BAB III

METODOLOGI PENELITIAN DAN PERANCANGAN SISTEM

3.1 Metodologi Penelitian

Penelitian "Implementasi Extreme Gradient Boosting pada *Sentiment Analysis* dalam *Social Media* Facebook" menggunakan beberapa tahap agar metodologi dan perancangan sistem dapat terpenuhi. Tahap-tahap yang dilaksanakan antara lain adalah sebagai berikut.

1. Studi Literatur

   Melakukan studi literatur dengan cara mencari, mempelajari dan membaca penelitian-penelitian yang bersumber dari jurnal ilmiah dan karya tulis ilmiah terkait dengan *sentiment analysis* dan algoritma XGBoost.

2. Pengumpulan Data

   Dataset didapatkan dari penelitian Rachmat dan Lukito (2016) yang berjudul *Dataset* Sentimen Komentar Pada Kampanye Pemilu Presiden 2014 dari Facebook Page. *Dataset* ini akan dinamakan Pemilu2014, *dataset* Pemilu2014 merupakan gabungan komentar dari 68 status berbeda dengan *dataset* sejumlah 3300 komentar. *Dataset* ini memiliki tingkat konfiden sebesar 95% dan validitas sebesar 95.3% (Rachmat dan Lukito, 2016). *Dataset* tambahan yang digunakan untuk menyeimbangkan label negatif dan netral didapatkan dari penelitian Saputri dan Mahendra (2018). *Dataset* tamabahan ini akan dinamakan SentimentTwitter, *dataset* SentimentTwitter merupakan gabungan *random* tweet dari media sosial

Twitter dengan *dataset* sejumlah 3115 tweet. *Dataset* yang digunakan dalam penelitian ini hanyalah data dengan label negatif berjumlah 1101 dan netral berjumlah 997.

3. Perancangan Aplikasi

Proses perancangan aplikasi dilakukan dengan merancang alur kerja dari implementasi XGBoost beserta *word embedding* FastText, struktur tabel *database* yang digunakan, dan alur antarmuka berdasarkan kebutuhan penelitian.

4. Implementasi

Proses implementasi dilakukan menggunakan bahasa pemrograman Python dengan Jupyter Notebook dan Google Collaboration dalam mengimplementasikan XGBoost untuk klasifikasi sentimen pada sebuah kalimat dengan menggunakan FastText *pre-trained model* sebagai *feature extraction*, dan membuat aplikasi *web* dari model yang telah dibuat menggunakan XGBoost sebagai visualisasi proses klasifikasi menggunakan *flask*.

5. Pengujian dan Evaluasi

Pengujian bertujuan untuk menguji apakah algoritma XGBoost berhasil diimplementasikan dan dapat berjalan dengan baik. Evaluasi performa dilakukan dengan menghitung *accuracy, precision*, *recall*, dan F1 *score* dari hasil klasifikasi teks yang dilakukan.

6. Penulisan laporan

Laporan dibuat sebagai dokumentasi dari penelitian dan pembuatan aplikasi. Laporan dilakukan secara bertahap. Pengerjaan

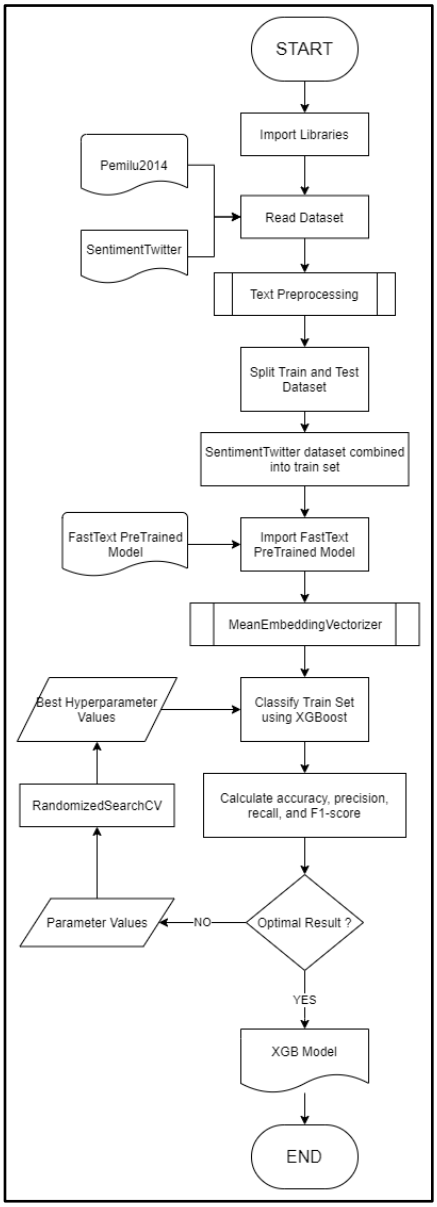
laporan bagian implementasi dikerjakan setelah program selesai dilakukan.

3.2 Perancangan Aplikasi

Sistem yang dibuat dapat dijelaskan melalui *flowchart* proses klasifikasi, *flowchart* aplikasi *web,* rancangan struktur tabel, dan rancangan tampilan antarmuka.

3.2.1 Flowchart Proses Klasifikasi

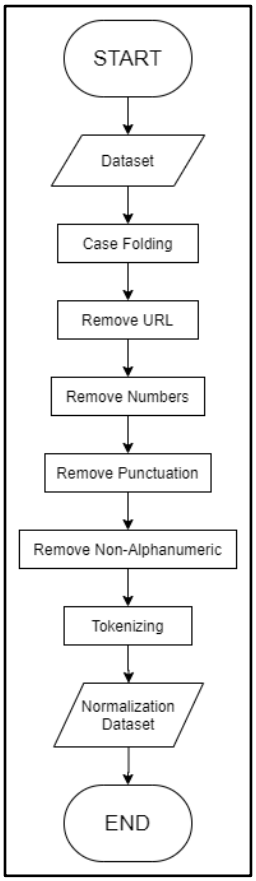
Gambar 3.1 adalah *flowchart* dari proses klasifikasi sentimen menggunakan XGBoost secara umum. Sebelum melakukan proses klasifikasi lebih lanjut, *library* yang dibutuhkan dalam penelitian di-*import* terlebih dahulu. Kemudian, *dataset* Pemilu2014 dimasukkan untuk dapat diolah. *Dataset* SentimentTwitter juga dimasukkan untuk dapat digabungkan setelah proses pemecahan data *train set* dan *test* set yang bertujuan mengatasi ketidakseimbangan *dataset* utama pada label negatif dan netral. *Train set* akan mengalami penambahan jumlah data pada label negatif dan netral.



Gambar 3.1 *Flowchart* analisa sentimen secara umum

Data yang diperoleh dari *dataset* Pemilu2014 dan SentimentTwitter akan dinormalisasikan dengan mengambil kolom ataupun *feature* yang dibutuhkan dalam proses pengolahan data. Selain itu, tahap *preprocessing* dilakukan agar penelitian menjadi lebih efektif. Tahapan *preprocessing* yang dilakukan antara lain *Case Floding* (Mengubah setiap huruf besar pada kalimat menjadi huruf kecil), menghilangkan URL (https dan http), menghilangkan angka, menghilangkan tanda baca, menghilangkan *non-alphanumeric*, dan *tokenizing* (pemisahan kata dari

kalimat). Proses *Text Preprocessing* dapat dilihat pada Gambar 3.2. Tahapan ini dilakukan dengan bantuan *library regex* (*regular expression)* dan *string*. *Feature extraction* yang digunakan dalam penelitian ini adalah *word embedding* sehingga tidak perlu dilakukan lagi *stemming* dan *stopword removal.*



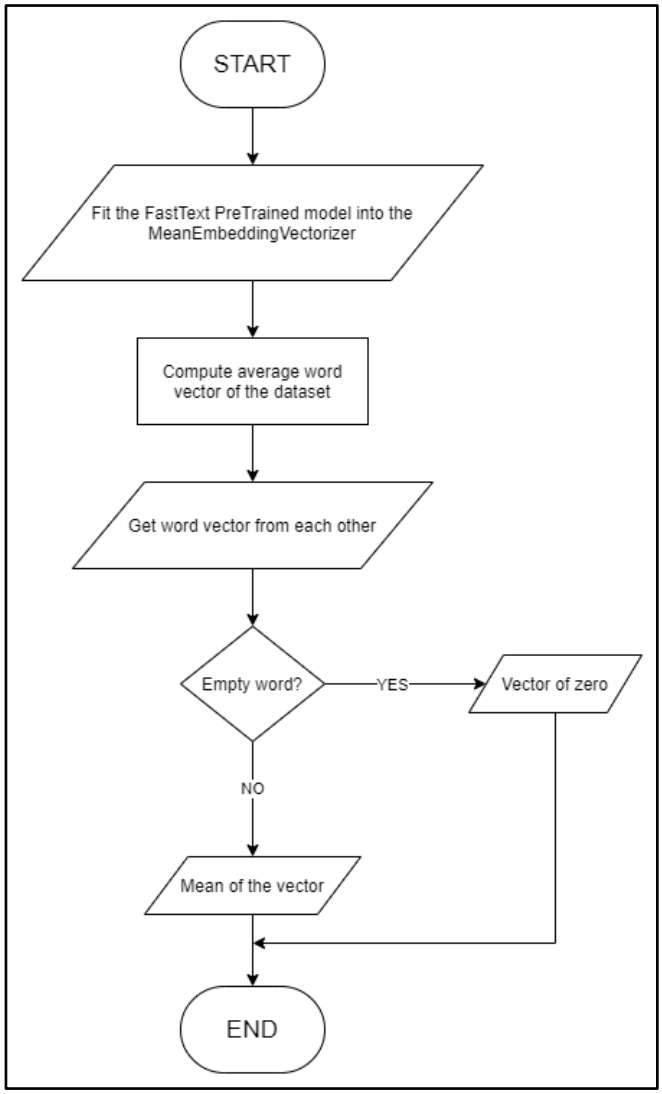
Gambar 3. 2 *Text Preprocessing*

Setelah tahap *preprocessing,* dilakukan pembagian data ke dalam *train set* dan *test set* sesuai dengan *scenario* yang dilakukan. Tahapan ini dilakukan dengan bantuan *library Sklearn Train Test Split*. *Train set* merupakan data yang akan digunakan untuk melatih *model machine learning*. *Test set* merupakan data yang digunakan untuk memberikan evaluasi model yang tidak bias setelah di-*train*. Kemudian, dilakukan penggabungan *dataset* SentimentTwitter ke dalam *train set.* Langkah perbaikan dilakukan untuk meng-*handle* ketidakseimbangan jumlah data

yang dimiliki kelas positif, negatif, dan netral dengan menggabungkan *dataset* SentimentTwitter untuk memberikan hasil klasifikasi lebih baik pada setiap label.

RandomizedSearchCV digunakan dalam *tuning hyperparameter* XGBoost dalam pembuatan model. Tahapan ini dilakukan dengan bantuan *library* RandomizedSearchCV dari Sklearn Model Selection. Setiap *hyperparameter* yang ingin digunakan diberikan angka. *Output* yang didapatkan merupakan hasil penggabungan antar *hyperparameter* yang memberikan hasil paling optimal.

Gambar 3.3 merupakan *flowchart* dari proses *vectorize dataset* menggunakan *MeanEmbeddingVectorizer*. Sebelum proses ini dilakukan, FastText *pre-trained model* yang telah diunduh di-*load* terlebih dahulu yang kemudian di-*fit* ke dalam *MeanEmbeddingVectorizer*. Kemudian, setiap *input* kata yang telah di *tokenize* akan dihitung dan diambil nilai vektor rata-ratanya dari model FastText *pre-trained*. Apabila *input* merupakan sebuah kata maka *output* akan berupa nilai angka rata-rata dari vektor setiap kata tersebut dan nilai vektor akan menjadi 0 (nol) jika tidak terdapat sebuah kata atau bukan merupakan sebuah karakter.



Gambar 3.3 *MeanEmbeddingVectorizer*

Sebelum memasuki tahap pembuatan model menggunakan XGBoost, data direpresentasikan menjadi sebuah *metrics* menggunakan DMatrix. DMatrix memiliki sebuah parameter yang sangat berguna untuk mengatasi permasalahan *dataset* yang tidak seimbang, yaitu parameter *weight.* Parameter *weight* akan memberikan bobot pada tiap label yang ada. Parameter *weight* merupakan salah satu cara untuk mengatasi ketidakseimbangan *dataset* selain cara *upsampling* dan *downsampling.*

Pada tahap XGBoost dilakukan klasifikasi dengan cara memberikan bobot untuk tiap fitur di setiap *tree* yang ditumbuhkan. Jumlah *tree* dalam model ditentukan oleh parameter n_estimator. Data latih dibagi menjadi sejumlah subset sesuai dengan n_estimator yang ditentukan. Setiap subset data menghasilkan satu *tree*, sehingga pada akhirnya *tree* yang terbentuk dalam model adalah sejumlah n_estimator.

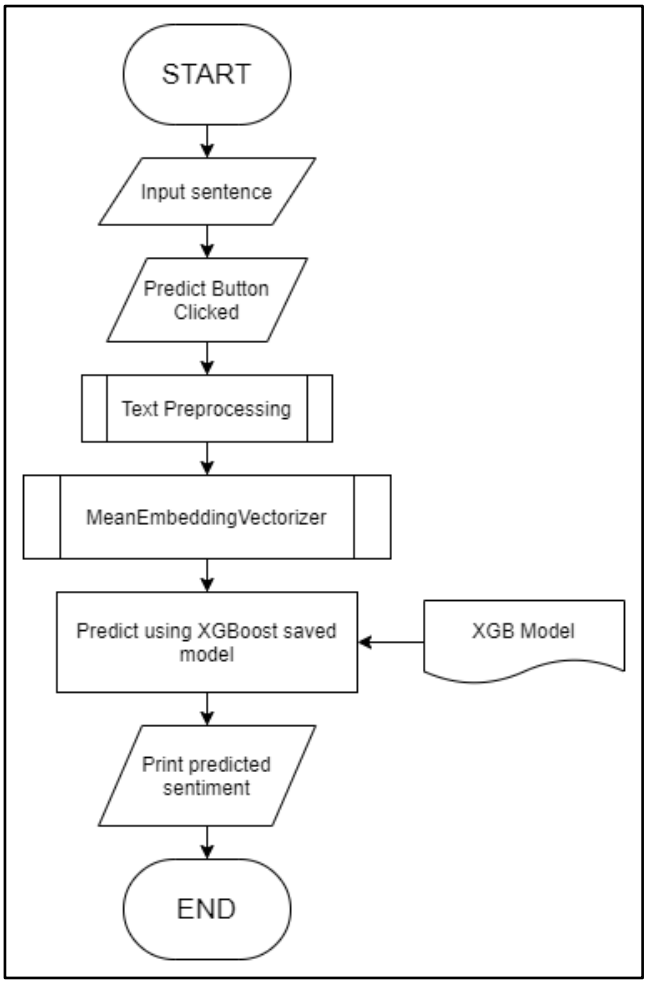
Tahap terakhir yaitu melakukan evaluasi performa model yang telah dibuat menggunakan *confusion matrix*. *Confusion matrix* mempresentasikan prediksi dan kondisi sebenarnya (*actual*) dari data yang dihasilkan oleh model. Evaluasi dengan *confusion matrix* menghasilkan nilai *accuracy, precision, recall.* Kemudian nilai ini akan digunakan untuk melakukan perhitungan nilai F1. Tahap ini dilakukan dengan bantuan *library Sklearn Metrics.*

3.2.2 Flowchart Aplikasi Web

Agar penelitian Analisis *sentiment* dengan XGBoost dapat dirasakan manfaatnya oleh masyarakat luas, peneliti lainnya, dan lembaga statistika lainnya, maka model klasifikasi hasil *training* yang telah dilakukan akan dievaluasi dan di-*deploy* ke dalam aplikasi *web*. Dalam aplikasi *web* yang dibangun akan ada 2 (dua) jenis *input* yang bisa digunakan untuk memvisualisasikan hasil prediksi dari model yang telah dibuat, yaitu *input* berupa kalimat dan *input* berupa *file* dengan *format* xlsx dengan diberikan nama kolom.

Gambar 3.4 Merupakan *flowchart* dari proses prediksi dari *input* kalimat melalui *text box* pada aplikasi *web. Input* berupa kalimat tersebut akan diproses setelah tombol Predict ditekan. Setelah itu, *web* akan menjalankan proses klasifikasi

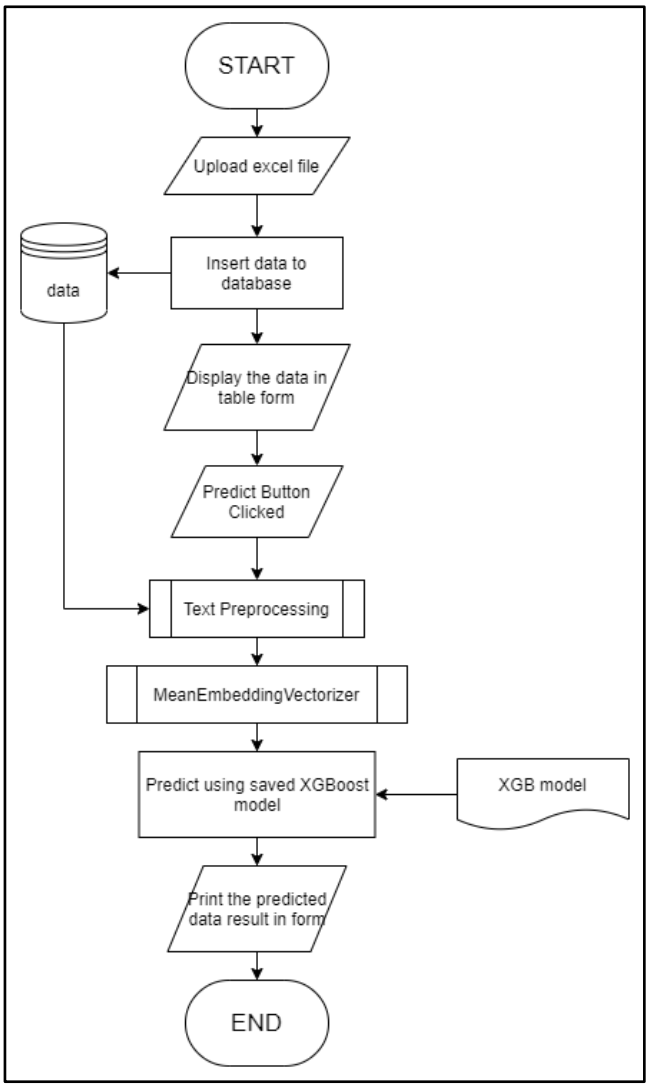
seperti yang telah dijelaskan sebelumnya, namun tanpa melalui proses *training* karena sudah menggunakan hasil model dari XGBoost yang telah dilakukan. Hasil yang didapat akan dikembalikan ke laman *web* berupa prediksi dari *input* kalimat yang diberikan.



Gambar 3.4 *Flowchart Sentence Prediction*

Gambar 3.5 merupakan *flowchart* dari proses prediksi dari *input* file excel yang diunggah ke dalam aplikasi *web*. Isi konten dari file yang diunggah akan ditampung sementara ke dalam *database* dan ditampilkan di halaman *web* dalam bentuk tabel. Apabila tombol Predict ditekan, maka proses klasifikasi yang telah dijelaskan sebelumnya akan dilakukan. Model XGBoost akan memprediksi setiap *row* data yang terdapat dalam *database*. Hasil prediksi akan ditampilkan pada

laman *web* dalam bentuk tabel dengan label setiap kalimat dan jumlah data dari setiap label dengan diikuti persentasenya.



Gambar 3.5 *Flowchart File Prediction*

3.2.3 Rancangan Struktur Tabel

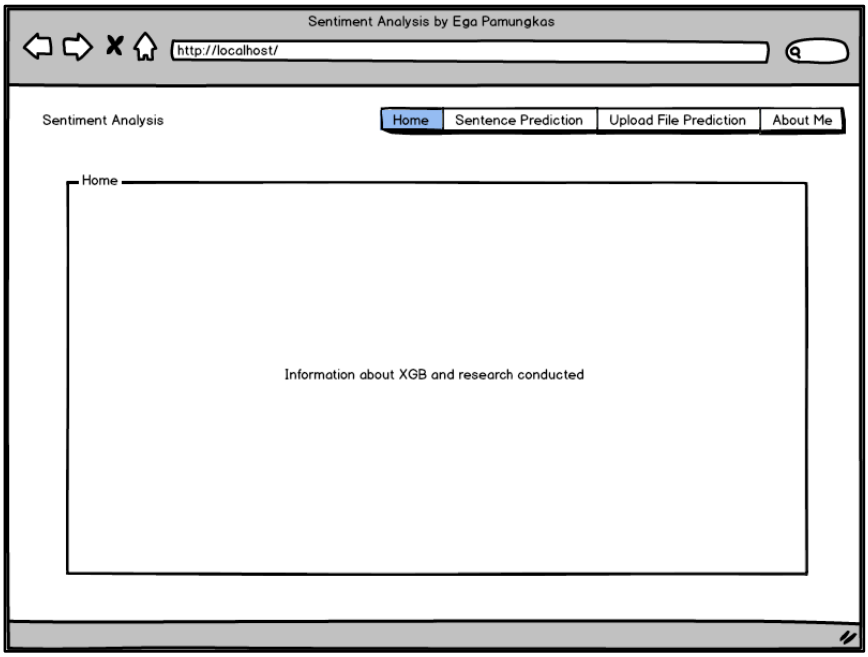
Tabel 3.1 merupakan struktur tabel pada *database* yang MySQL. Tabel ini hanya memiliki satu kolom dengan tipe data TEXT. Tabel ini berfungsi untuk menampung sementara data yang diunggah ke halaman *web* sebelum diprediksi.

Tabel 3.1 Struktur Tabel Data

Nama Kolom	Tipe Data	Atribut
data	TEXT	NOT NULL

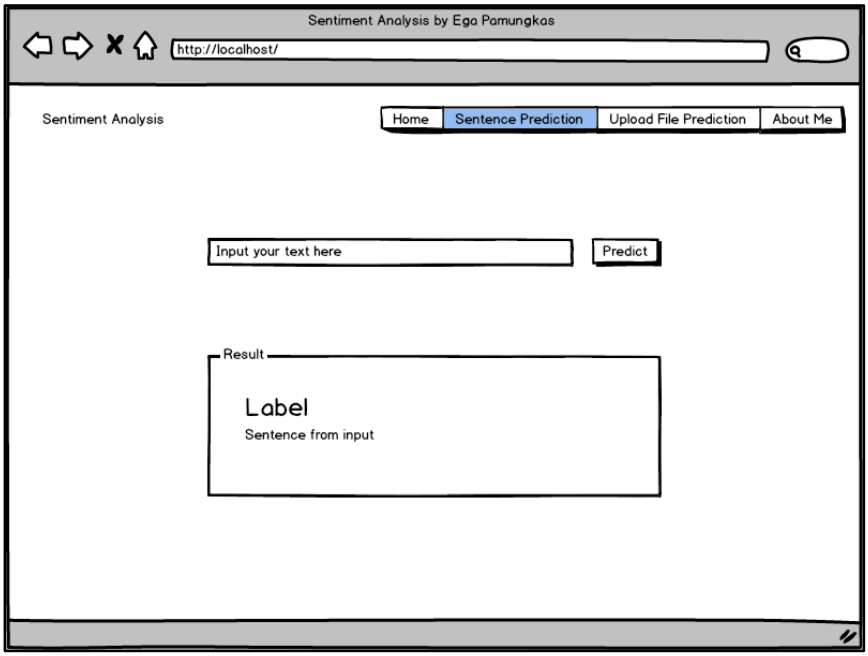
3.2.4 Rancangan Tampilan Antarmuka

Tampilan antarmuka aplikasi *web* terbagi menjadi empat halaman, yaitu halaman Home, Sentence Prediction, Upload File Prediction dan halaman About Me. Pada Gambar 3.6 terdapat rancangan antarmuka untuk halaman Home. Halaman ini memberikan informasi terkait dengan penelitian yang dilakukan.



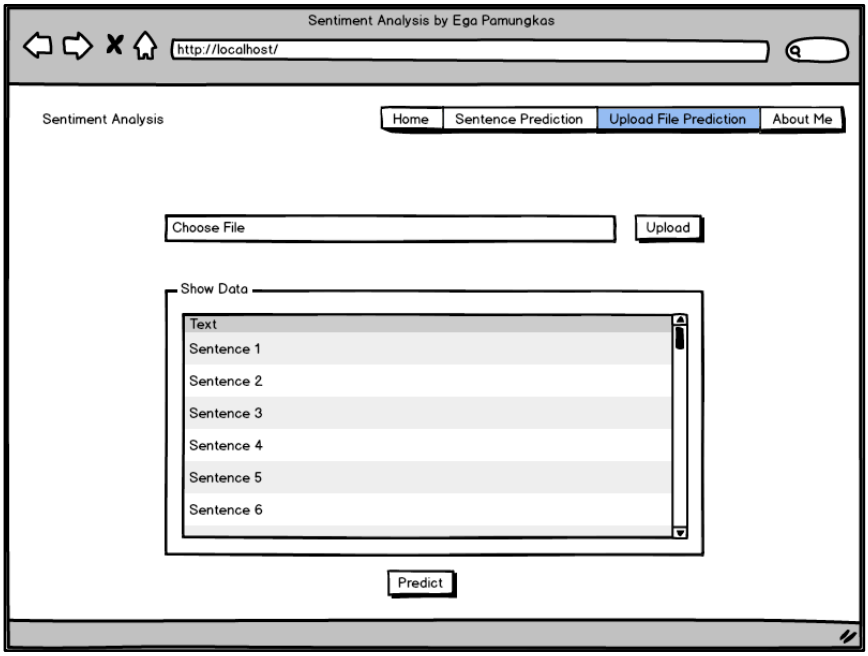
Gambar 3.6 Halaman Utama atau Home

Pada Gambar 3.7 terdapat rancangan antarmuka untuk halaman Sentence Prediction. Di halaman ini, *user* dapat memasukkan *input* berupa kalimat pada *text box* dan tombol Predict untuk dapat melihat hasil prediksi dari kalimat yang telah dimasukkan.



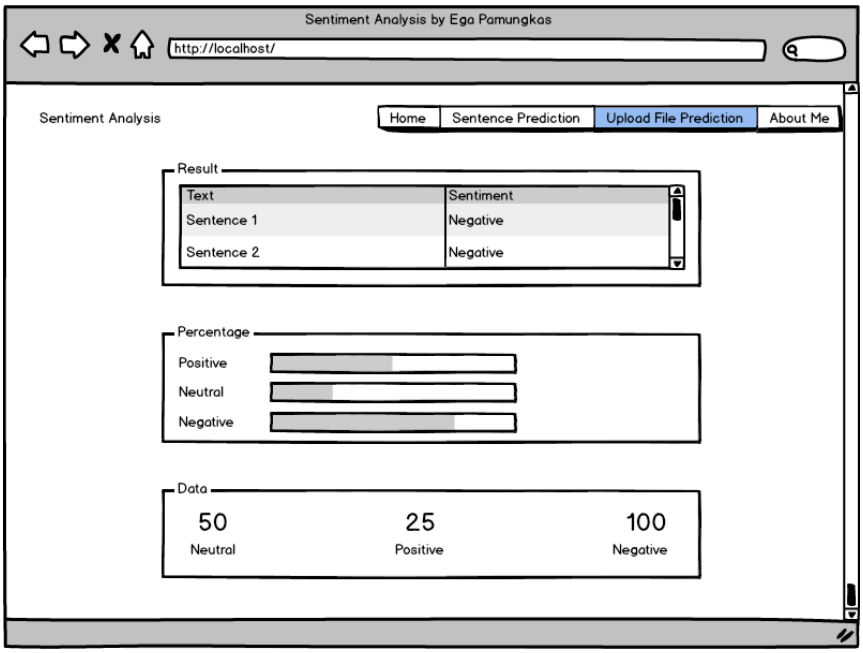
Gambar 3.7 Halaman Upload File

Pada Gambar 3.8 terdapat rancangan antarmuka untuk halaman Upload File Prediction. Di halaman ini, *user* dapat mengunggah *file* berekstensi .xlsx dan sebuah tombol *upload* yang berfungsi mengunggah *file* yang sudah dipilih.



Gambar 3.8 Halaman Upload File Sentence

Pada Gambar 3.9 merupakan lanjutan antarmuka untuk halaman Upload File Prediction. Di halaman ini menampilkan hasil prediksi dari *file* yang sudah diunggah. Diberikan persentase dan data dari setiap label dari hasil prediksi yang dilakukan.



Gambar 3.9 Halaman Upload File Prediction (Lanjutan)