | 50 | 200 | 1000 | 3000 | 300 | 500 | 1000 | 2000 | 3000 | 0 | Yes | 1 | 0 | 0 | |
| 10 | DIY Furniture | Number of furniture pieces you've crafted | 9,73 | DIY Furniture | DIY | 5 | 5 | 50 | 200 | 1000 | 3000 | 300 | 500 | 1000 | 2000 | 3000 | 0 | Yes | 1 | 0 | 0 | |
| 11 | Furniture Freshener | Number of times you've customized furniture at a workbench | 10,14 | Remake Furniture | DIY | 5 | 5 | 20 | 50 | 100 | 200 | 300 | 500 | 1000 | 2000 | 3000 | 0 | Yes | 1 | 0 | 0 | |
| 12 | Rough-hewn | Number of times you've chopped wood from trees | 11,67 | Strike Wood | DIY | 5 | 20 | 100 | 500 | 2000 | 5000 | 300 | 500 | 1000 | 2000 | 3000 | 0 | Yes | 1 | 0 | 0 | |
| 13 | Trashed Tools | Number of tools you've broken | 12,49 | Break Tool | HHA | 5 | 1,20 | 50 | 100 | 200 | 300 | 500 | 1000 | 2000 | 3000 | 0 | Yes | 1 | 0 | 0 | 0 | |
| 14 | Rock-Splitting Champ | Awarded the first time you get 8 ores/stones from striking a rock | 13,57 | Strike Rock | Times | HHA | 1 | 1,0 | 0,0 | 0,500 | 500 | 1000 | 2000 | 3000 | 0 | Yes | 1 | 0 | 0 | 0 | 0 | |
| 15 | Bona Fide Bone Finder! | Awarded the first time you dig up a fossil | 14,61 | Dig Fossil | HHA | 1 | 1,0 | 0,0 | 0,300 | 500 | 1000 | 2000 | 3000 | 0 | Yes | 1 | 0 | 0 | 0 | 0 | 0 | |
| 16 | Fossil Assessment | Number of fossils you've had assessed by Blathers | 15,62 | Judge Fossil | HHA | 5 | 30 | 100 | 300 | 500 | 1000 | 2000 | 3000 | 0 | Yes | 1 | 0 | 0 | 0 | 0 | 0 | |
| 17 | Greedy Weeder | Number of weed clumps you've sold to the shop | 16,37 | Sell Weed | Plant | 5 | 50 | 200 | 1000 | 2000 | 3000 | 300 | 500 | 1000 | 2000 | 3000 | 0 | Yes | 1 | 0 | 0 | |
| 18 | Flower Tender | Number of flower seeds you've planted (does not count digging up and replanting existing flowers) | 17,6 | Plant Flower Seed | Plant | 5 | 10 | 50 | 100 | 200 | 300 | 500 | 1000 | 2000 | 3000 | 0 | Yes | 1 | 0 | 0 | 0 | |
| 19 | Flower Power | Number of flowers you've watered | 18,23 | Water Plant | Plant | 5 | 10 | 50 | 100 | 500 | 1000 | 3000 | 300 | 500 | 1000 | 2000 | 3000 | 0 | Yes | 1 | 0 | 0 |
| 20 | Tomorrow's Trees Today | Number of tree saplings you've planted (does not count planting fruits or replanting existing trees) | 19,8 | Plant Tree Seedling | Plant | 3 | 5 | 10 | 30 | 0 | 0 | 300 | 500 | 1000 | 2000 | 3000 | 0 | Yes | 1 | 0 | 0 | |
| 21 | Pick of the Bunch | Number of fruits you've sold to the shop | 20,70 | Sell Fruit | Plant | 5 | 20 | 100 | 500 | 1000 | 3000 | 300 | 500 | 1000 | 2000 | 3000 | 0 | Yes | 1 | 0 | 0 | |
| 22 | Fruit Roots | Awarded for each unique fruit tree you've planted (cherry, orange, pear, peach, apple, and coconut) | 21,7 | Plant Fruit | Plant | 6 | 0 | 0 | 0 | 0 | 0 | 100 | 200 | 300 | 500 | 700 | 1000 | No | 1 | 0 | 0 | |
| 23 | Shrubbery Hubbubbery | Number of shrub starts you've planted (replanting existing shrubs) | 22,92 | Plant Bush Seedling | Plant | 3 | 1,2 | 20 | 0 | 0 | 300 | 500 | 1000 | 0 | 0 | 0 | Yes | 1 | 2 | 0 | 0 | |
| 24 | Go Ahead. Be Shellfish! | Number of seashells you've sold to the shop | 23,69 | Sell Shell | Fish | 5 | 10 | 50 | 200 | 500 | 1000 | 300 | 500 | 1000 | 2000 | 3000 | 0 | Yes | 1 | 0 | 0 | |
| 25 | Clam and Collected | Number of clams you've dug up | 24,63 | Dig Shell | Fish | 5 | 5 | 20 | 50 | 100 | 200 | 300 | 500 | 1000 | 2000 | 3000 | 0 | Yes | 1 | 0 | 0 | |
| 26 | Trash Fishin' | Number of trash items (cans, tires, or boots) you've fished up | 25,32 | Catch Trash | Fish | 3 | 3 | 10 | 20 | 0 | 0 | 300 | 500 | 1000 | 2000 | 3000 | 0 | Yes | 1 | 0 | 0 | |

Gambar 4.17 Contoh *Dataset 1*

| | A | B | C | D | E | F | G | H | I | J | K | L |
|----|--|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Name,Buy,Category,Source,Filename,Version,Unique Entry ID | | | | | | | | | | | |
| 2 | brick bridge,198000,Bridge,Resident Services Upgrade,BridgeBricks,1.0.0,Rvkrq9TcubTMakcBS | | | | | | | | | | | |
| 3 | iron bridge,228000,Bridge,Resident Services Upgrade,BridgeIron,1.0.0,RmQ6iikKfF3Sy4iSQ | | | | | | | | | | | |
| 4 | log bridge,98000,Bridge,Resident Services Upgrade,BridgeLog,1.0.0,Px55sTfjKSbbMtjAK | | | | | | | | | | | |
| 5 | red zen bridge,228000,Bridge,Resident Services Upgrade,BridgeRed,1.0.0,jTziEHXkeYaugKDQQ | | | | | | | | | | | |
| 6 | stone bridge,168000,Bridge,Resident Services Upgrade,BridgeStone,1.0.0,JmrwC2tRRCCgbZnpr | | | | | | | | | | | |
| 7 | suspension bridge,129800,Bridge,Resident Services Upgrade,BridgeSuspension,1.0.0,q52JebMTHDqHJKBba | | | | | | | | | | | |
| 8 | wooden bridge,168000,Bridge,Resident Services Upgrade,BridgeWood,1.0.0,7oZ7FMrJEpBLZzsmw | | | | | | | | | | | |
| 9 | zen bridge,228000,Bridge,Resident Services Upgrade,BridgeJapanese,1.0.0,TbKbHzJM8j8ezmAEF | | | | | | | | | | | |
| 10 | basic door,5000,Door,4th House Upgrade (Right Room),HouseDoorCercleWindowAR,1.0.0,HrAsZSpm2pRj2u8ad | | | | | | | | | | | |
| 11 | beige basic door,5000,Door,4th House Upgrade (Right Room),HouseDoorCercleWindowGR,1.0.0,uF4YeCTGBx4Tawcp8 | | | | | | | | | | | |
| 12 | beige simple door,5000,Door,4th House Upgrade (Right Room),HouseDoorSimpleBR,1.0.0,jQFLgioHYBRbRkBYK | | | | | | | | | | | |
| 13 | beige windowed door,5000,Door,4th House Upgrade (Right Room),HouseDoorWindowGR,1.0.0,oMqf22Pf8fZzeJdY | | | | | | | | | | | |
| 14 | beige wooden door,5000,Door,4th House Upgrade (Right Room),HouseDoorStandardGR,1.0.0,N5LASg9qjhgexBMe7 | | | | | | | | | | | |
| 15 | black common door,5000,Door,4th House Upgrade (Right Room),HouseDoorReliefHR,1.0.0,6mmdmWbK5ZRpbCsWc | | | | | | | | | | | |
| 16 | black fancy door,5000,Door,4th House Upgrade (Right Room),HouseDoorCarvingBR,1.0.0,3ZahH8gvtHGdkEM68 | | | | | | | | | | | |
| 17 | black imperial door,5000,Door,4th House Upgrade (Right Room),HouseDoorChineseHR,1.0.0,qT9ghD7YkZpMDgBdf | | | | | | | | | | | |
| 18 | black latticework door,5000,Door,4th House Upgrade (Right Room),HouseDoorLatticeHR,1.0.0,Kzu4WzWmSWzLdMc6 | | | | | | | | | | | |
| 19 | black rustic door,5000,Door,4th House Upgrade (Right Room),HouseDoorSimplicityDR,1.0.0,TKDhdqyJBdrz8yQt6 | | | | | | | | | | | |
| 20 | black vertical-panes door,5000,Door,4th House Upgrade (Right Room),HouseDoorVerticalWindowHR,1.0.0,Z4efmoeAjzYlHtsKJ | | | | | | | | | | | |
| 21 | black zen door,5000,Door,4th House Upgrade (Right Room),HouseDoorJapaneseER,1.0.0,AxrkW9SNKqmh9kX4S | | | | | | | | | | | |
| 22 | blue basic door,5000,Door,4th House Upgrade (Right Room),HouseDoorCercleWindowER,1.0.0,BxqZmtEty82ND08J3 | | | | | | | | | | | |
| 23 | blue common door,5000,Door,4th House Upgrade (Right Room),HouseDoorReliefCR,1.0.0,hfWfx4mxeV3qjp2zH | | | | | | | | | | | |
| 24 | blue fancy door,5000,Door,4th House Upgrade (Right Room),HouseDoorCarvingER,1.0.0,fm6MupiuPAK6HPNNj | | | | | | | | | | | |
| 25 | blue iron door,5000,Door,4th House Upgrade (Right Room),HouseDoorIronCR,1.0.0,5tgy5p3AE38NBeG9B | | | | | | | | | | | |
| 26 | blue latticework door,5000,Door,4th House Upgrade (Right Room),HouseDoorLatticeFR,1.0.0,JwdJLi5HHZe24Dmkf | | | | | | | | | | | |

Gambar 4.18 Contoh *Dataset 2*

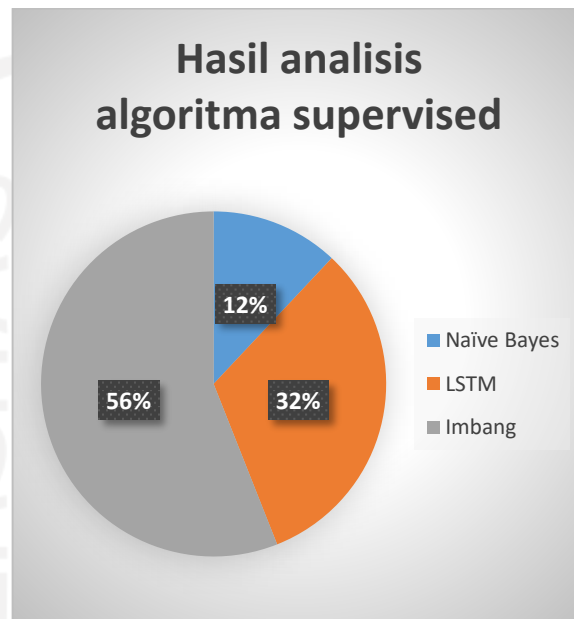
Dari Gambar 4.19 bisa dilihat hasil dari 53 dataset yang diuji sistem bisa mendeteksi 28 file *unsupervised* dan 25 file *supervised*.



Gambar 4.19 Hasil *Context-Aware*

Algoritma *unsupervised* tidak bisa dibandingkan akurasi, sehingga aplikasi *Context-Aware* hanya menampilkan rekomendasi algoritma *unsupervised*. Sedangkan untuk algoritma *supervised*, aplikasi menghitung akurasi dari dataset yang diunggah dengan masing-masing algoritma untuk dibandingkan tingkat akurasi.

Gambar 4.20 merupakan hasil dari analisis algoritma *supervised* oleh fitur *Context-Aware*. Dari 25 file sistem merekomendasikan 3 file menggunakan naïve bayes dan 8 file menggunakan LSTM. Sebanyak 14 file kedua algoritma di rekomendasikan karena hasil pengujian akurasinya sama. Berdasarkan hasil Gambar 4.20 dilakukan pengujian secara manual, dengan menjalankan satu per satu dataset menggunakan aplikasi naïve bayes dan LSTM, hasil yang didapat sama seperti Gambar 4.20.



Gambar 4.20 Hasil Analisis Algoritma *Supervised*

4.5 Hasil Implementasi Sistem

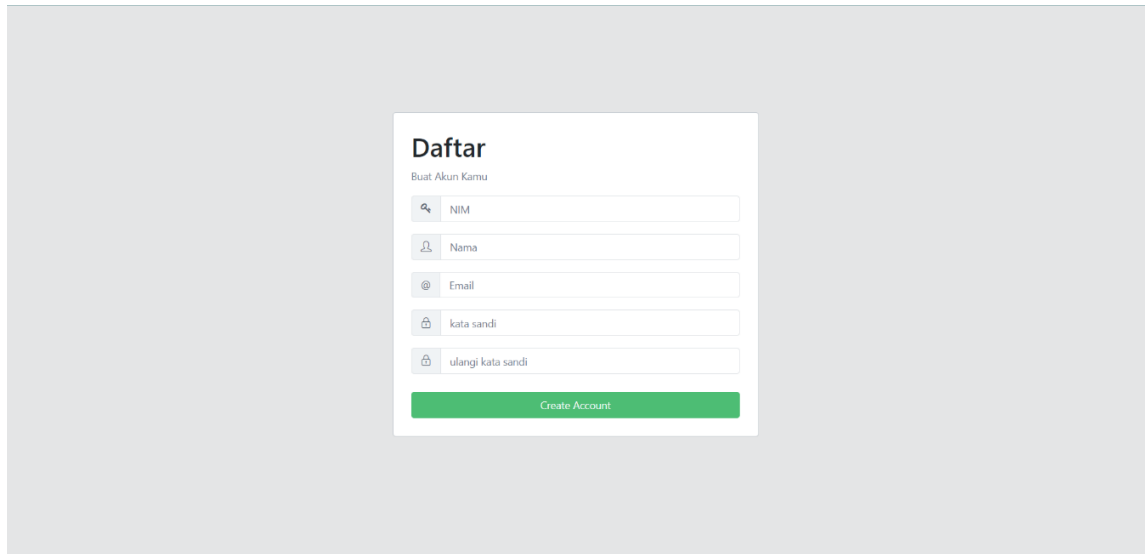
Natural Language Processing Services dapat di akses melalui dua cara yaitu dengan menggunakan *portal* dan mengakses menggunakan API.

4.5.1 Akses *Portal*

Portal Natural Language Processing Services memudahkan pengguna yang kurang menguasai programing. Karena dengan menggunakan *portal* pengguna cukup menggunakan UI yang sudah disediakan tanpa perlu melakukan konfigurasi API. Berikut hasil implementasi sistem pada *portal NLP Services* :

Halaman Daftar

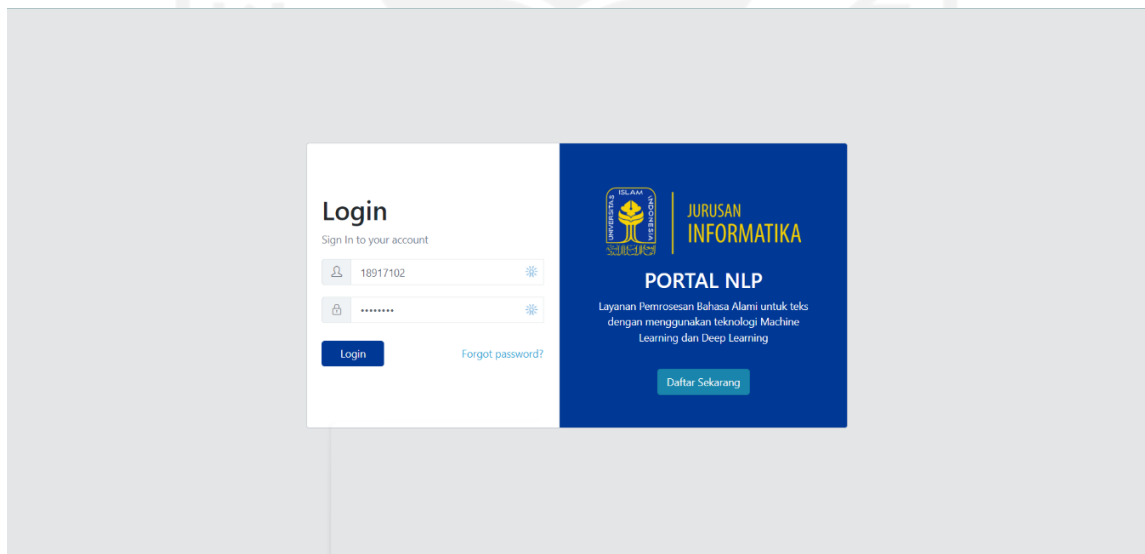
Gambar 4.21 adalah halaman daftar yang digunakan pengguna untuk mendaftar ke sistem *Natural Language Processing Services*. Data yang harus di *input* oleh pengguna adalah NIM, Nama, Email, Kata sandi dan Ulangi kata sandi.



Gambar 4.21 Halaman Daftar

Halaman *Login*

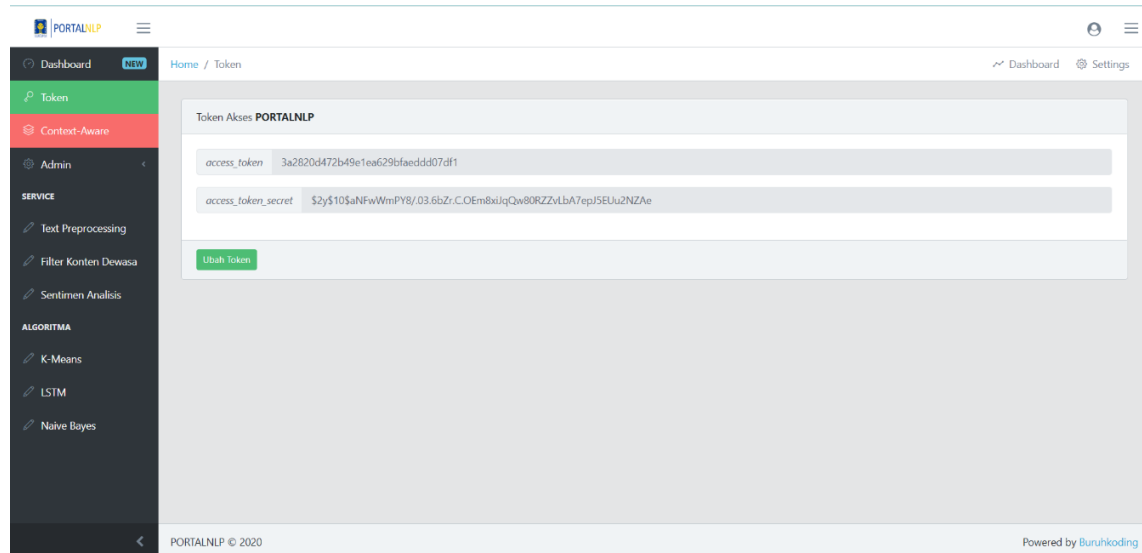
Gambar 4.22 adalah halaman login yang digunakan pengguna untuk masuk ke sistem *Natural Language Processing Services*. Data yang harus di *input* oleh pengguna adalah NIM dan kata sandi, pada halaman ini juga terdapat *button* untuk mendaftar bagi pengguna yang belum mempunyai akses ke *PORTAL NLP*.



Gambar 4.22 Halaman *Login*

Halaman *Token*

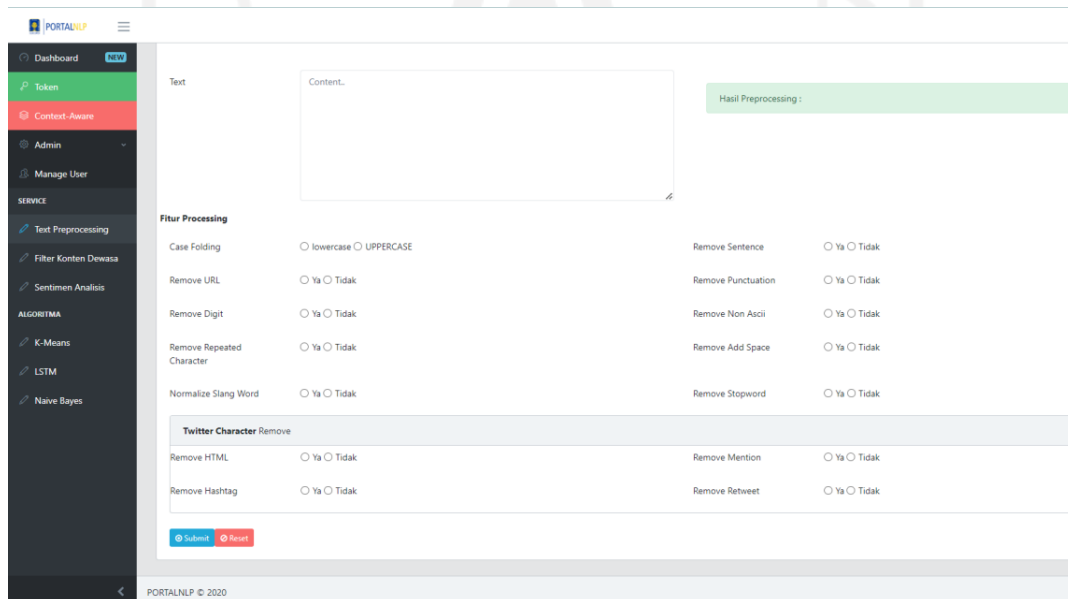
Halaman ini digunakan pengguna untuk mendapatkan atau mengubah *access_token* dan *access token secret*. *Token* berfungsi untuk mengakses *Natural Language Processing Services* melalui API. Untuk mendapatkan atau mengubah *token*, pengguna hanya perlu menekan *button* ubah *token*. Tampilan halaman daftar dapat dilihat pada Gambar 4.23.



Gambar 4.23 Halaman *Token*

Halaman *Text Preprocessing*

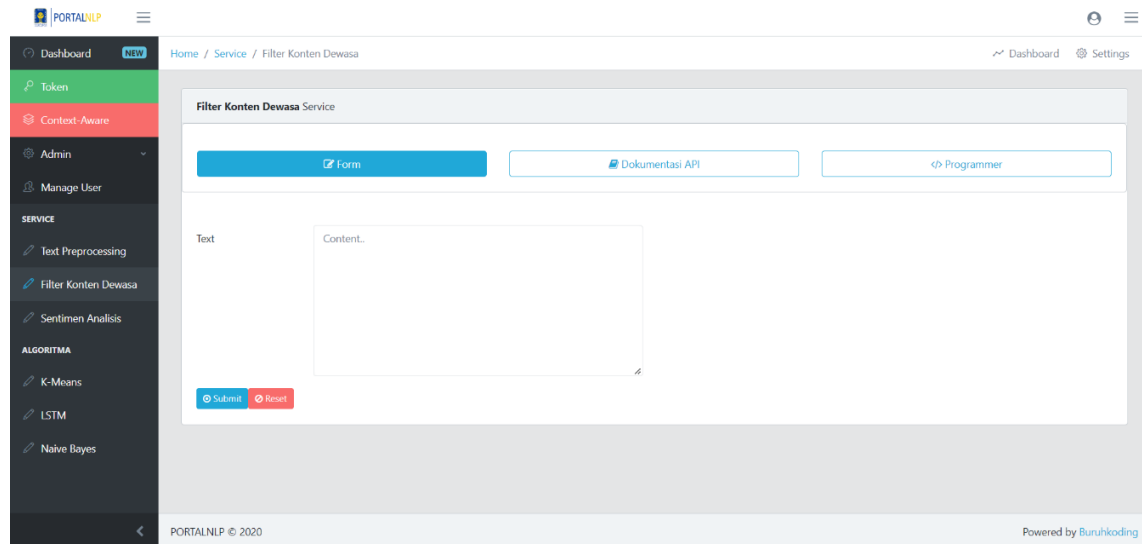
Halaman ini digunakan pengguna untuk mengakses aplikasi *text preprocessing*. Data yang harus di *input* oleh pengguna adalah *text* untuk dilakukan *text preprocessing*, serta memilih fitur *processing* sesuai kebutuhan pengguna. Tampilan halaman *text preprocessing* dapat dilihat pada Gambar 4.24.



Gambar 4.24 Halaman *Text preprocessing*

Halaman Filter Konten Dewasa

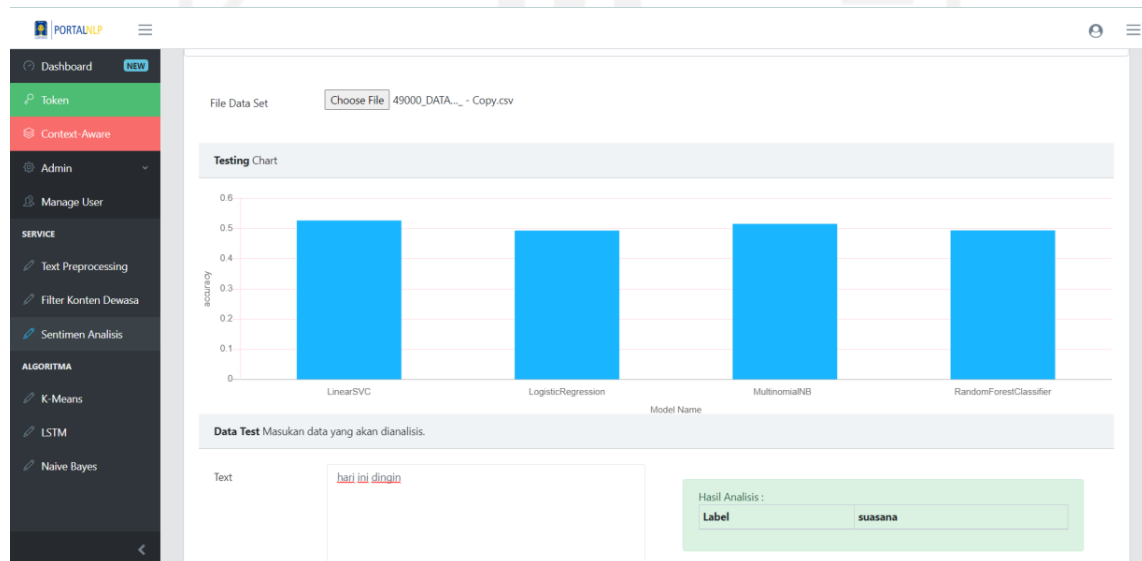
Halaman ini digunakan pengguna untuk mengakses aplikasi filter konten dewasa. Data yang harus di *input* oleh pengguna adalah *text* untuk dilakukan proses *filtering*. Tampilan halaman filter konten dewasa dapat dilihat pada Gambar 4.25.



Gambar 4.25 Halaman Filter Konten Dewasa

Halaman Sentimen Analisis

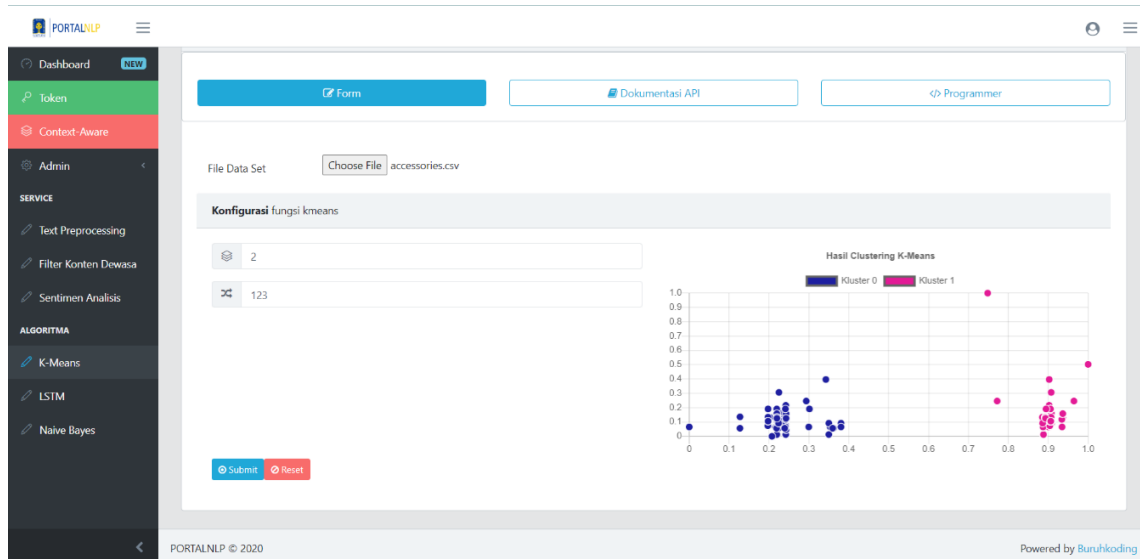
Halaman ini digunakan pengguna untuk mengakses aplikasi sentimen analisis. Data yang harus di *input* oleh pengguna adalah dataset dan *text* untuk dilakukan proses sentimen analisis. Tampilan halaman sentimen analisis dapat dilihat pada Gambar 4.26.



Gambar 4.26 Halaman Sentimen Analisis

Halaman K-means

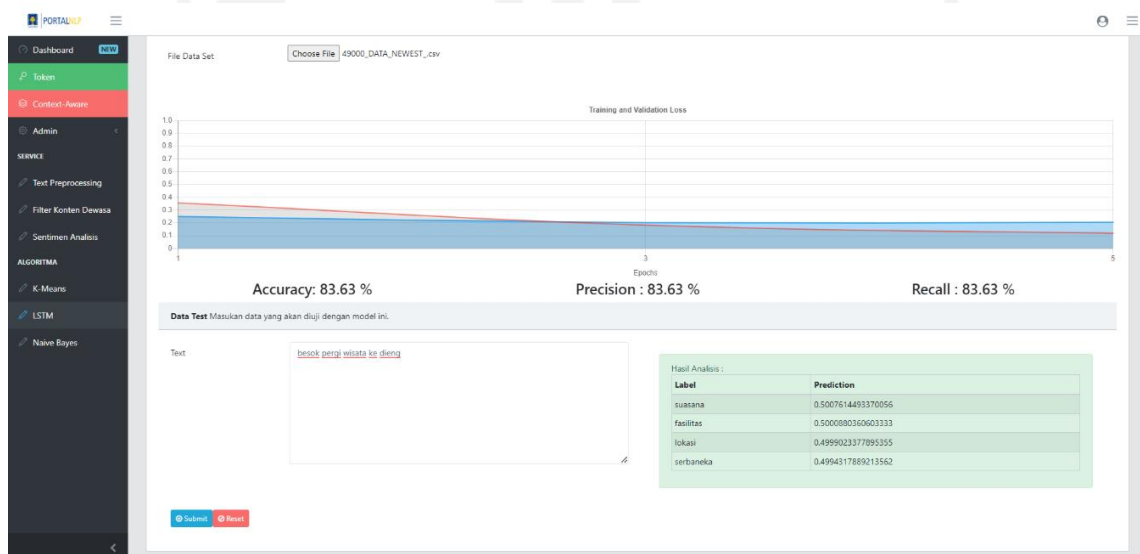
Halaman ini digunakan pengguna untuk mengakses aplikasi K-means. Data yang harus di *input* oleh pengguna adalah dataset, jumlah *cluster* dan *random state* untuk dilakukan proses *clustering* menggunakan algoritma K-means. Tampilan halaman K-means dapat dilihat pada Gambar 4.27.



Gambar 4.27 Halaman K-means

Halaman LSTM

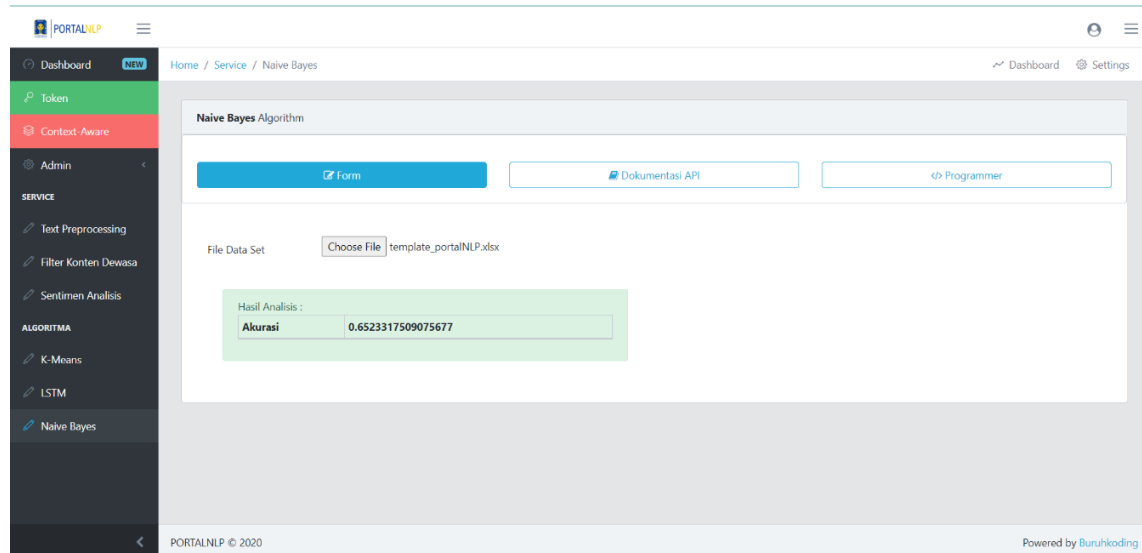
Halaman ini digunakan pengguna untuk mengakses aplikasi LSTM. Data yang harus di *input* oleh pengguna adalah dataset dan untuk dilakukan proses dengan algoritma LSTM. Tampilan halaman LSTM dapat dilihat pada Gambar 4.28.



Gambar 4.28 Gambar Halaman LSTM

Halaman Naïve Bayes

Halaman ini digunakan pengguna untuk mengakses aplikasi naïve bayes. Data yang harus di *input* oleh pengguna adalah dataset. Sistem membagi dataset menjadi data *training* dan data *testing*, kemudian sistem akan menampilkan hasil akurasi menggunakan algoritma naïve bayes. Tampilan halaman naïve bayes dapat dilihat pada Gambar 4.29.



Gambar 4.29 Halaman Naïve Bayes

4.5.2 Akses API

API *Natural Language Processing Services* bisa digunakan untuk menghubungkan *source code* atau aplikasi lain dengan *Natural Language Processing Services*. Untuk mendapatkan akses API pengguna harus mempunyai *token* yang bisa didapatkan pada menu *token* pada portal *Natural Language Processing Services*. Berikut hasil implementasi API *Natural Language Processing Services* :

API Text Preprocessing

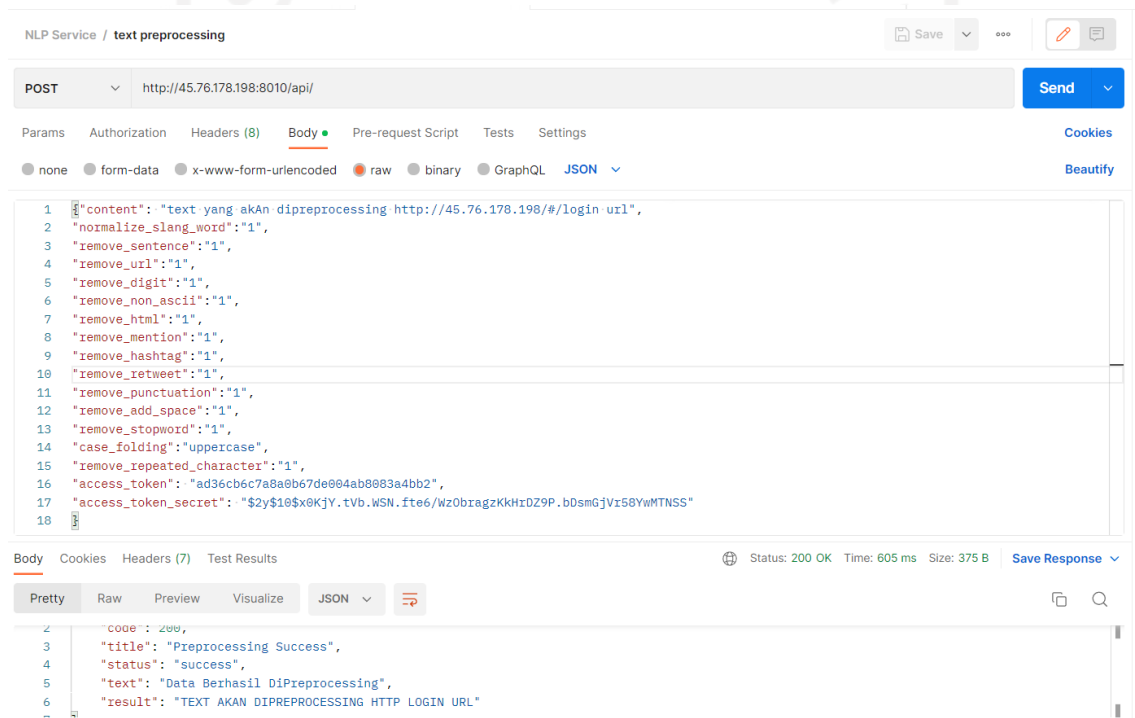
Akses API *text preprocessing* seperti Gambar 4.30. *Port* yang digunakan untuk API *text preprocessing* adalah 8010 dan *endpoint* yang digunakan adalah <http://45.76.178.198:8010/api/>. *Input* paramater yang digunakan untuk API *text preprocessing* seperti Tabel 4.2. Hasil *request* API seperti Tabel 4.3.

Tabel 4.2 *Input* Parameter API *Text* Preprocessing

| Parameter | | Keterangan |
|---------------------------|----------|--|
| content | required | <i>content</i> adalah <i>text</i> yang akan di preprocessing |
| access_token | required | access_token adalah token yang digunakan agar dapat mengakses <i>service</i> |
| access_token_secret | required | access_token_secret adalah token yang digunakan agar dapat mengakses <i>service</i> |
| remove_stopword | required | remove_stopword adalah fitur untuk menghapus kata stopwords |
| normalize_slang_word | required | normalize_slang_word adalah fitur untuk menormalisasi kata slang |
| remove_sentence | required | remove_sentence adalah fitur untuk menghapus kalimat |
| remove_url | required | remove_url adalah fitur untuk menghapus URL |
| remove_digit | required | remove_digit adalah fitur untuk menghapus digit |
| remove_non_ascii | required | remove_non_ascii adalah fitur untuk menghapus karakter yang bukan ASCII |
| remove_html | required | remove_html adalah fitur untuk menghapus code html dalam <i>text</i> |
| remove_mention | required | remove_html adalah fitur untuk menghapus mention |
| remove_hashtag | required | remove_hashtag adalah fitur untuk menghapus hashtag |
| remove_retweet | required | remove_retweet adalah fitur untuk menghapus retweet |
| remove_punctuation | required | remove_punctuation adalah fitur untuk menghapus tanda baca |
| remove_add_space | required | remove_add_space adalah fitur untuk menghapus spasi berlebih |
| case_folding | required | case_folding adalah fitur untuk mungubah kata/kalimat menjadi huruf kecil atau huruf besar semua |
| remove_repeated_character | required | remove_repeated_character adalah fitur untuk menghapus karakter yang sama dan berulang |

Tabel 4.3 Hasil *Request API Text Preprocessing*

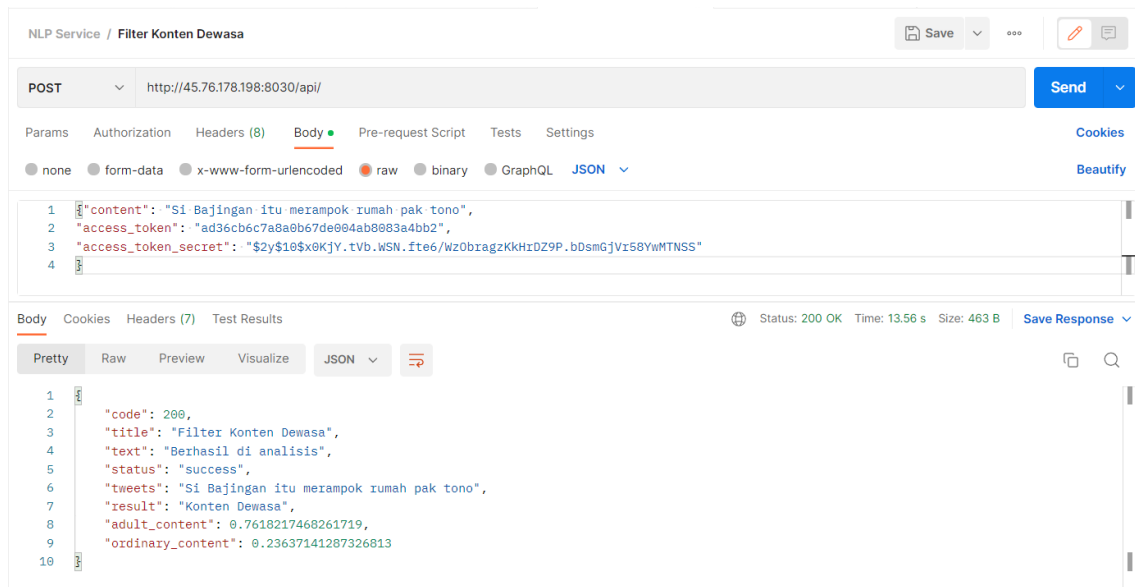
| Parameter | Deskripsi |
|---------------|---|
| <i>code</i> | 200 jika <i>token</i> valid 401 jika <i>token</i> tidak valid |
| <i>title</i> | <i>Text Preprocessing Service</i> |
| <i>status</i> | <i>success</i> jika <i>text</i> berhasil di <i>preprocessing</i> warning jika <i>token</i> tidak valid |
| <i>result</i> | <i>text</i> berhasil di <i>preprocessing</i> / tidak di <i>preprocessing</i> |



Gambar 4.30 Akses API Aplikasi *Text Preprocessing*

API Filter Konten Dewasa

Akses API filter konten dewasa seperti Gambar 4.31, *port* yang digunakan untuk API filter konten dewasa adalah 8030 dan *endpoint* yang digunakan adalah `http://45.76.178.198:8030/api/`. *Input* parameter yang digunakan untuk API filter konten dewasa seperti Tabel 4.4. Hasil *request* API seperti Tabel 4.5.



Gambar 4.31 Akses API Aplikasi Filter Konten Dewasa

Tabel 4.4 *Input* Parameter API Filter Konten Dewasa

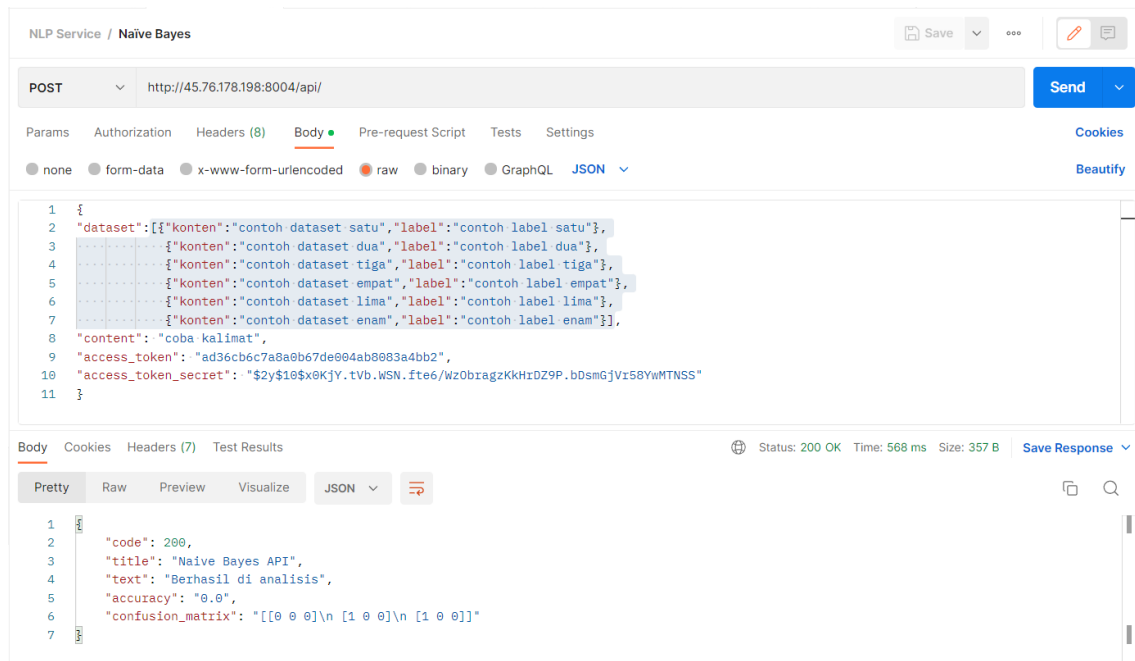
| Parameter | | Keterangan |
|----------------------------|-----------------|---|
| <i>content</i> | <i>required</i> | content adalah <i>text</i> yang akan dianalisis |
| <i>access_token</i> | <i>required</i> | <i>access_token</i> adalah <i>token</i> yang digunakan agar dapat mengakses aplikasi |
| <i>access_token_secret</i> | <i>required</i> | <i>access_token_secret</i> adalah <i>token</i> yang digunakan agar dapat mengakses aplikasi |

Tabel 4.5 Hasil *Request* API Filter Konten Dewasa

| Parameter | Deskripsi |
|-------------------------|---|
| <i>code</i> | 200 jika <i>token</i> valid 401 jika <i>token</i> tidak valid |
| <i>title</i> | <i>Filter Konten Dewasa Service</i> |
| <i>status</i> | <i>success</i> jika <i>token</i> valid <i>warning</i> jika <i>token</i> tidak valid |
| <i>result</i> | Konten Dewasa/Bukan Konten Desawa |
| <i>adult_content</i> | Hasil analisis perhitungan konten dewasa (Angka) |
| <i>ordinary_content</i> | Hasil analisis perhitungan bukan konten dewasa (Angka) |

API Naïve Bayes

Akses API Naïve Bayes seperti Gambar 4.32, *port* yang digunakan untuk API Naïve Bayes adalah 8004 dan *endpoint* yang digunakan adalah `http://45.76.178.198:8004/api/`. *Input* paramater yang digunakan untuk API Naïve Bayes seperti Tabel 4.6. Hasil *request* API seperti Tabel 4.7.



Gambar 4.32 Akses API aplikasi Naive Bayes

Tabel 4.6 Input Parameter API Naive Bayes

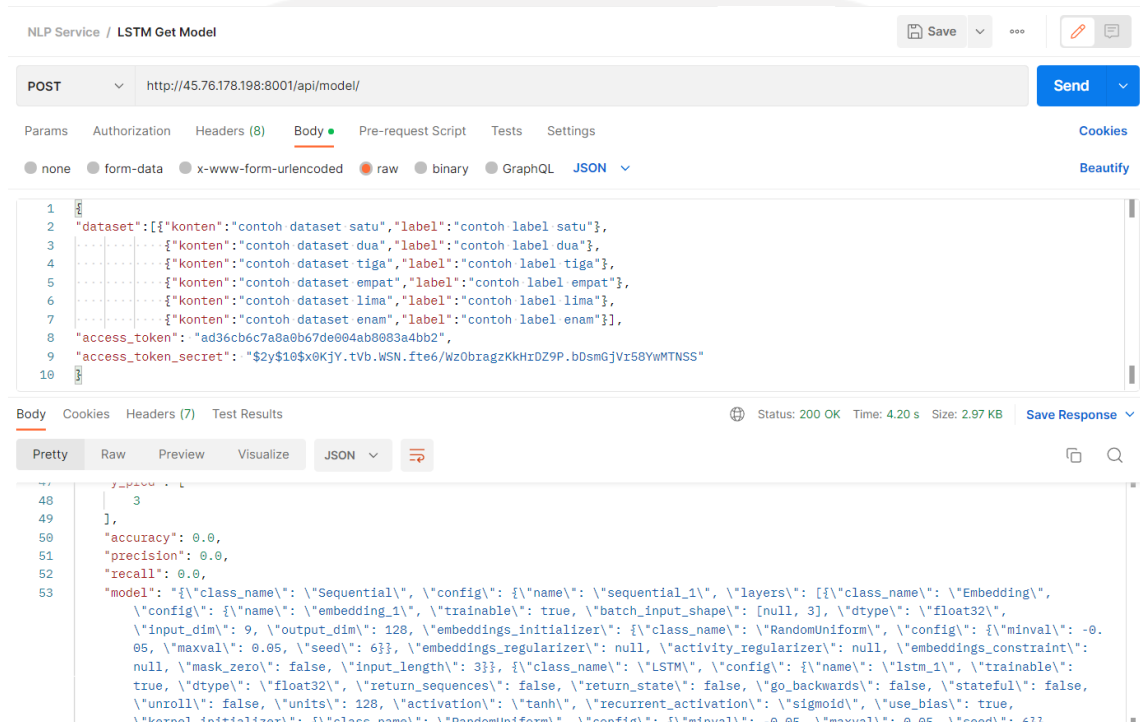
| Parameter | | Keterangan |
|---------------------|----------|---|
| dataset | required | data yang akan dibuat modelnya (dataset dikirim dalam bentuk JSON, banyak tools online untuk covert xls to JSON) Harus sesuai template data |
| content | required | content adalah <i>text</i> yang akan dianalisis |
| access_token | required | access_token adalah <i>token</i> yang digunakan agar dapat mengakses aplikasi |
| access_token_secret | required | access_token_secret adalah <i>token</i> yang digunakan agar dapat mengakses aplikasi |

Tabel 4.7 Hasil Request API Naive Bayes

| Parameter | Deskripsi |
|------------------|--|
| code | 200 jika <i>token</i> valid 401 jika <i>token</i> tidak valid |
| title | Naive Bayes API |
| status | <i>success</i> jika <i>text</i> berhasil dibuatkan modelnya <i>warning</i> jika <i>token</i> tidak valid |
| accuracy | Nilai <i>accuracy</i> berdasarkan train |
| confusion_matrix | nilai <i>Confusion Matrix</i> |

API LSTM

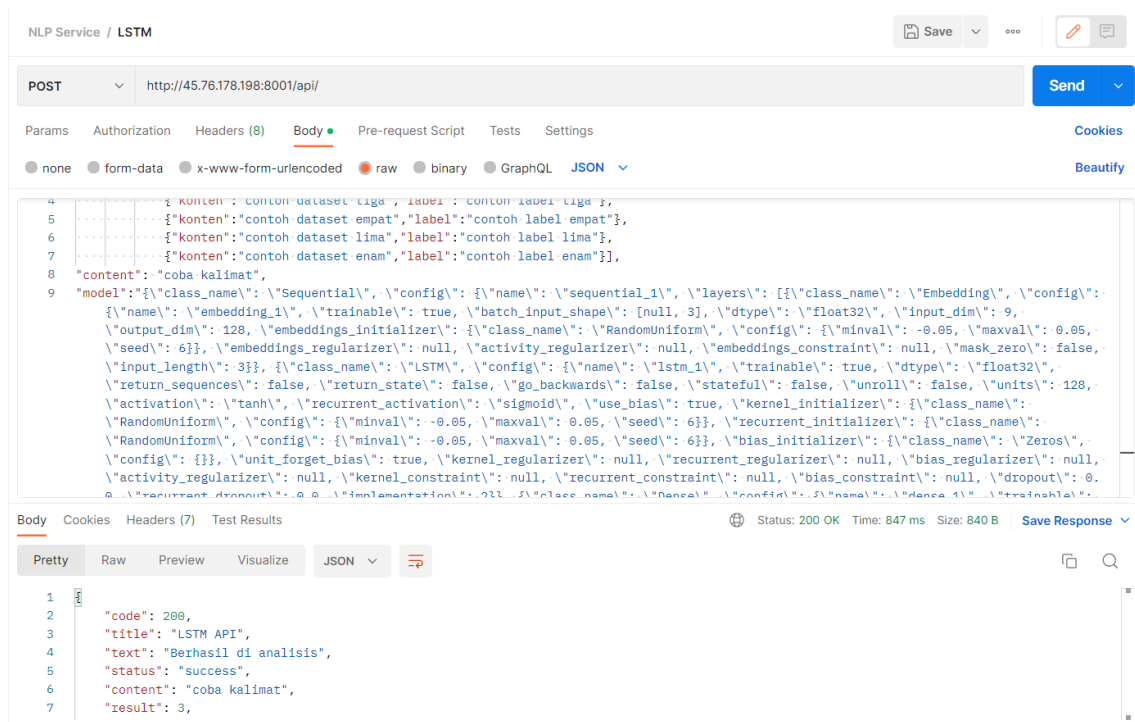
Untuk mengakses API LSTM harus menggunakan 2 *endpoint* yaitu <http://45.76.178.198:8001/api/model/> untuk membuat model lstm, contoh akses *endpoint* ini seperti Gambar 4.33 dan <http://45.76.178.198:8001/api/> untuk mengakses api LSTM contoh akses *endpoint* ini seperti Gambar 4.34. *Port* yang digunakan untuk API LSTM adalah 8001. *Input* paramater yang digunakan untuk API Naïve Bayes seperti Tabel 4.8. Hasil *request* API seperti Tabel 4.9.



Gambar 4.33 Akses API aplikasi LSTM Model

Tabel 4.8 *Input* Parameter API LSTM Model

| Parameter | | Keterangan |
|---------------------|----------|---|
| dataset | required | data yang akan dibuat modelnya (dataset dikirim dalam bentuk JSON, banyak tools online untuk covert xls to JSON) Harus sesuai template data |
| access_token | required | access_token adalah token yang digunakan agar dapat mengakses aplikasi |
| access_token_secret | required | access_token_secret adalah token yang digunakan agar dapat mengakses aplikasi |



Gambar 4.34 Akses API aplikasi LSTM

Tabel 4.9 Hasil *Request* API LSTM Model

| Parameter | Deskripsi |
|------------------|--|
| <i>code</i> | 200 jika <i>token</i> valid 401 jika <i>token</i> tidak valid |
| <i>title</i> | LSTM API |
| <i>status</i> | <i>success</i> jika <i>text</i> berhasil dibuatkan modelnya <i>warning</i> jika <i>token</i> tidak valid |
| <i>accuracy</i> | nilai <i>accuracy</i> berdasarkan <i>train</i> model (model.fit()) |
| <i>category</i> | label dari <i>dataset</i> |
| <i>history</i> | nilai <i>epoch</i> |
| <i>y_act</i> | label <i>test</i> |
| <i>y_pred</i> | nilai prediksi dari model.predict() |
| <i>precision</i> | nilai <i>precision</i> berdasarkan <i>train</i> model (model.fit()) |
| <i>recall</i> | nilai <i>recall</i> berdasarkan <i>train</i> model (model.fit()) |
| <i>model</i> | data model |

4.5.3 Service Oriented Architecture

Implementasi SOA sangat memudahkan dalam membangun dan mengembangkan aplikasi NLP *service*. SOA mampu mengintegrasikan aplikasi NLP *Service* yang di bangun dengan bahasa pemrograman yang berbeda.

Keunggulan tersebut dibuktikan dengan aplikasi *frontend* dibuat dengan bahasa typescript, aplikasi *backend* dibuat dengan bahasa PHP dan aplikasi LSTM dibuat dengan bahasa python. ketiga aplikasi tersebut bisa terintegrasi menggunakan API.

Setiap aplikasi terisolasi sehingga tidak mengganggu satu sama lain, ini sangat memudahkan saat melakukan *maintenance system* dan penambahan aplikasi.

4.6 Hasil Pengujian Sistem

Pada tahapan ini merupakan hasil pengujian sistem. Pengujian sistem dilakukan dengan metode *Black-box*. Pengujian dengan metode *Black-box* dilakukan terhadap seluruh fitur yang tersedia di dalam sistem. Setiap fitur yang ada terdapat beberapa aktivitas kasus uji. Adapun hasil dari pengujian *Black-box* adalah sebagai berikut :

4.6.1 Hasil Skenario Pengujian Login

Tabel 4.10 merupakan hasil pengujian terhadap fitur *login*, dari 2 aktivitas pengujian menggunakan *Black-box* semua aktivitas mendapatkan hasil yang sesuai.

Tabel 4.10 Hasil Pengujian Fitur *Login*

| Skenario | Kode | Aktivitas | Keterangan | Hasil Pengujian |
|----------------------------------|------|---|---|-----------------|
| Menampilkan halaman <i>login</i> | L-01 | Memasukkan NIM dan <i>password</i> yang sudah terdaftar di dalam sistem | Sistem akan mengecek ketersediaan <i>user</i> , jika user sudah terdaftar maka user akan dibawa ke halaman <i>dashboard</i> | Sesuai |
| | L-02 | Memasukkan <i>username</i> dan <i>password</i> yang tidak terdaftar di dalam sistem | Sistem akan menampilkan pesan “ <i>Username</i> tidak terdaftar” | Sesuai |

4.6.2 Hasil Skenario Pengujian *Token*

Tabel 4.11 merupakan hasil pengujian terhadap fitur *token*, dari 3 aktivitas pengujian menggunakan *Black-box* semua aktivitas mendapatkan hasil yang sesuai.

Tabel 4.11 Hasil Pengujian Fitur *Token*

| Skenario | Kode | Aktivitas | Keterangan | Hasil Pengujian |
|--------------------------------------|------|--|--|-----------------|
| Menampilkan halaman <i>Dashboard</i> | T-01 | <i>User</i> klik menu <i>Token</i> | Sistem akan menampilkan halaman <i>token</i> | Sesuai |
| Menampilkan halaman <i>Token</i> | T-02 | mengklik tombol ubah <i>token</i> saat belum memiliki <i>token</i> | Sistem akan memberikan <i>token</i> | Sesuai |
| | T-03 | mengklik tombol ubah <i>token</i> saat sudah memiliki <i>token</i> | Sistem akan memberikan <i>token</i> baru | Sesuai |

4.6.3 Hasil Skenario Pengujian *Text Preprocessing*

Tabel 4.12 dan Tabel 4.13 merupakan hasil pengujian terhadap fitur *text preprocessing*, dari 7 aktivitas pengujian menggunakan *Black-box* semua aktivitas mendapatkan hasil yang sesuai.

Tabel 4.12 Hasil Pengujian Fitur *Text Preprocessing*

| Skenario | Kode | Aktivitas | Keterangan | Hasil Pengujian |
|---|-------|---|--|-----------------|
| Menampilkan halaman <i>dashboard</i> | TP-01 | <i>User</i> klik menu <i>text preprocessing</i> | Sistem akan menampilkan halaman <i>text preprocessing</i> | Sesuai |
| Menampilkan halaman <i>text preprocessing</i> | TP-02 | Memasukkan <i>text</i> dan memilih fitur <i>preprocessing</i> lalu klik <i>submit</i> | Sistem akan memproses <i>text</i> dan menampilkan hasil <i>preprocessing</i> | Sesuai |

| Skenario | Kode | Aktivitas | Keterangan | Hasil Pengujian |
|----------|-------|---|---|-----------------|
| | TP-03 | <i>User</i> langsung klik tombol <i>submit</i> tanpa mengisi data apa pun | Sistem menolak permintaan dan menampilkan pesan “Fitur harus diisi !” | Sesuai |
| | TP-04 | <i>User</i> tidak memilih semua fitur <i>preprocessing</i> | Sistem menolak permintaan dan menampilkan pesan “Fitur harus diisi !” | Sesuai |

Tabel 4.13 Hasil Pengujian Fitur API *Text Preprocessing*

| Skenario | Kode | Aktivitas | Keterangan | Hasil Pengujian |
|-------------------------------------|-------|--|---|-----------------|
| <i>Akses API text preprocessing</i> | TP-05 | <i>User</i> mengirimkan data json sesuai dengan format | Sistem akan mengirimkan <i>file</i> JSON hasil proses <i>text preprocessing</i> | Sesuai |
| | TP-06 | <i>User</i> tidak mengirimkan data json sesuai dengan format | Sistem akan menolak | Sesuai |
| | TP-07 | <i>access_token</i> dan <i>access_token_secret</i> tidak benar | Sistem akan menolak | Sesuai |

4.6.4 Hasil Skenario Pengujian Filter Konten Dewasa

Tabel 4.14 dan Tabel 4.15 merupakan hasil pengujian terhadap fitur filter konten dewasa, dari 6 aktivitas pengujian menggunakan *Black-box* semua aktivitas mendapatkan hasil yang sesuai.

Tabel 4.14 Hasil Pengujian Filter Konten Dewasa

| Skenario | Kode | Aktivitas | Keterangan | Hasil Pengujian |
|--|--------|---|--|-----------------|
| Menampilkan halaman <i>dashboard</i> | FKD-01 | <i>User</i> klik menu filter konten dewasa | Sistem akan menampilkan halaman filter konten dewasa | Sesuai |
| Menampilkan halaman filter konten dewasa | FKD-02 | Memasukkan <i>text</i> dan klik <i>submit</i> | Sistem akan memproses <i>text</i> dan menampilkan hasil filter konten dewasa | Sesuai |
| | FKD-03 | <i>User</i> langsung klik tombol <i>submit</i> tanpa mengisi data apa pun | Sistem menolak permintaan dan menampilkan pesan “Fitur harus diisi !” | Sesuai |

Tabel 4.15 Hasil Pengujian API Filter Konten Dewasa

| Skenario | Kode | Aktivitas | Keterangan | Hasil Pengujian |
|--------------------------------|--------|--|--|-----------------|
| Akses API filter konten dewasa | FKD-04 | <i>User</i> tidak mengirimkan data json sesuai dengan format | Sistem akan menolak | Sesuai |
| | FKD-05 | <i>access_token</i> dan <i>access_token_secret</i> tidak benar | Sistem akan menolak | Sesuai |
| | FKD-06 | <i>User</i> mengirim data sesuai format | Sistem akan mengirimkan <i>file</i> JSON hasil proses filter konten dewasa | Sesuai |

4.6.5 Hasil Skenario Pengujian Sentimen Analisis

Tabel 4.16 merupakan hasil pengujian terhadap fitur filter konten dewasa, dari 3 aktivitas pengujian menggunakan *Black-box* semua aktivitas mendapatkan hasil yang sesuai.

Tabel 4.16 Hasil Pengujian Sentimen Analisis

| Skenario | Kode | Aktivitas | Keterangan | Hasil Pengujian |
|---------------------------------------|-------|--|---|-----------------|
| Menampilkan halaman <i>dashboard</i> | SA-01 | User klik menu sentimen analisis | Sistem akan menampilkan halaman sentimen analisis | Sesuai |
| Menampilkan halaman Sentimen Analisis | SA-02 | Pengguna mengunggah <i>file dataset</i> yang sesuai format. Pengguna mengisi dan mengirim <i>text</i> yang akan di proses oleh fitur sistem analisis | Sistem menerima permintaan dan menampilkan analisis menggunakan fitur sentimen analisis | Sesuai |
| | SA-03 | Pengguna mengunggah <i>file dataset</i> yang tidak sesuai format. Pengguna mengisi dan mengirim <i>text</i> yang akan di proses oleh fitur sistem analisis | Sistem tidak akan menerima dan akan menampilkan pesan <i>error</i> . | Sesuai |

الجامعة الإسلامية
الاستاذ الدكتور

4.6.6 Hasil Skenario Pengujian K-means

Tabel 4.17 merupakan hasil pengujian terhadap fitur K-means, dari 3 aktivitas pengujian menggunakan *Black-box* semua aktivitas mendapatkan hasil yang sesuai.

Tabel 4.17 Hasil Pengujian K-means

| Skenario | Kode | Aktivitas | Keterangan | Hasil Pengujian |
|--------------------------------------|------|--|---|-----------------|
| Menampilkan halaman <i>Dashboard</i> | K-01 | User klik menu K_means | Sistem akan menampilkan halaman <i>token</i> | Sesuai |
| Menampilkan halaman K-means | K-02 | Pengguna mengunggah <i>file dataset</i> yang sesuai format. Pengguna mengisi dan mengirim jumlah <i>cluster</i> dan random <i>state</i> yang akan di proses oleh fitur K-means | Sistem menerima permintaan dan menampilkan analisis menggunakan fitur K-means | Sesuai |
| | K-03 | Pengguna mengunggah <i>file dataset</i> yang tidak sesuai format. Pengguna mengisi dan mengirim jumlah <i>cluster</i> dan random <i>state</i> yang akan di proses oleh fitur K-means | Sistem tidak akan menerima dan akan menampilkan pesan <i>error</i> . | Sesuai |

4.6.7 Hasil Skenario Pengujian LSTM

Tabel 4.18 dan Tabel 4.19 merupakan hasil pengujian terhadap fitur LSTM, dari 7 aktivitas pengujian menggunakan *Black-box* semua aktivitas mendapatkan hasil yang sesuai.

Tabel 4.18 Hasil Pengujian LSTM

| Skenario | Kode | Aktivitas | Keterangan | Hasil Pengujian |
|--------------------------------------|------|---|---|-----------------|
| Menampilkan halaman <i>dashboard</i> | L-01 | <i>User</i> klik menu LSTM | Sistem akan menampilkan halaman LSTM | Sesuai |
| Menampilkan halaman LSTM | L-02 | Pengguna mengunggah <i>file dataset</i> yang sesuai format. | Sistem akan membuat model dari <i>dataset</i> | Sesuai |
| | L-03 | Pengguna mengunggah <i>file dataset</i> yang tidak sesuai format. | Sistem akan menampilkan pesan <i>error</i> | Sesuai |
| | L-04 | Pengguna mengisi dan mengirim <i>text</i> yang akan di proses oleh LSTM | Sistem akan menampilkan hasil proses LSTM | Sesuai |

Tabel 4.19 Hasil Pengujian API LSTM

| Skenario | Kode | Aktivitas | Keterangan | Hasil Pengujian |
|----------------|------|--|--|--|
| Akses API LSTM | L-05 | <i>User</i> tidak mengirimkan data json sesuai dengan format | Sistem akan menolak | <i>User</i> tidak mengirimkan data json sesuai dengan format |
| | L-06 | <i>access_token</i> dan <i>access_token_secret</i> tidak benar | Sistem akan menolak | <i>access_token</i> dan <i>access_token_secret</i> tidak benar |
| | L-07 | <i>User</i> mengirim data sesuai format | Sistem akan mengirimkan <i>file</i> JSON hasil proses LSTM | <i>User</i> mengirim data sesuai format |

4.6.8 Hasil Skenario Pengujian Naïve Bayes

Tabel 4.20 dan

Tabel 4.21 merupakan hasil pengujian terhadap fitur naïve bayes, dari semua aktivitas pengujian menggunakan *Black-box* semua aktivitas mendapatkan hasil yang sesuai.

Tabel 4.20 Hasil Pengujian Fitur Naïve Bayes

| Skenario | Kode | Aktivitas | Keterangan | Hasil Pengujian |
|--------------------------------------|-------|---|---|-----------------|
| Menampilkan halaman <i>Dashboard</i> | NB-01 | <i>User</i> klik menu Naïve Bayes | Sistem akan menampilkan halaman Naïve Bayes | Sesuai |
| Menampilkan halaman u Naïve Bayes | NB-02 | Pengguna mengunggah <i>file dataset</i> yang sesuai format. | Sistem akan menampilkan hasil proses Naïve Bayes dari dataset | Sesuai |

Tabel 4.21 Hasil Pengujian Fitur API Naïve Bayes

| Skenario | Kode | Aktivitas | Keterangan | Hasil Pengujian |
|-----------------------|-------|--|---|-----------------|
| Akses API Naïve Bayes | NB-03 | <i>User</i> tidak mengirimkan data json sesuai dengan format | Sistem akan menolak | Sesuai |
| | NB-04 | <i>access_token</i> dan <i>access_token_secret</i> tidak benar | Sistem akan menolak | Sesuai |
| | NB-05 | <i>User</i> mengirim data sesuai format | Sistem akan mengirimkan <i>file</i> JSON hasil proses Naïve Bayes | |

BAB 5

Kesimpulan dan Saran

5.1 Kesimpulan

Penelitian ini mendapat kesimpulan pembuatan *Context-Aware* untuk NLP Service berhasil diimplementasikan. Adapun Langkah-langkah yang dilakukan untuk melakukan implementasi *Context-Aware* untuk NLP Service adalah pengumpulan data, *refactoring source code*, *containerization*, pemodelan *Context-Aware*, implementasi sistem, pengujian sistem.

Pengumpulan data pada penelitian ini adalah mencari *source code* dengan bahasa python untuk dilakukan implementasi. *Refactoring source code* pada penelitian ini adalah mengondisikan *source code* yang didapat agar bisa di *containerization* dengan baik. Tantangan pada tahap ini adalah penggunaan *library* yang telah usang, ini menghambat saat *containerization* karena *library* yang telah usang tidak bisa di-*install* lagi atau kode dasarnya tidak *support* dengan *library* terbaru.

Containerization pada tahap ini adalah membuat *source code* menjadi *docker container*. Hambatan pada tahap ini tentu jika *library* dan kode program tidak *support*. Untuk mengatasi masalah ini bisa menurunkan atau memasang *library* sesuai versi pada *source code* sebelum di *refactoring* atau dengan melakukan *refactoring code* kembali.

Pemodelan *Context-Aware* adalah menentukan jenis sebuah *file* termasuk jenis *categorical* atau *uncategorical*. Jika jumlah data unik pada fitur kurang atau sama dengan tiga (≤ 3) maka *file* tersebut dikategorikan *categorical* dan jika lebih dari tiga maka dikategorikan *uncategorical*. Kondisi ini dipilih karena semakin sedikit data unik pada fitur semakin mudah menentukan kategori pada setiap baris datanya.

Implementasi Sistem pada tahap ini adalah melakukan integrasi pada semua *docker container* agar setiap aplikasi bisa saling terhubung dan berkomunikasi. SOA juga sangat membantu dalam pengembangan terutama saat penambah aplikasi atau algoritma tidak akan mengganggu aplikasi yang sudah ada.

Pengujian Sistem pada tahap ini adalah melakukan pengujian fungsionalitas sistem menggunakan metode *black box testing*. Aplikasi *Context-Aware* dapat mengenali file apakah *supervised* atau *unsupervised* seperti pada Gambar 4.19. Dari Gambar 4.20 bisa disimpulkan secara keseluruhan LSTM ataupun Naïve bayes memiliki akurasi yang sama. Namun pada pengujian yang hasil akurasinya berbeda LSTM lebih unggul dibandingkan *naïve bayes*.

5.2 Saran

Saran untuk penelitian selanjutnya adalah melakukan penambahan algoritma agar rekomendasi yang diberikan lebih banyak. Serta melakukan komparasi algoritma yang sama dengan bahasa pemrograman yang berbeda. Sehingga aplikasi *context-aware* selain memberikan rekomendasi algoritma juga memberikan rekomendasi bahasa pemrograman yang digunakan. Karena dibangun dengan menggunakan arsitektur SOA, NLP service ini bisa di kembangkan lagi dengan menggunakan arsitektur *microservice*.



Daftar Pustaka

- Abdollahi Vayghan, L., Saied, M. A., Toeroe, M., & Khendek, F. (2018). Deploying Microservice Based Applications with Kubernetes: Experiments and Lessons Learned. *IEEE International Conference on Cloud Computing, CLOUD, 2018-July*, 970–973. <https://doi.org/10.1109/CLOUD.2018.00148>
- Dewabharata, A., Wen, D. M. H., & Chou, S. Y. (2013). An activity ontology for context-aware health promotion application. *Proceedings - International Computer Software and Applications Conference*, 421–426. <https://doi.org/10.1109/COMPSACW.2013.90>
- Eryigit, G. (2015). *ITU Turkish NLP Web Service*. 1–4. <https://doi.org/10.3115/v1/e14-2001>
- Flores-Rodriguez, C., & Aguilar-Alonso, I. (2020). Initial results of Software Development Models based on Service Oriented Architecture. *Proceedings - IEEE 2020 2nd International Conference on Advances in Computing, Communication Control and Networking, ICACCCN 2020*, 924–928. <https://doi.org/10.1109/ICACCCN51052.2020.9362928>
- Hamza, Z. A., & Hammad, M. (2019). *Web and Mobile Applications' Testing using Black and White Box approaches*. <https://doi.org/10.1049/cp.2019.0210>
- Hidayatullah, A. F. (2019). *Sentiment Analysis Example*. Github. <https://github.com/fathanick/Sentiment-Analysis-Example>
- Hidayatullah, A. F. (2021). *NLP-Course-UII*. Github. <https://github.com/fathanick/NLP-Course-UII/tree/master/Preprocessing>
- Hidayatullah, A. F., Hakim, A. M., & Sembada, A. A. (2019). Adult content classification on Indonesian tweets using LSTM neural network. *2019 International Conference on Advanced Computer Science and Information Systems, ICACISIS 2019*, 235–240. <https://doi.org/10.1109/ICACISIS47736.2019.8979982>
- Komoda, N. (n.d.). *Service Oriented Architecture (SOA) in Industrial Systems*.
- Kousiouris, G., Tsarsitalidis, S., Psomakelis, E., Koloniaris, S., Bardaki, C., Tserpes, K., Nikolaidou, M., & Anagnostopoulos, D. (2019). A microservice-based framework for integrating IoT management platforms, semantic and AI services for supply chain management. *ICT Express*, 5(2), 141–145. <https://doi.org/10.1016/j.ict.2019.04.002>

- Lee, A. (2019). *Why NLP is important and it'll be the future — our future*.
<https://Towardsdatascience.Com/>. <https://towardsdatascience.com/why-nlp-is-important-and-itll-be-the-future-our-future-59d7b1600dda>
- Lin, B., Chen, Y., Chen, X., & Yu, Y. (2012). Comparison between JSON and XML in Applications Based on AJAX. *Proceedings - 2012 International Conference on Computer Science and Service System, CSSS 2012*, 1174–1177.
<https://doi.org/10.1109/CSSS.2012.297>
- Modak, A., Prof. S.D.Chaudhary, Prof. P.S.Paygude, & Prof. S.R.Idate. (2018). *Techniques to Secure Data on Cloud: Docker Swarm or Kubernetes?* Proceedings of the International Conference on Inventive Communication and Computational Technologies : ICICCT 2018 : 20-21, April 2018.
<https://doi.org/10.1109/ICICCT.2018.8473104>
- Monteiro, E. L., & Silva, P. C. da. (2015). *Risk Management Lifecycle Implementation Services in SOA*. 2015 12th International Conference on Information Technology - New Generations. <https://doi.org/10.1109/ITNG.2015.137>
- Muddinagiri, R., Ambavane, S., & Bayas, S. (2020). *SELF-HOSTED KUBERNETES: DEPLOYING DOCKER CONTAINERS LOCALLY WITH MINIKUBE*. 2019 International Conference on Innovative Trends and Advances in Engineering and Technology (ICITAET). <https://doi.org/10.1109/ICITAET47105.2019.9170208>
- Muhardany, Suharjito, Sukmadhani, A. A., & Gunawan, E. (2020). *Development of E-Commerce Approach to Service Oriented Architecture Using SOMA Methodology*. 2020 International Conference on Information Management and Technology (ICIMTech). <https://doi.org/10.1109/ICIMTech50083.2020.9211161>
- Nagpal, A., & Gabrani, G. (2019). Python for Data Analytics, Scientific and Technical Applications. *Proceedings - 2019 Amity International Conference on Artificial Intelligence, AICAI 2019*, 140–145. <https://doi.org/10.1109/AICAI.2019.8701341>
- Osipov, G., Smirnov, I., Tikhomirov, I., & Vybornova, O. (2012). *Technologies for Semantic Analysis of Scientific Publications*. <https://doi.org/10.1109/IS.2012.6335191>
- Ouda, A. H., Allison, D. S., & Capretz, M. A. M. (2010). Security protocols in service-oriented architecture. *Proceedings - 2010 6th World Congress on Services, Services-I 2010*, 185–186. <https://doi.org/10.1109/SERVICES.2010.44>
- Papulovskaya, N., Izotov, I., & Orekhov, P. (2019, July 28). Implementing IoT Systems in Service-Oriented Architecture. *2019 Ural Symposium on Biomedical Engineering*,

- Radioelectronics and Information Technology (USBREIT)*.
<https://doi.org/10.1109/USBREIT.2019.8736593>
- Parveen, T., & Tilley, S. (2008). *A Research Agenda for Testing SOA-Based Systems*.
- Perdana, G., & Ashari, A. (2019). Penggunaan Metode Ontology Untuk Perancangan Purwarupa Sistem Smart Home Berbasis Context Aware. *IJEIS (Indonesian Journal of Electronics and Instrumentation Systems)*, 9(2), 119.
<https://doi.org/10.22146/ijeis.39042>
- Perera, C., Zaslavsky, A., Christen, P., & Georgakopoulos, D. (2014). Context aware computing for the internet of things: A survey. *IEEE Communications Surveys and Tutorials*, 16(1), 414–454. <https://doi.org/10.1109/SURV.2013.042313.00197>
- Prashant, P., Tickoo, A., Sharma, S., & Jamil, J. (2019). Optimization of cost to calculate the release time in software reliability using python. *Proceedings of the 9th International Conference On Cloud Computing, Data Science and Engineering, Confluence 2019*, 470–474. <https://doi.org/10.1109/CONFLUENCE.2019.8776620>
- Quaranta, L., Calefato, F., & Lanubile, F. (2021). KGTorrent: A dataset of python jupyter notebooks from kaggle. *Proceedings - 2021 IEEE/ACM 18th International Conference on Mining Software Repositories, MSR 2021*, 550–554.
<https://doi.org/10.1109/MSR52588.2021.00072>
- Santaholma, M. (2007). *Grammar sharing techniques for rule-based multilingual NLP systems*. <https://archive-ouverte.unige.ch/unige:3455>
- Santoro, C., Messina, F., D'Urso, F., & Santoro, F. F. (2018). Wale: A dockerfile-based approach to deduplicate shared libraries in docker containers. *Proceedings - IEEE 16th International Conference on Dependable, Autonomic and Secure Computing, IEEE 16th International Conference on Pervasive Intelligence and Computing, IEEE 4th International Conference on Big Data Intelligence and Computing and IEEE 3, Vmm*, 776–784.
<https://doi.org/10.1109/DASC/PiCom/DataCom/CyberSciTec.2018.00135>
- Sembada, A. A. (2020a). *Long Short-Term Memory (LSTM) API*. Github.
<https://github.com/azizsembada/lstm-api>
- Sembada, A. A. (2020b). *Text Preprocessing*. Github. <https://github.com/azizsembada/text-preprocessing-api>
- Siddiqui, I. F., & Babar, N. (2020). NLP Based Service Recommendation System. *International Conference on Internet, January*, 321–324.

https://www.researchgate.net/publication/338711391_NLP_BASED_SERVICE_RECOMMENDATION_SYSTEM

- Sudjana, A. R. L., Jm, H. O., Suprpto, F., Husin, E., & Fajar, A. N. (2019, November 1). Design of Travel Service System in Indonesia Based on Service-Oriented Architecture. *2019 7th International Conference on Cyber and IT Service Management, CITSM 2019*. <https://doi.org/10.1109/CITSM47753.2019.8965392>
- Trofin, R. S., Chiru, C., Vizitiu, C., Dinculescu, A., Vizitiu, R., & Nistorescu, A. (2019). Detection of astronauts' speech and language disorder signs during space missions using natural language processing techniques. *2019 7th E-Health and Bioengineering Conference, EHB 2019*, 1–4. <https://doi.org/10.1109/EHB47216.2019.8969950>
- van den Berg, T., Siegel, B., & Cramp, A. (2017). Containerization of high level architecture-based simulations: A case study. *Journal of Defense Modeling and Simulation*, *14*(2), 115–138. <https://doi.org/10.1177/1548512916662365>
- Vanderplas, J. (2018). *Python Data Science Handbook*. <https://Github.Com/>.
<https://github.com/jakevdp/PythonDataScienceHandbook>
- Voita, E., Serdyukov, P., Sennrich, R., & Titov, I. (2018). Context-aware neural machine translation learns anaphora resolution. *ArXiv*.
- Wang, Y., Sun, Y., & Chen, Y. (2019). Design and Research of Intelligent Tutor System Based on Natural Language Processing. *2019 IEEE International Conference on Computer Science and Educational Informatization, CSEI 2019*, 33–36.
<https://doi.org/10.1109/CSEI47661.2019.8939031>
- Wigmore, I. (2016, May). *Context Awareness*. <https://Www.Techtarget.Com/>.
<https://www.techtarget.com/whatis/definition/context-awareness>
- Yu, Q., & Yang, W. (2019). The Analysis and Design of System of Experimental Consumables Based on Django and QR code. *Proceedings - 2019 2nd International Conference on Safety Produce Informatization, IICSPI 2019*, 137–141.
<https://doi.org/10.1109/IICSPI48186.2019.9095914>