

3. ANALISA DAN DESAIN SISTEM

Pada bab ini akan dibahas mengenai desain dan rancangan aplikasi *sentiment analysis*. Bagian pertama dari bab ini akan menjelaskan dan menganalisis data yang digunakan dalam penelitian. Kemudian bagian selanjutnya akan membahas mengenai kinerja dari sistem yang akan dipaparkan dalam bentuk *use case* diagram dan *activity* diagram beserta dengan penjelasannya mengenai alur kerja dan aktivitas yang akan dilakukan. Bagian terakhir dari bab ini akan menjelaskan mengenai struktur penyimpanan data di *database*.

3.1 Analisa Permasalahan

Excellence in Learning and Teaching Center (ELTC) merupakan unit yang menangani pengembangan pembelajaran dan pengajaran di Universitas Kristen Petra. Dalam menjalankan fungsinya, ELTC menerima masukan dari mahasiswa terkait proses pembelajaran di Universitas Kristen Petra. Hal ini penting untuk mengetahui kepuasan mahasiswa terkait proses pembelajaran di Universitas Kristen Petra. Masukan dari mahasiswa didapatkan dari kuesioner berisi beberapa pertanyaan yang dibagikan dan diisi oleh mahasiswa. Hasil dari kuesioner tersebut kemudian akan digunakan untuk melakukan evaluasi mengenai proses belajar dan mengajar di Universitas Kristen Petra. Selama ini proses pengolahan informasi tersebut dilakukan secara manual yaitu setiap kuesioner dibaca oleh tenaga manusia satu persatu. Dalam proses pengolahan hasil kuesioner yang berjumlah besar tersebut, ditemukan beberapa kesulitan yaitu dibutuhkan waktu yang lama dalam membaca dan menilai jawaban responden karena proses ini dilakukan secara manual. Maka dari itu dibutuhkan sebuah aplikasi yang dapat membantu dalam pengolahan kuesioner menjadi informasi yang berguna dalam waktu yang lebih cepat dan tidak membutuhkan banyak sumber daya.

3.2 Analisa Kebutuhan

Berdasarkan permasalahan yang dihadapi oleh ELTC tersebut, maka dapat disimpulkan bahwa dibutuhkan sebuah aplikasi yang dapat membantu dalam proses pengolahan data hasil kuesioner untuk mendapatkan informasi dalam waktu yang cepat. Beberapa fitur dari aplikasi yang dapat membantu ELTC adalah sebagai berikut

1. Aplikasi yang dapat menentukan masukan mahasiswa sebagai komentar positif atau negatif untuk mengetahui tanggapan mahasiswa terkait sistem pembelajaran yang sedang berjalan.
2. Aplikasi yang dapat mengelompokkan komentar berdasarkan aspeknya untuk memudahkan pihak terkait melakukan tindak lanjut.

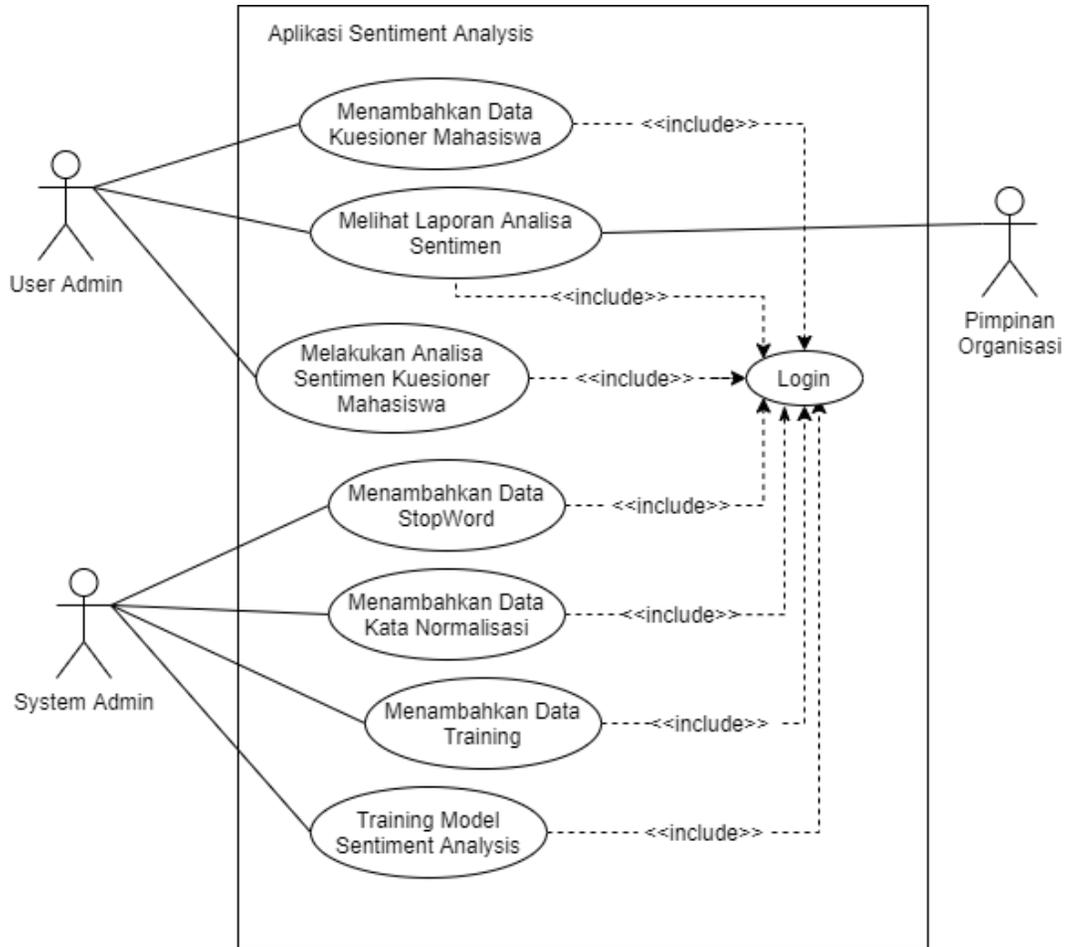
3.3 Desain Sistem

Sistem yang akan dibangun memiliki dua fungsi utama yaitu melihat laporan sentimen mahasiswa dan melakukan analisa sentimen. Pada bagian ini akan dijelaskan penggambaran penggunaan dan alur proses program bekerja.

3.3.1 *Use Case Diagram*

Use Case Diagram menggambarkan aktivitas yang dapat digunakan oleh pengguna yang terlibat dalam sistem aplikasi ini yaitu, *user administrator*, *system administrator*, dan pimpinan organisasi seperti yang digambarkan pada Gambar 3.1. Setiap aktor memiliki fungsi dan aktivitasnya masing-masing ketika menggunakan aplikasi. *User Administrator* melakukan aktivitas yang berkaitan dengan penggunaan aplikasi untuk proses pengolahan data kuesioner mahasiswa seperti penambahan data kuesioner mahasiswa, melakukan klasifikasi topik dan sentimen terhadap data kuesioner mahasiswa, dan melihat laporan hasil klasifikasi data kuesioner mahasiswa. *System Administrator* bertugas untuk melakukan aktivitas yang berkaitan dengan pemeliharaan sistem dan model klasifikasi. Aktivitas yang melibatkan aktor *system administrator* adalah melakukan penambahan pada *stopword*, menambahkan data untuk kata-kata normalisasi,

menambahkan data *training*, dan melakukan *training* pada data untuk membuat model. Kemudian terdapat aktor Pimpinan Organisasi yang dapat melakukan aktivitas melihat laporan hasil klasifikasi data kuesioner mahasiswa.



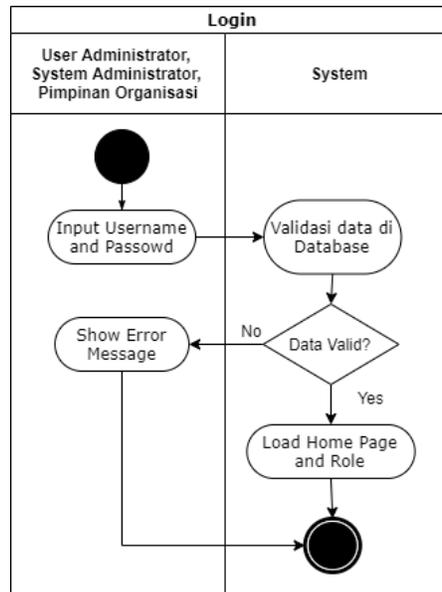
Gambar 3.1 Use Case Diagram

3.3.2 Activity Diagram

Activity Diagram merupakan penggambaran yang lebih detail dari aktivitas yang ada pada *use case diagram*. Penggambaran ini akan dijelaskan dalam bentuk alur kerja bagi setiap *use case* dan aktivitas yang dilakukan oleh aktor yang terlibat dalam sistem.

a. Login

Activity Diagram untuk login ditampilkan pada Gambar 3.2. Aktivitas login merupakan aktivitas awal yang wajib dilakukan oleh setiap aktor sebelum menggunakan aplikasi. Aktor diharuskan untuk memasukkan *username* dan *password* yang kemudian akan divalidasi oleh sistem tentang kecocokan data yang dimasukkan dan yang tercatat di sistem. Jika data yang dimasukkan sesuai, maka aktor akan diarahkan ke *homepage*.

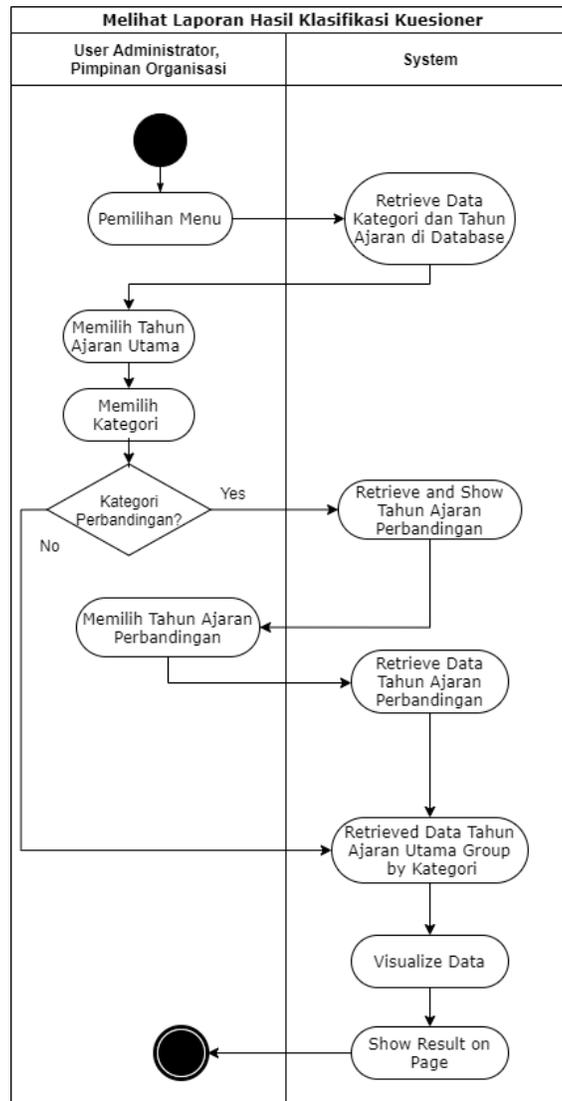


Gambar 3.2 *Activity Diagram Login*

b. Melihat Laporan Hasil Klasifikasi Kuesioner

Activity Melihat Laporan Hasil Klasifikasi Kuesioner ini dilakukan oleh aktor *user administrator* dan pimpinan organisasi. *Activity* ini dimulai ketika admin memilih untuk melihat laporan dari data yang sudah ada di *database*. Kemudian aktor diminta untuk memilih semester yang akan dilihat laporannya. Terdapat pula kategori yang dapat dipilih admin untuk melihat laporan kepuasan mahasiswa pada semester tertentu, kategori tersebut adalah berdasarkan angkatan, fakultas, gender, keseluruhan, dan perbandingan dengan semester lain. Jika kategori yang dipilih adalah perbandingan, maka *user* akan diminta untuk memilih tahun ajaran yang akan dibandingkan dan sistem akan mengambil data dari *database* sesuai tahun ajaran pembandingan

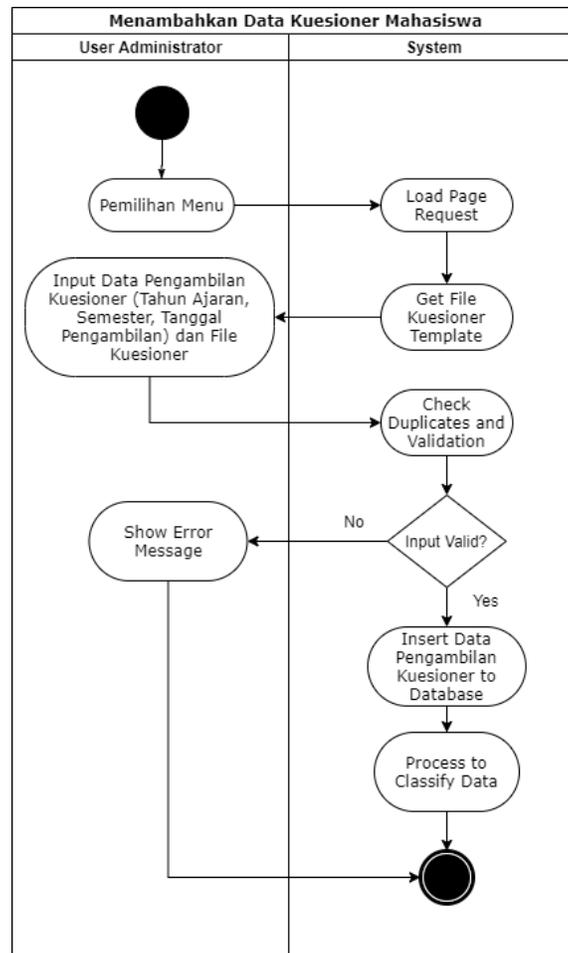
tersebut. Setelah proses tersebut sistem akan melakukan pengambilan data di *database* sesuai semester yang dipilih di awal. Proses pengambilan data semester yang dipilih di awal ini dilakukan oleh semua kategori yang dipilih. Setelah semua data telah terkumpul, data akan dikelompokkan dan divisualisasikan untuk ditampilkan pada halaman. Proses tersebut digambarkan dalam *activity diagram* pada Gambar 3.3.



Gambar 3.3 *Activity Diagram* Melihat Laporan Hasil Klasifikasi Kuesioner

c. Menambahkan Data Kuesioner Mahasiswa

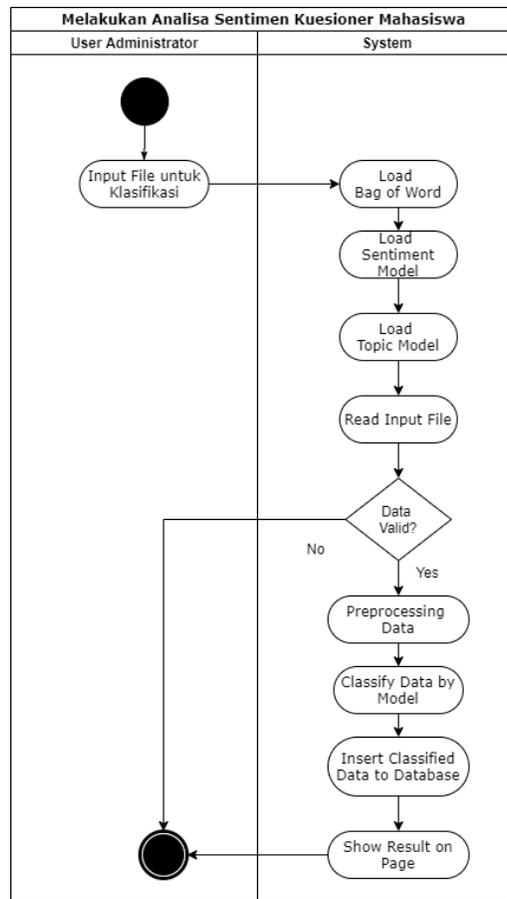
Activity untuk menambahkan data kuesioner mahasiswa merupakan aktivitas yang dilakukan oleh *user administrator*. Aktivitas ini dimulai dengan aktor memilih menu untuk menambahkan data kuesioner baru. Kemudian sistem akan memberikan contoh *file* kuesioner yang dapat diproses. Aktor akan mengisi data pengambilan kuesioner dan menambahkan *file* kuesioner mahasiswa yang akan diproses. Sistem akan melakukan pengecekan *file* dan memberikan peringatan jika terjadi kesalahan. Jika *file* yang diberikan telah sesuai maka akan diproses untuk klasifikasi dan penyimpanan ke *database*. Aktivitas ini digambarkan pada sebuah *activity diagram* pada Gambar 3.4.



Gambar 3.4 *Activity Diagram* Menambahkan Data Kuesioner Mahasiswa

d. Melakukan Klasifikasi Kuesioner Mahasiswa

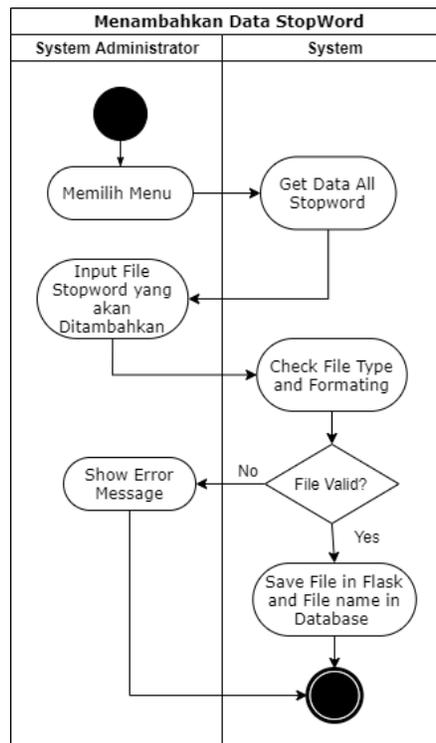
Activity melakukan klasifikasi kuesioner mahasiswa dimulai ketika *user administrator* menambahkan data kuesioner yang ingin diklasifikasikan dan sesuai dengan ketentuan *file*. Sistem akan mengambil beberapa file yang menunjang proses klasifikasi seperti *file Bag of Word*, *file Sentiment Model*, dan *file Topic Model* yang telah tersimpan di server. Kemudian admin diminta untuk memilih semester pertama yang akan dibandingkan dengan semester kedua yang dipilih. Setelah semester yang diinginkan telah dipilih, data pada dua semester tersebut akan diambil di *database* dan ditampilkan dalam grafik untuk menampilkan perbandingannya. Aktivitas ini digambarkan pada sebuah *activity diagram* pada Gambar 3.5.



Gambar 3.5 Activity Diagram Perbandingan Dua Semester

e. Menambahkan *Data Stopword*

Activity menambahkan *data stopwords* dilakukan oleh *system administrator*. Tujuan dari aktivitas ini adalah menambahkan *stopword* yang akan digunakan pada proses *preprocessing*. Aktivitas dimulai ketika aktor memilih menu dan mendapatkan seluruh *data stopwords* di sistem. Aktor akan menambahkan file baru untuk diperiksa dan ditambahkan pada *data stopwords*. *File* tersebut akan diperiksa dan disimpan jika sesuai dengan format yang telah ditentukan. Aktivitas ini digambarkan pada sebuah *activity diagram* pada Gambar 3.6.

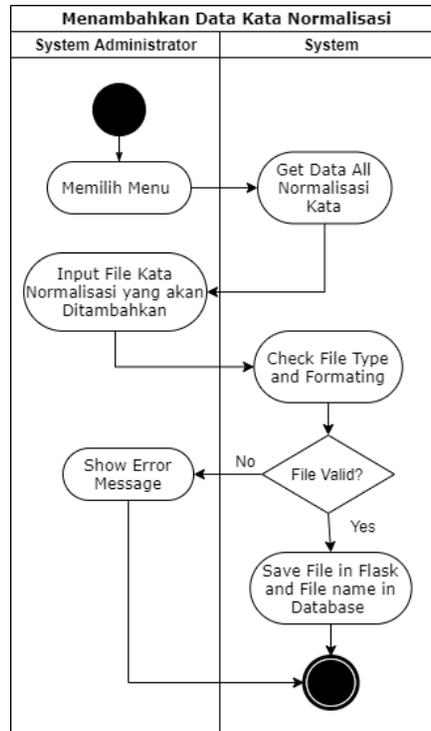


Gambar 3.6 *Activity Diagram* Menambahkan Data Stopword

f. Menambahkan Data Kata Normalisasi

Activity menambahkan data kata normalisasi dilakukan oleh *system administrator*. Tujuan dari aktivitas ini adalah menambahkan kata-kata yang akan dinormalisasi berupa kata singkatan dan artinya. Kata normalisasi akan digunakan pada proses *preprocessing*. Aktivitas dimulai ketika aktor memilih menu dan mendapatkan seluruh data kata normalisasi di sistem. Aktor akan

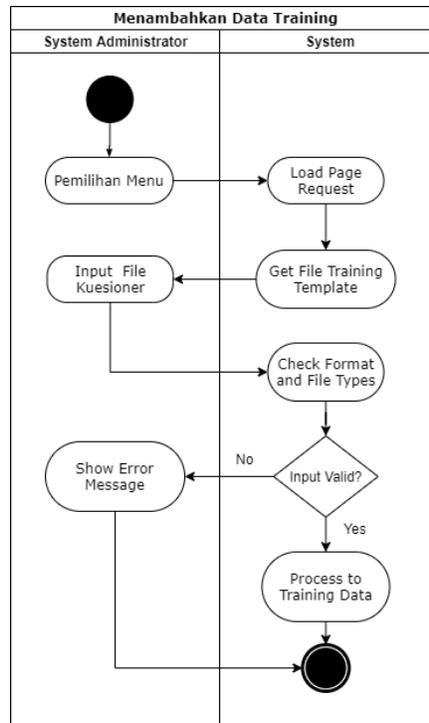
menambahkan file baru untuk diperiksa dan ditambahkan pada data kata normalisasi. File tersebut akan diperiksa dan disimpan jika sesuai dengan format yang telah ditentukan. Aktivitas ini digambarkan pada sebuah *activity diagram* pada Gambar 3.7.



Gambar 3.7 *Activity Diagram* Menambahkan Data Kata Normalisasi

g. Menambahkan *Data Training*

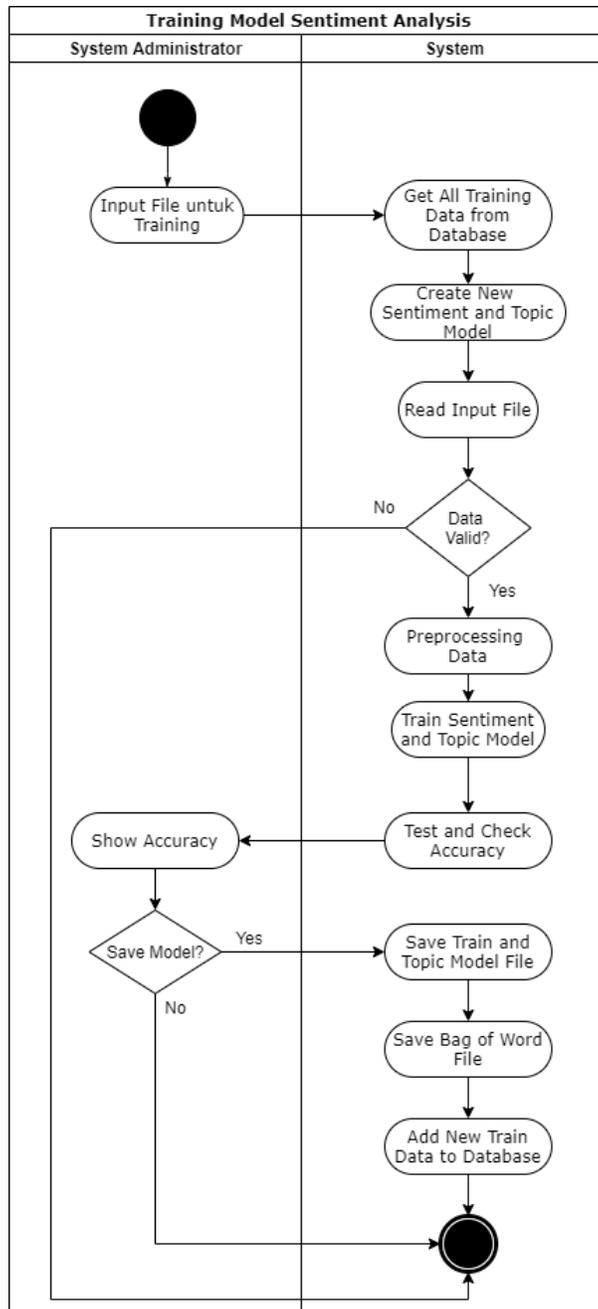
Aktivitas menambahkan data training dilakukan oleh *system administrator*. Tujuan dari aktivitas ini adalah menambahkan data *training* dan memperbaharui model. Awal dari aktivitas ini adalah ketika aktor memilih menu menambahkan data *training*. Aktor diharuskan menambahkan *file excel* yang berisi data training baru sesuai dengan contoh file data training yang diberikan sistem. Jika format yang diberikan sesuai dan valid, maka proses training akan dijalankan. Aktivitas ini digambarkan pada sebuah *activity diagram* pada Gambar 3.8.



Gambar 3.8 *Activity Diagram* Perbandingan Dua Semester

h. *Training Data*

Aktivitas *training data* digambarkan pada sebuah *activity diagram* Gambar 3.9. Tujuan dari aktivitas ini adalah melakukan pelatihan pada model sentimen dan topik. Awal dari aktivitas ini adalah ketika aktor yaitu *system administrator* memilih untuk melakukan *training* dan memberikan input berupa *file* data training. Sistem akan mengambil data training yang telah tersimpan di database untuk dijadikan satu kumpulan data dengan data training yang baru. Kemudian sistem akan menyiapkan model baru untuk klasifikasi topik dan sentimen. Data training kemudian akan diperiksa satu persatu dan akan dipersiapkan untuk training jika datanya valid. Model baru akan melalui proses *training* untuk klasifikasi dan topik. Setelah proses training selesai, sistem akan menghitung akurasi dari model dan menampilkannya pada aktor. Jika akurasi yang diinginkan sesuai dan aktor ingin menyimpan model yang baru, maka sistem akan menyimpan file model dan *bag of word* yang baru.



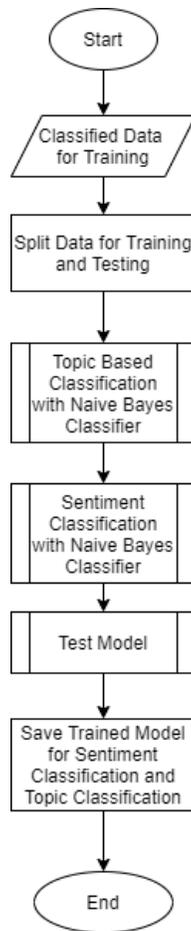
Gambar 3.9 Activity Diagram Training Data

3.4 Desain Flowchart Naive Bayes Classifier

Bagian ini menjelaskan mengenai alur kerja dari sistem dalam melakukan klasifikasi komentar menjadi sentimen dan topik yang sesuai. Alur kerja dari sistem akan digambarkan dalam bentuk *flowchart* yang membantu dalam keterangan *input*, *output*, dan proses yang terjadi. Secara keseluruhan model *Naive Bayes* yang digunakan pada aplikasi sentimen analisis ini terbagi menjadi dua bagian utama yaitu saat model *training* dan klasifikasi dengan model yang telah tersimpan.

3.4.1 *Flowchart Model Training*

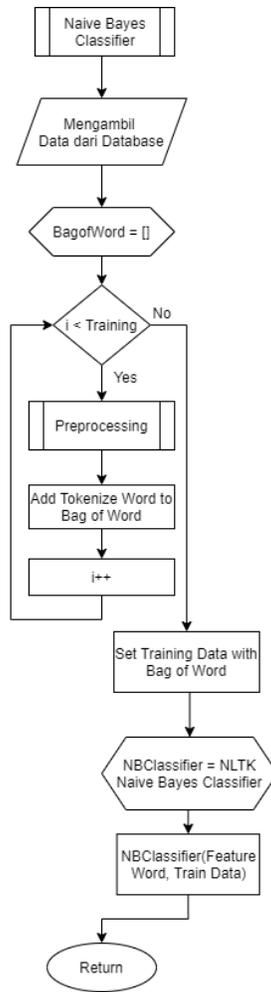
Flowchart untuk model *training* digambarkan pada Gambar 3.10. Alur kerja dimulai dengan adanya *input* berupa data yang akan diproses dalam pembuatan model. Data yang diterima akan dibagi menjadi data *training* dan data *testing*. Kemudian data yang digunakan sebagai *training* akan diproses untuk permodelan topik kemudian sentimennya dengan *Naive Bayes Classifier*, seperti yang ditunjukkan pada sub bab 3.4.2. Setelah model dibuat, maka akan dilakukan pengujian model dengan data *testing* seperti yang ditunjukkan pada sub bab 3.4.4. Jika model telah sesuai dan akurasiya baik, maka model klasifikasi topik dan sentimen akan disimpan.



Gambar 3.10 *Flowchart model training*

3.4.2 *Flowchart Model Naive Bayes Classifier*

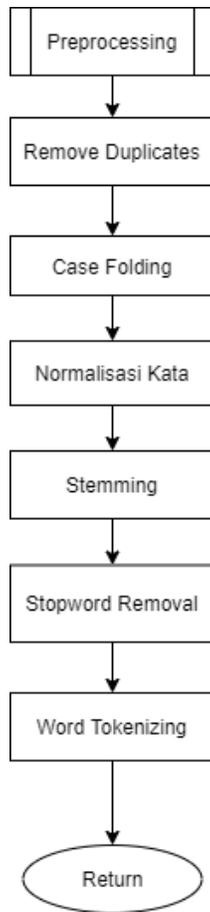
Pada Gambar 3.11. akan digambarkan alur kerja dari pembuatan model *Naive Bayes Classifier*. Tujuan dari alur kerja ini digunakan untuk membentuk model klasifikasi sentimen dan topik. Pembuatan model untuk klasifikasi dilakukan oleh *library Natural Language Toolkit (NLTK)*. Sistem akan mengambil data *training*, kemudian akan satu persatu dibersihkan melalui proses *preprocessing*. Komentar yang telah dibersihkan maka setiap katanya akan dimasukkan ke dalam variabel *Bag of Word* yang menampung setiap kata yang ada. Setelah semua data telah melalui proses penambahan kata dan *preprocessing* maka sistem akan membuat sebuah variabel bagi model. Variabel model akan melalui proses *training* dengan bantuan variabel *Bag of Word*. Setelah proses *training*, maka model dapat digunakan untuk klasifikasi baik sentimen maupun topik.



Gambar 3.11 *Flowchart Model Naive Bayes Classifier*

3.4.3 *Flowchart Preprocessing*

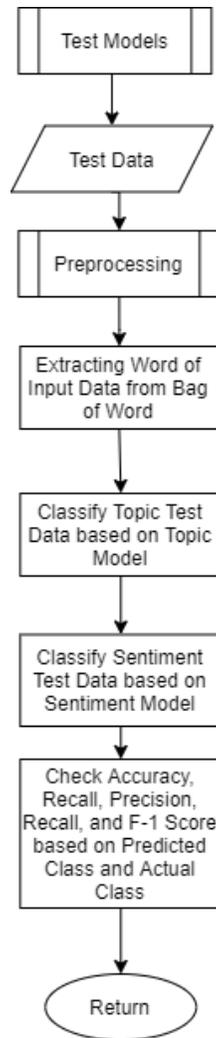
Preprocessing merupakan hal yang penting dilakukan untuk mempersiapkan data sebelum diproses dalam pembentukan model maupun klasifikasi. Gambar *flowchart* pada *preprocessing* dapat dilihat pada Gambar 3.12. Dengan adanya proses *preprocessing*, data diharapkan lebih seragam dan program dapat memiliki tingkat akurasi yang lebih tinggi. Proses *preprocessing* memiliki beberapa bagian yaitu melakukan penghapusan data yang memiliki komentar sama, *case folding*, normalisasi kata, *stopword removal*, *stemming*, dan *word tokenizing*. Berikut penjelasan terkait proses-proses yang dilakukan pada saat *preprocessing*.



Gambar 3.12 *Flowchart Preprocessing*

3.4.4 *Flowchart Model Testing*

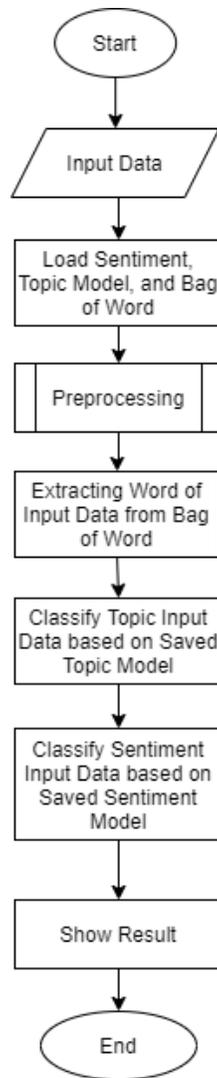
Testing merupakan proses pengujian sebuah model untuk menilai kinerja model dan akurasi. Data yang digunakan untuk model *testing* adalah data yang telah dibagi sebelum model *training*. Data akan melalui proses *preprocessing* dan kemudian diklasifikasikan berdasarkan topik dan sentimen menggunakan model yang sebelumnya telah dibuat. Hasil klasifikasi kemudian akan dicek akurasi dengan label yang ada di data *testing*. Pengecekan akurasi dilakukan dengan perhitungan akurasi, presisi, *recall*, dan *f-1 score*. Alur kerja model *testing* digambarkan dengan *flowchart* pada Gambar 3.13.



Gambar 3.13 *Flowchart Model Testing*

3.4.5 *Flowchart Klasifikasi Model yang Telah Disimpan*

Klasifikasi menggunakan model yang telah disimpan merupakan aktivitas yang dilakukan *main program* untuk melakukan klasifikasi bagi data baru dengan model yang sebelumnya telah melalui proses *training* dan modelnya disimpan dalam *file*. Alur kerja dimulai dengan data yang dimasukkan, kemudian sistem akan mengambil model sentimen dan topik, juga daftar kata *Bag of Word* dari file. Kemudian data akan melalui proses *preprocessing* dan ekstraksi kata sebelum dilakukan klasifikasi model. *Output* dari alur kerja ini adalah hasil klasifikasi sentimen dan topik dari kata yang dimasukkan. *Flowchart* klasifikasi model digambarkan pada Gambar 3.14.



Gambar 3.14 *Flowchart* Klasifikasi Model yang Telah Disimpan

3.5 Perhitungan Klasifikasi *Naive Bayes*

Dalam menentukan kelas sebuah kalimat, sistem menggunakan rumus dari *Teorema Bayes* seperti pada Persamaan 2.1. Perhitungan ini dipilih karena memiliki rumus yang sederhana dan hasil yang baik, meskipun mengasumsikan atribut sebagai atribut yang independen tanpa keterkaitan satu sama lain.

Perhitungan yang akan dilakukan untuk menentukan sentimen akan disederhanakan menjadi perhitungan probabilitas berdasarkan kata yang ada dalam kalimat dengan Persamaan di 2.2.

Berikut merupakan contoh perhitungan dari *Naive Bayes Classifier* untuk klasifikasi sentimen. Kalimat yang merupakan data awal atau data *training* ditampilkan pada tabel 3.1. Kemudian terdapat kalimat “Saya senang jika melihat anjing kecil” yang akan ditentukan kelas sentimennya.

Tabel 3.1 Data Awal

Kalimat	Sentimen
Saya senang sekali	Positif
Saya sedang sedih	Positif
Suasana hati ibu sedang senang	Negatif
Muka anjing itu terlihat sedih	Negatif

Jika dimasukkan sesuai dengan Persamaan 2.1. di atas, maka hasil dari contoh berikut dapat dilihat di Persamaan 3.1.. Kemudian sistem akan mencari probabilitas dari kata-kata masing-masing kelas dan kalimat seperti pada Persamaan 3.2.. Pada Persamaan 3.3. setiap kata yang ada pada kalimat yang akan diklasifikasi dihitung probabilitas munculnya pada kedua kelas. Daftar kata dan jumlah kata yang muncul pada kelas ditunjukkan pada Tabel 3.2. Perhitungan probabilitas kata untuk tiap kelas ditunjukkan pada Persamaan 3.3 dan 3.4.

$$P(\text{Positif} | \text{Saya senang jika melihat anjing kecil}) = \frac{P(\text{Saya senang jika melihat anjing kecil} | \text{Positif}) \times P(\text{Positif})}{P(\text{Saya senang jika melihat anjing kecil})} \quad (3.1)$$

$$P(\text{Positif}) = \frac{\text{Jumlah Kalimat Kelas Positif}}{\text{Jumlah Kalimat Keseluruhan Kelas}} = \frac{2}{4} = 0.5 \quad (3.2)$$

Tabel 3.2 Daftar Kata dan Frekuensi Kelas

Kata	Positif	Negatif
Saya	1	1
senang	2	
sekali	1	
sedang	1	1
sedih		2
Suasana	1	
hati	1	
ibu	1	
Muka		1
anjing		1
itu		1
terlihat		1
Total Jumlah Kata	8	8

$$\begin{aligned}
 P(\text{Saya senang jika melihat anjing kecil} \mid \text{Positif}) &= P(\text{Saya} \mid \text{Positif}) \times P(\text{senang} \mid \text{Positif}) \times P(\text{jika} \mid \text{Positif}) \times \\
 &P(\text{melihat} \mid \text{Positif}) \times P(\text{anjing} \mid \text{Positif}) \times P(\text{kecil} \mid \text{Positif}) \quad (3.3) \\
 &= (1/8) \times (2/8) \times ((0 \times 8) + 1) \\
 &= 0.03125
 \end{aligned}$$

$$\begin{aligned}
 P(\text{Saya senang jika melihat anjing kecil} \mid \text{Negatif}) &= P(\text{Saya} \mid \text{Negatif}) \times P(\text{senang} \mid \text{Negatif}) \times P(\text{jika} \mid \text{Negatif}) \times \\
 &P(\text{melihat} \mid \text{Negatif}) \times P(\text{anjing} \mid \text{Negatif}) \times P(\text{kecil} \mid \text{Negatif}) \quad (3.4) \\
 &= (1/8) \times ((0/8) + 1) \times ((0 \times 8) + 1) \times ((0 \times 8) + 1) \times ((0 \times 8) + 1) \times (1/8) \\
 &= 0.01525
 \end{aligned}$$

Berdasarkan perhitungan probabilitas kata, *class* Positif memiliki probabilitas yang lebih tinggi dibandingkan dengan *class* Negatif. Kemudian hasil dari perhitungan di Persamaan 3.3 dan

3.4. akan digunakan untuk memperlengkapi Persamaan 3.1, sehingga ditemukan bahwa probabilitas kata “Saya senang jika melihat anjing kecil” pada kelas Positif adalah 0.015625 seperti yang ditunjukkan pada Persamaan 3.5. Angka ini lebih tinggi dibandingkan dengan probabilitas dari kelas Negatif yang nilainya adalah 0.007813. Oleh karena itu, kalimat “Saya senang jika melihat anjing kecil” diklasifikasikan sebagai kelas “Positif”.

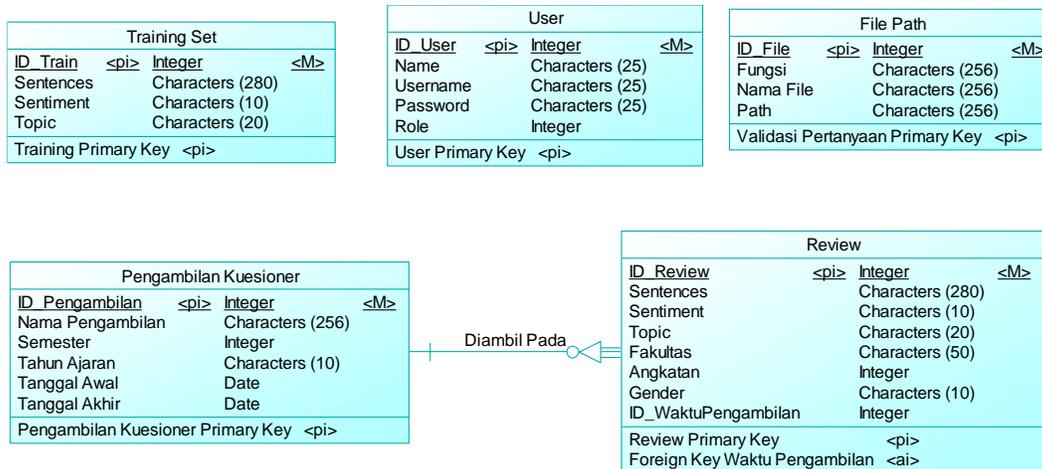
$$P(\text{Positif} \mid \text{Saya senang jika melihat anjing kecil}) = \frac{0.03125 \times 0.5}{1} = 0.015625 \quad (3.5)$$

3.6 Analisis Data

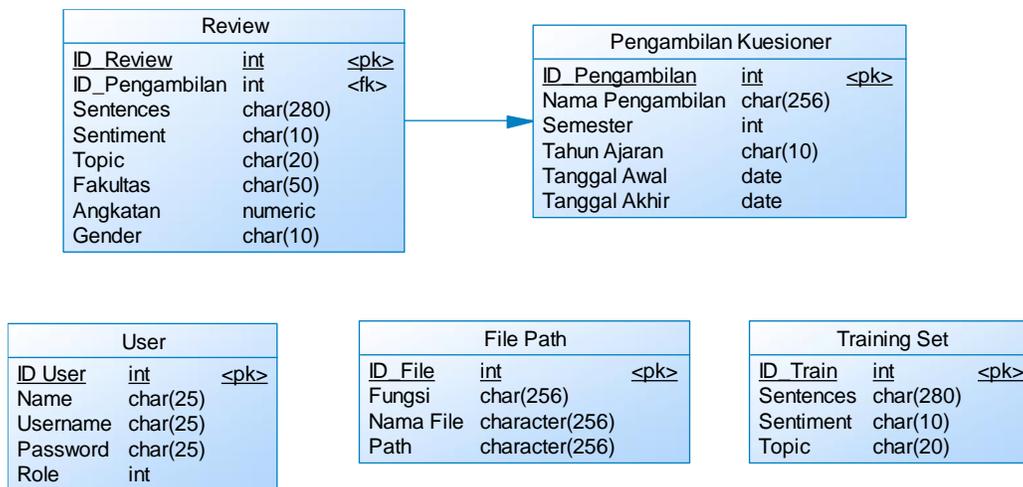
Pada penelitian ini data yang akan digunakan didapatkan dari kuesioner yang dibagikan kepada mahasiswa terkait pelaksanaan Pembelajaran Jarak Jauh (PJJ) di Universitas Kristen Petra pada semester genap 2019/2020 dan semester gasal 2020/2021. Kuesioner yang dibagikan pada semester genap 2019 berisi evaluasi mahasiswa terkait pelaksanaan pembelajaran jarak jauh pada paruh kedua semester tersebut (awal pandemi) tanpa label dan topik khusus. Sedangkan kuesioner yang dibagikan pada semester gasal 2020/2021 berisi beberapa pertanyaan dengan label sentimen dan topik yang ditentukan oleh responden sendiri. Hasil kuesioner pada semester gasal 2020/2021 akan digunakan sebagai data *training* dan *testing*, sedangkan hasil kuesioner pada semester 2019/2020 akan digunakan sebagai pembanding pelaksanaan pembelajaran jarak jauh. Hasil kedua kuesioner tersebut kemudian akan dikonsultasikan dengan pakar untuk melakukan pengecekan penggunaan Bahasa Indonesia yang baik dan benar dan label untuk topik dan sentimen.

3.7 Entity Relationship Diagram (ERD)

Entity Relationship Diagram merupakan model yang menggambarkan struktur penyimpanan data di *database*. Terdapat dua bentuk model dari ERD yaitu *conceptual model* dan *physical model*. *Conceptual model* dari struktur penyimpanan data di *database* aplikasi ini dapat dilihat pada Gambar 3.15. Sedangkan *physical model* dari struktur penyimpanan data di *database* aplikasi ini dapat dilihat pada Gambar 3.16.



Gambar 3.15 *Conceptual Entity Relationship Diagram*



Gambar 3.16 *Physical Entity Relationship Diagram*

Berdasarkan *ERD* pada Gambar 3.15. dan Gambar 3.16., dibuatlah sebuah *database* yang berisi tabel training set, validasi form, pengambilan kuesioner, dan review. Desain dari masing-masing tabel akan dijelaskan sebagai berikut:

a. Tabel *Training Set*

Tabel *training set* berisi data yang dapat dijadikan data acuan untuk model klasifikasi. Isi dari tabel *training set* adalah id yang digunakan untuk mengidentifikasi baris, kalimat yang berisi komentar, klasifikasi sentimen bagi kalimat tersebut, dan klasifikasi topik. Tabel untuk tabel *training set* dapat dilihat pada Tabel 3.3.

Tabel 3.3 *Training Set*

Nama Kolom	Tipe Data	Keys	Deskripsi
<i>ID_train</i>	integer	<i>Primary Key</i>	Identifier data <i>training</i> pada database
<i>sentences</i>	character(280)	-	Kalimat berisi komentar dari pengguna yang maksimal memiliki 280 karakter.
<i>sentiment</i>	character(10)	-	Label klasifikasi sentimen dari kalimat yang bernilai “Positif” atau “Negatif”
<i>topic</i>	character(20)	-	Label hasil klasifikasi topik dari kalimat

b. Tabel *File Path*

Tabel *File Path* akan mencatat lokasi dan nama file yang ada dibutuhkan sistem. Tabel ini membantu untuk melakukan validasi, klasifikasi, dan *training model* pada sistem. Desain dari tabel validasi form diberikan pada Tabel 3.4.

Tabel 3.4 *File Path*

Nama Kolom	Tipe Data	Keys	Deskripsi
ID_File	integer	<i>Primary Key</i>	Identifier data file pada database
Fungsi	character(256)	-	Mencatat fungsi dari file
Nama	character(256)	-	Mencatat nama <i>file</i>
Path	character(256)	-	Mencatat lokasi penyimpanan <i>file</i>

c. Tabel Pengambilan Kuesioner

Tabel Pengambilan Kuesioner mencatat tanggal kuesioner dibagikan dan dimasukan pada sistem. Tabel ini membantu untuk melakukan pendataan dari *review* mahasiswa berdasarkan nama atau judul pengambilan, tanggal awal pengambilan, tanggal akhir pengambilan, semester dan tahun ajarannya. Desain dari tabel pengambilan kuesioner diberikan pada Tabel 3.5.

Tabel 3.5 Pengambilan Kuesioner

Nama Kolom	Tipe Data	Keys	Deskripsi
ID_Pengambilan	integer	<i>Primary Key</i>	Identifier data pengambilan kuesioner pada database
Nama	character(256)	-	Nama atau Judul dari pengambilan kuesioner
Tanggal Awal	date	-	Mencatat tanggal awal pengambilan kuesioner
Tanggal Akhir	date	-	Mencatat tanggal akhir pengambilan kuesioner

Nama Kolom	Tipe Data	Keys	Deskripsi
Semester	integer	-	Mencatat semester saat pengambilan kuesioner, seperti contoh 1 untuk Semester Gasal, dan 2 untuk Semester Genap.
Tahun Ajaran	character(10)	-	Mencatat tahun ajaran saat pengambilan kuesioner dilaksanakan

d. Tabel *Review*

Tabel *review* berisi data dari angket mahasiswa yang telah disimpan pada *database*. Tabel ini berisi data mahasiswa seperti komentar mahasiswa, asal fakultas mahasiswa, tahun masuk mahasiswa, klasifikasi sentimen, dan klasifikasi topik dari komentar mahasiswa. Tabel 3.6. menunjukkan desain dari tabel *review*

Tabel 3.6 *Review*

Nama Kolom	Tipe Data	Keys	Deskripsi
<i>ID_review</i>	integer	<i>Primary Key</i>	Identifier data <i>review</i> pada database
<i>sentences</i>	character(280)	-	Kalimat berisi komentar dari pengguna yang maksimal memiliki 280 karakter.
<i>sentiment</i>	character(10)	-	Label klasifikasi sentimen dari kalimat yang bernilai “Positif” atau “Negatif”

Nama Kolom	Tipe Data	Keys	Deskripsi
<i>Topic</i>	character(20)	-	Label hasil klasifikasi topik dari kalimat
fakultas	character(50)	-	Fakultas asal mahasiswa
angkatan	numeric	-	Tahun masuk universitas mahasiswa
ID_pengambilan	character(10)	<i>Foreign Key</i>	<i>Identifier</i> untuk data semester

e. Tabel *User*

Tabel *user* berisi data dari pengguna sistem yang telah disimpan pada *database*. Kegunaan tabel ini adalah untuk melakukan pengecekan sebelum pengguna menggunakan sistem. Tabel ini berisi data pengguna seperti username, password, dan role user. Tabel 3.7. menunjukkan desain dari tabel *user*.

Tabel 3.7 *User*

Nama Kolom	Tipe Data	Keys	Deskripsi
<i>ID_User</i>	integer	<i>Primary Key</i>	Identifier data <i>user</i> pada database
<i>Name</i>	character(25)		Data nama pengguna
<i>username</i>	character(25)	-	Nama yang akan digunakan pengguna untuk <i>login</i>
<i>Password</i>	character(25)	-	Kata sandi yang akan digunakan pengguna untuk <i>login</i>

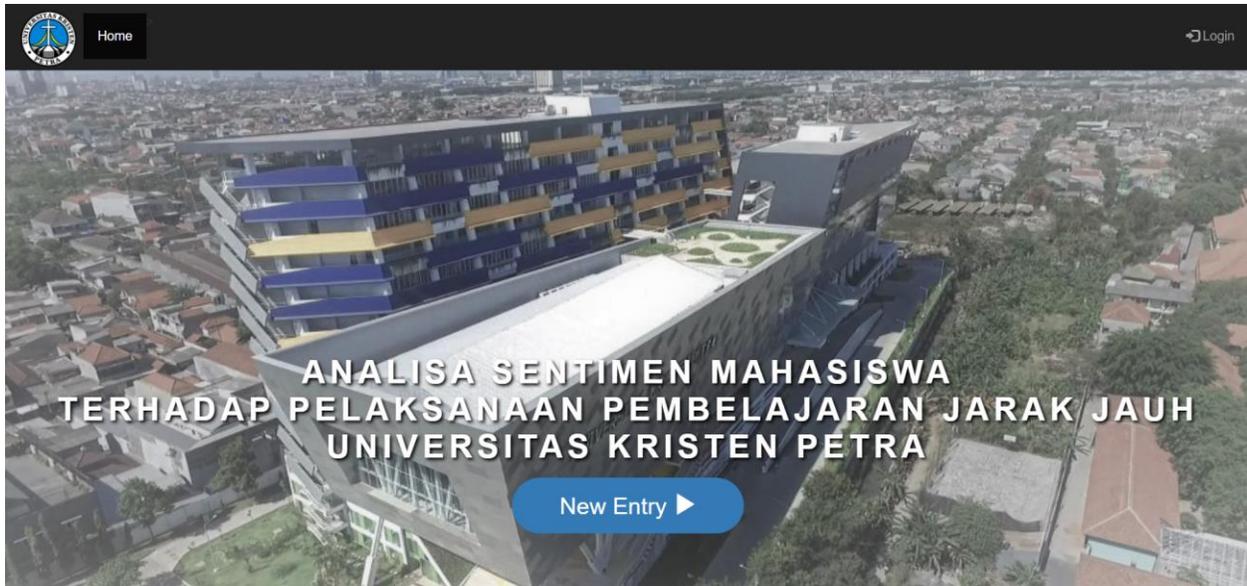
Nama Kolom	Tipe Data	Keys	Deskripsi
<i>Role</i>	integer	-	Hak akses yang diberikan kepada pengguna dalam bentuk angka, dengan 0 berarti “Pimpinan Organisasi”, 1 berarti “ <i>User Administrator</i> ”, dan 2 berarti “ <i>System Administrator</i> ”.

3.8 Desain User Interface

Pada bagian ini akan ditampilkan rancangan tampilan dari aplikasi yang akan dibuat. Tampilan dari aplikasi terdiri dari beberapa halaman utama seperti halaman utama, halaman laporan, halaman *input*, dan halaman admin.

3.8.1 Desain Halaman Utama

Desain dari halaman utama terdiri dari judul aplikasi dan menu untuk melakukan *login* sebelum pengguna dapat melakukan aktivitas sesuai dengan *role* masing-masing. Menu *login* ditampilkan di halaman utama yang jika dipilih akan menampilkan sub-halaman untuk mengisi *username* dan *password*. Tampilan untuk desain halaman utama dan menu login dapat dilihat pada Gambar 3.17. dan Gambar 3.18.



Gambar 3.17 Desain Tampilan Halaman Utama

Login	
Username	<input type="text"/>
Password	<input type="password"/>
<input type="button" value="Login"/>	

Gambar 3.18 . Desain Tampilan Menu *Login*

3.8.2 *Desain Halaman Entry*

Desain halaman *entry* berisi kolom pengisian data-data dari kuesioner dan sebuah tombol yang akan melakukan proses data jika ditekan. Data-data diisi pada *field* yang telah disediakan dan *file* yang akan diproses. Tampilan dari halaman *entry* dapat dilihat pada Gambar 3.19.

Gambar 3.19 Desain Tampilan Halaman *Entry*

3.8.3 *Desain Halaman Report*

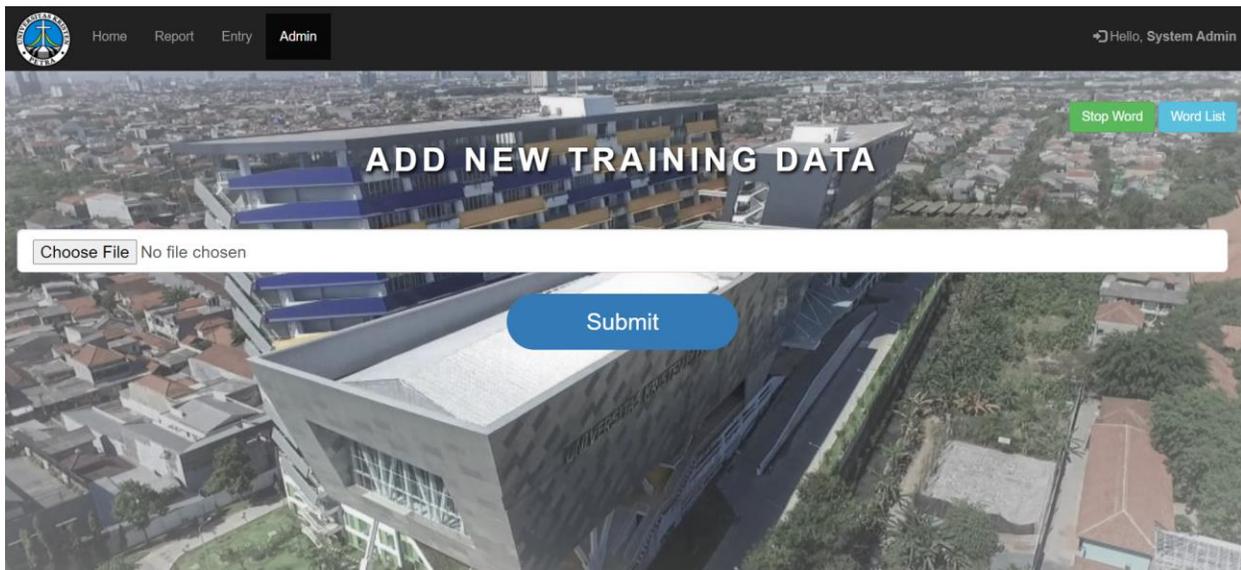
Halaman *Report* merupakan halaman yang menunjukkan hasil dari kuesioner yang telah diproses dan kuesioner yang sebelumnya. Pada halaman ini akan ditampilkan grafik untuk masing-masing topik sesuai dengan kategori pembandingan yang dipilih pada *field* yang tersedia. Tampilan dari halaman ini dapat dilihat pada Gambar 3.20.



Gambar 3.20 Desain Tampilan Halaman *Report*

3.8.4 *Desain Halaman System Administrator*

Halaman sistem administrator digunakan untuk melakukan pengelolaan model dan penambahan data-data yang berkaitan dengan proses klasifikasi. Pada halaman ini disediakan *field* untuk menambahkan data *training*, daftar kata *stopword*, dan kata normalisasi. Tampilan dari desain halaman *system administrator* dapat dilihat pada Gambar 3.21.



Gambar 3.21 Desain Tampilan Halaman *System Administrator*