

BAB 3

METODOLOGI DAN PERANCANGAN SISTEM

3.1 Metodologi Penelitian

Metode yang digunakan dalam tahap pengerjaan penelitian terdiri atas studi literatur, analisis kebutuhan, pengerjaan program, uji coba dan evaluasi, dan konsultasi dan penulisan.

1. Studi Literatur

Pada tahap telaah literatur dilakukan pembelajaran mengenai *text classification* dan *text processing*, *fastText*, *multilayer perceptron* dan evaluasi klasifikasi yang paling awal proses penelitian yang akan dilakukan.

2. Analisis Kebutuhan

Pada tahap ini, dilakukan analisis terhadap metode, data dan proses apa saja yang akan digunakan pada penelitian ini. Proses analisis dimulai dari teori yang akan digunakan ke dalam program yang akan dibuat. Pada tahap ini juga menganalisis bentuk keluaran apa saja yang akan dihasilkan oleh program.

3. Pengerjaan Program

Pada tahap ini dilakukan implementasi yang sudah dilakukan pada tahap-tahap sebelumnya. Pengerjaan program dilakukan dengan menggunakan bahasa pemrograman python dan HTML.

4. Uji Coba dan Evaluasi

Program yang sudah diimplementasikan, pada tahap ini diuji dan dievaluasi untuk memastikan semua yang ditentukan berjalan dengan baik.

5. Konsultasi dan Penulisan

Penulisan laporan dilakukan pada tahap ini dengan untuk mendokumentasikan segala bentuk proses penelitian dan menyimpulkan hasil akhir yang didapat dari penelitian ini.

3.2 Pengumpulan Data

Teknik pengumpulan data yang dilakukan pada penelitian ini menggunakan dataset berbahasa Indonesia yang telah dibuat oleh penelitian sebelumnya yang dilakukan oleh Ibrohim & Budi pada tahun 2018. Dalam dataset acuan, tersedia informasi multilabel untuk mengidentifikasi bahasa kasar dan *hatespeech* tanpa mengidentifikasi target, kategori, dan tingkat ujaran kebencian. Dalam penelitian yang akan dilakukan proses klasifikasi hanya akan dilakukan tanpa mengidentifikasi target, kategori, dan tingkat ujaran kebencian.

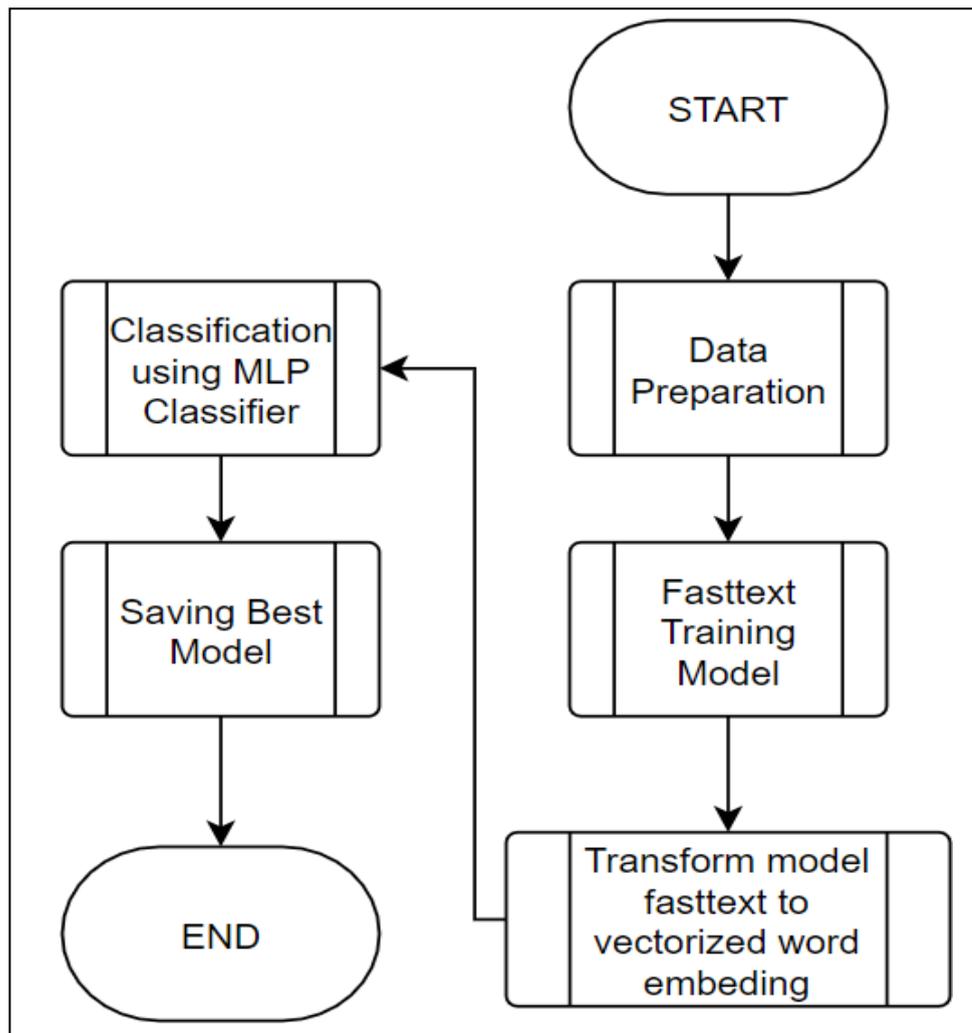
3.3 Perancangan Sistem

Perancangan sistem yang dilakukan terdiri dari dua aplikasi yang berbeda yaitu *Training and Evaluation* dan *Implementation Hatespeech Classification*. Pada *Training and Evaluation* dibuat menggunakan bahasa pemrograman *python 3* dengan bantuan aplikasi Jupiter Notebook yang menghasilkan FastText model dan *Multilayer Perceptron Classification Model* yang akan digunakan pada aplikasi

kedua yaitu *Hatespeech Classification*. Aplikasi *Hatespeech Classification* dibuat dengan menggunakan kerangka kerja *Flask* dan bahasa pemrograman *python* yang akan menampilkan HTML sebagai tampilan antarmuka.

3.3.1 Flowchart Umum Proses Training dan Evaluation

Flowchart umum proses *training* dan *evaluation* yang ditunjukkan oleh Gambar 3.1 merupakan *flowchart* dari proses klasifikasi menggunakan *Multilayer Perceptron Classifier* dengan *FastText Word Embedding*.



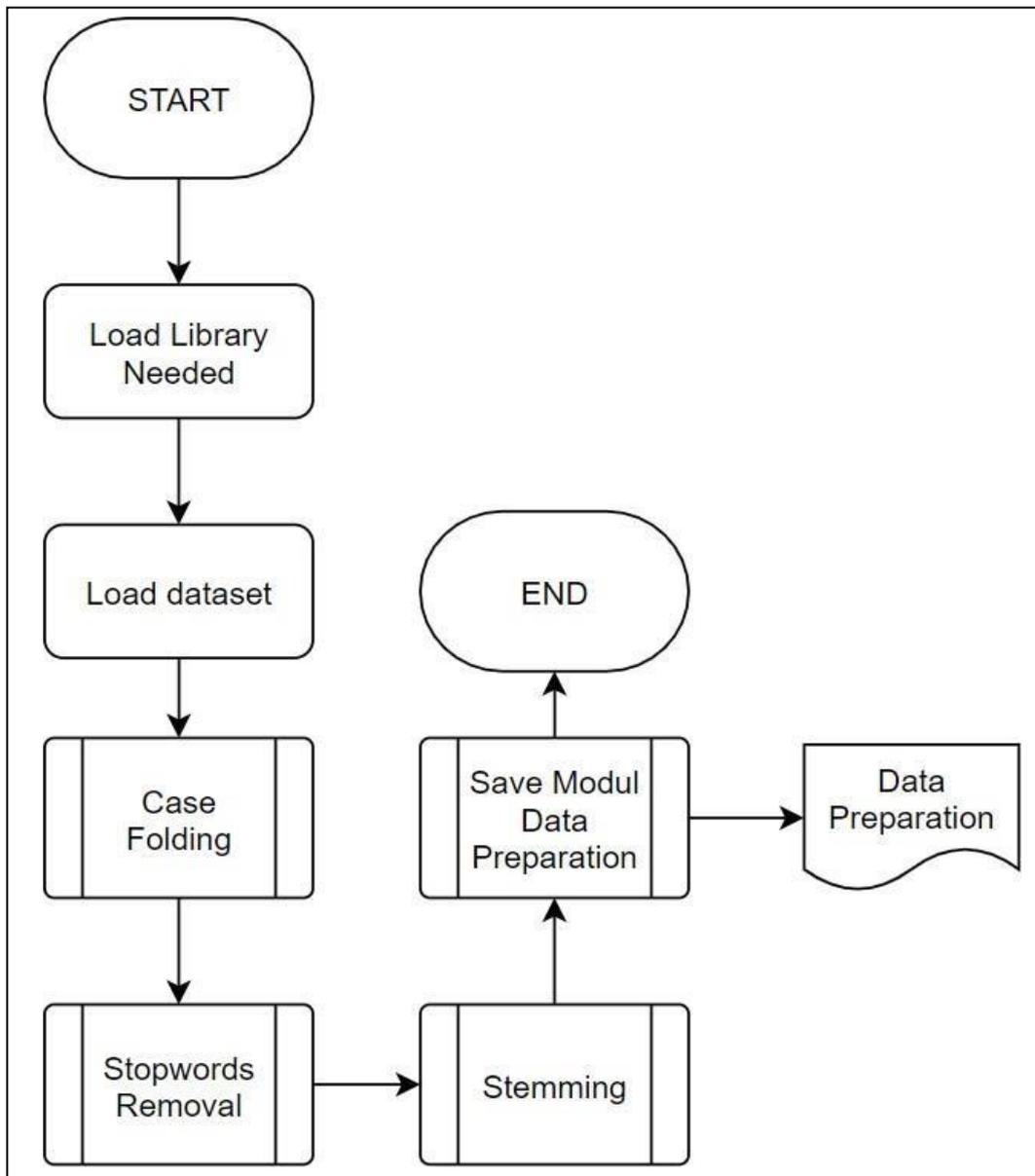
Gambar 3.1 Flowchart Umum Aplikasi Training dan Evaluation dengan Full Data Preparation

Alur proses dimulai dengan *data preparation*, dalam proses *preprocessing* pada modul *data preparation* dibagi menjadi dua skenario yang dimana dengan menggunakan *stemming* dan *stopwords removal*. Kemudian yang kedua tanpa menggunakan *stemming* dan *stopwords removal*. Di mana pada proses ini melakukan ekstraksi data yang kemudian hasil dari proses tersebut akan digunakan pada proses *fastText training model*.

Pada proses *FastText Training Model*, sistem melakukan *unsupervised train* yang akan menghasilkan *trained model FastText*. Setelah itu dari *trained model FastText* yang sudah dibuat, maka diubah menjadi vektor agar bisa di klasifikasi pada proses *Classification Using Multilayer Perceptron (MLP) Classifier*. Kemudian setelah data dilatih dan diuji dengan menggunakan proses *Classification Using Multilayer Perceptron (MLP) Classifier*, proses berlanjut pada *Saving Best Model* yang diperuntukan untuk menyimpan model data terbaik yang sudah dilatih dan diuji.

3.3.2 Flowchart Modul Data Preparation

Seperti yang dijelaskan sebelumnya, modul data preparation diskenariokan menjadi dua yang dimana tahap *stopwords removal* dan *stemming* itu bersifat optional. Flowchart Modul Data Preparation dapat dilihat pada Gambar 3.2.

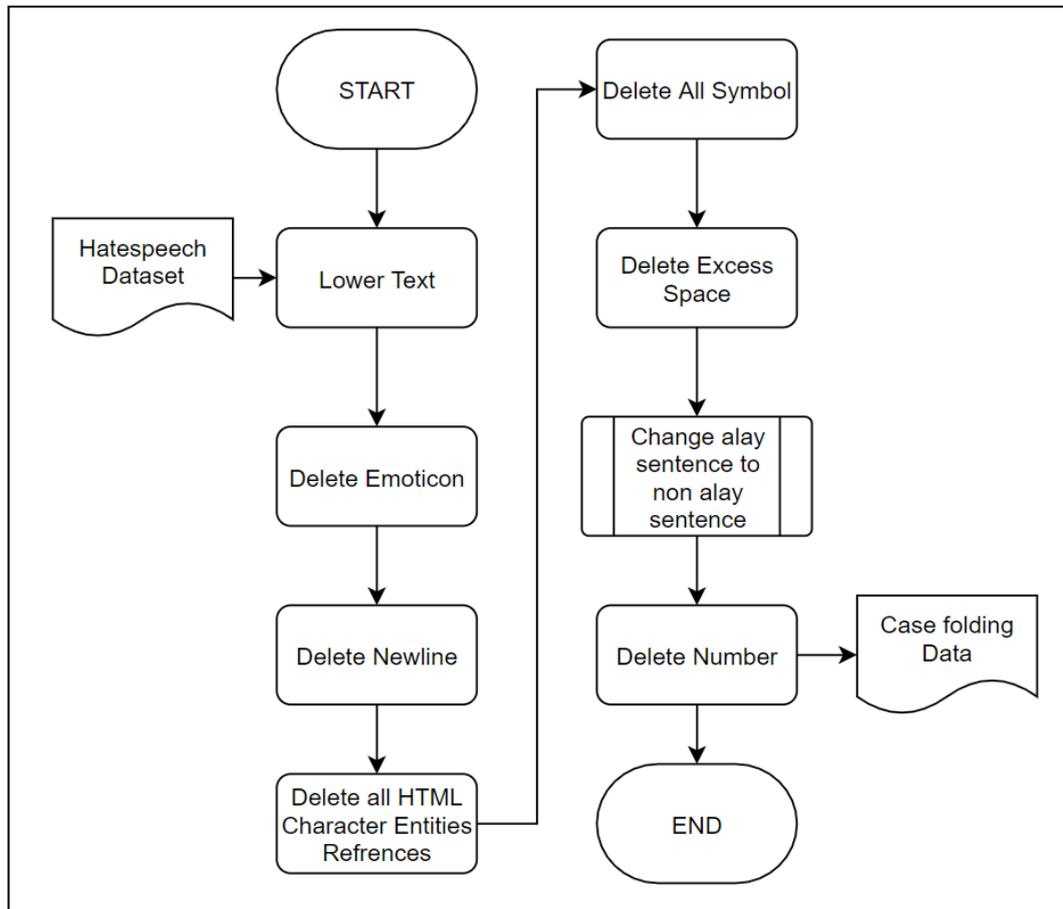


Gambar 3.2 Flowchart Modul Data Preparation

Pada awal pemrosesan data, aplikasi memuat seluruh library dependencies yang dibutuhkan. Setelah itu memuat file dataset hatespeech yang telah dibuat oleh Ibrohim dan Budi. Kemudian proses berlanjut ke text processing. Text processing berlangsung dari proses case folding, proses stopwords removal dan berakhir di proses stemming yang akan disave dan menghasilkan data preparation dalam bentuk txt dan proses alur modul data preparation sudah selesai.

3.3.3 Flowchart Modul Case Folding

Gambar 3.3 merupakan alur proses dari modul case folding, pada proses modul ini juga merupakan bagian dari data preparation.



Gambar 3.3 Flowchart Modul Case Folding

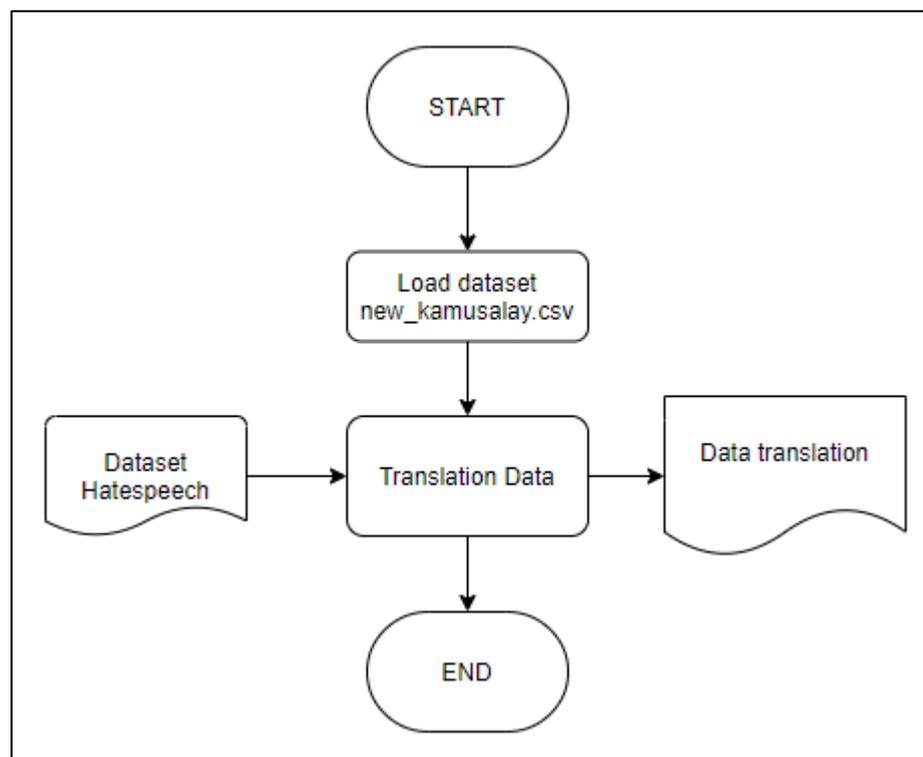
Proses modul case folding dimulai dari menerima dataset hatespeech dalam bentuk json dan akan diproses dengan mengubah semua huruf kapital menjadi huruf kecil. Kemudian proses akan berlanjut dengan menghapus semua karakter emoticon setiap tweet di twitter yang ada di dataset hatespeech. Lalu proses dilanjutkan dengan menghapus semua newline, menghapus seluruh HTML character entities references seperti &, , dan lain-lainnya.

Setelah itu proses berlanjut kepada proses penghapusan seluruh tanda baca atau karakter yang bukan alfabet dan huruf. Setelah itu alur proses dilanjutkan

dengan menghapus semua data yang memiliki kelebihan spasi. Setelah itu proses berlanjut ke modul pergantian kalimat alay menjadi kalimat yang tidak alay, yang kemudian setelah data telah diproses kalimat alay-nya maka akan dilanjutkan dengan penghapusan angka dalam data hatespeech tersebut. Setelah seluruh proses dilewati, maka proses akan menghasilkan data case folding dalam bentuk txt, dan proses dalam modul case folding telah selesai.

3.3.4 Flowchart Modul Transform Alay Sentence

Gambar 3.4 merupakan alur proses dari modul Transform Alay Sentence, pada proses modul ini juga merupakan bagian dari data preparation.



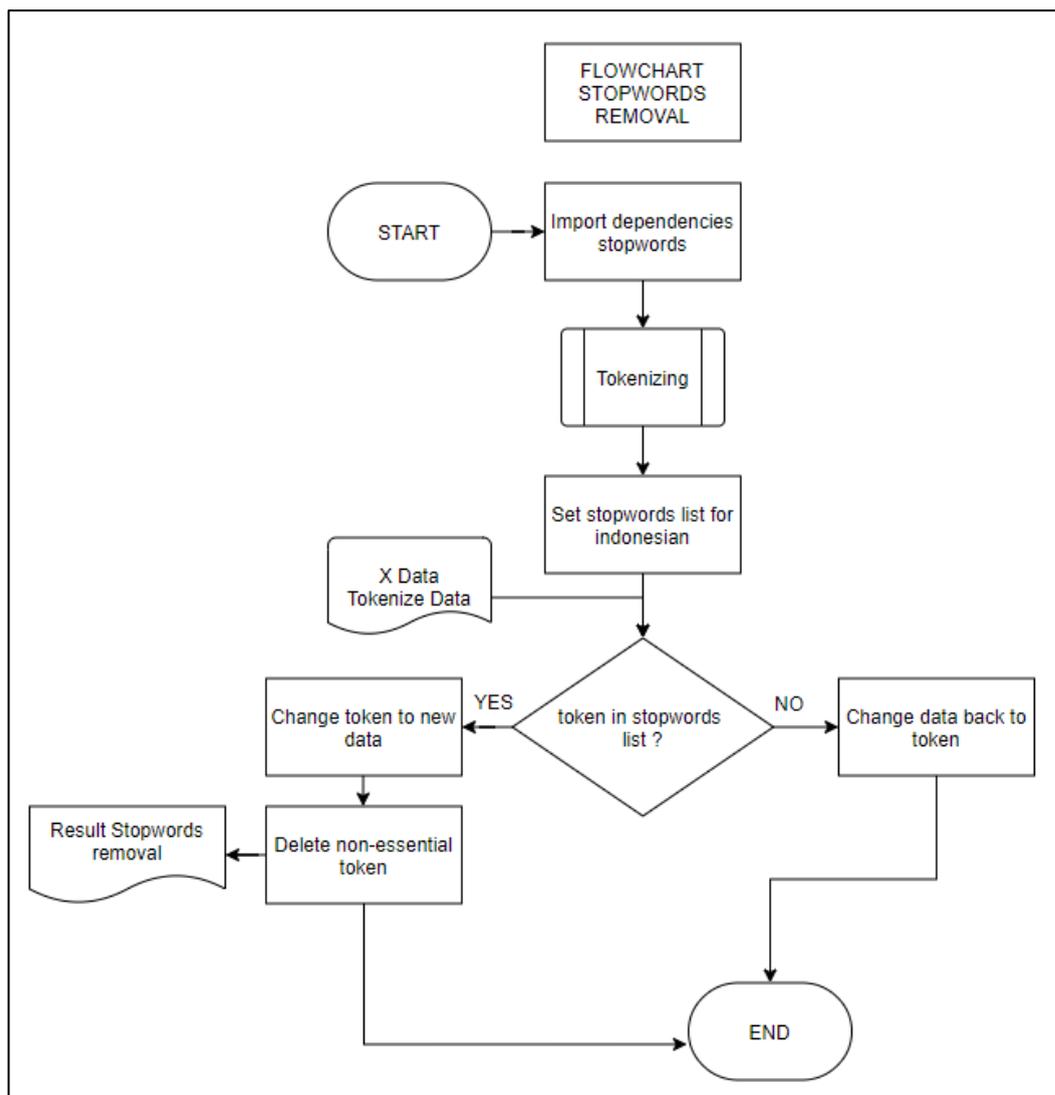
Gambar 3.4 Flowchart Modul Transform Alay Sentence

Proses dimulai pada saat memuat data dari dataset new_kamusalay.csv. Pada proses ini memberikan header origin dan dest serta mengubah new_kamusalay.csv menjadi data json. Setelah proses itu dilakukan, selanjutnya yaitu mengganti semua

kalimat alay yang ada dalam dataset hatespeech yang telah diproses sebelumnya dengan kamus alay tersebut. Setelah semua proses itu dilakukan, maka akan menghasilkan data yang baru yaitu data translation dalam bentuk txt dan proses dari modul transform alay sentences telah selesai.

3.3.5 Flowchart Modul Stopwords Removal

Gambar 3.5 merupakan alur proses dari modul stopwords removal, pada proses modul ini juga merupakan bagian dari data preparation.

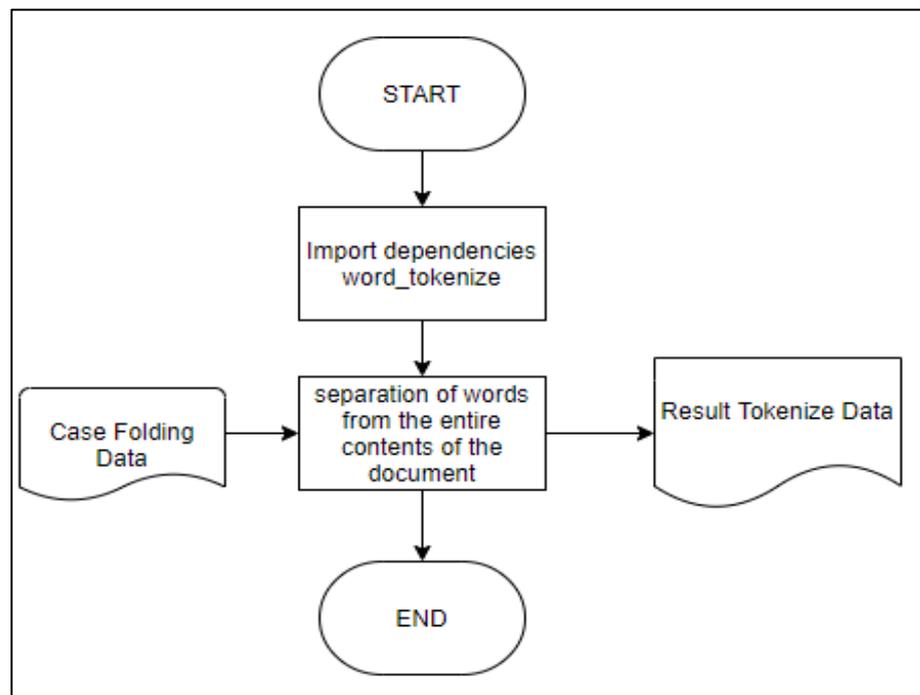


Gambar 3.5 Flowchart Modul Stopwords Removal

Alur dimulai dengan mengimpor semua library dependencies stopwords, lalu berlanjut ke modul tokenizing. Setelah proses tokenisasi dilakukan, data dari hasil tokenisasi akan diperiksa apakah token-token tersebut itu terdaftar dalam stopwords list? Ketika tidak termasuk stopwords list, maka kata dalam data tersebut akan kembali menjadi token dan proses selesai. Namun, jika token dalam data tersebut terdaftar, maka proses berlanjut ke proses yang mengubah token menjadi data baru, lalu dilanjutkan dengan menghapus token yang tidak penting berdasarkan daftar yang ada di dalam list stopwords, kemudian proses ini menghasilkan data dari stopwords removal dalam bentuk txt dan proses dari modul stopwords removal selesai.

3.3.6 Flowchart Modul Tokenizing

Gambar 3.6 merupakan alur proses dari modul Tokenizing, pada proses modul ini juga merupakan bagian dari data preparation.

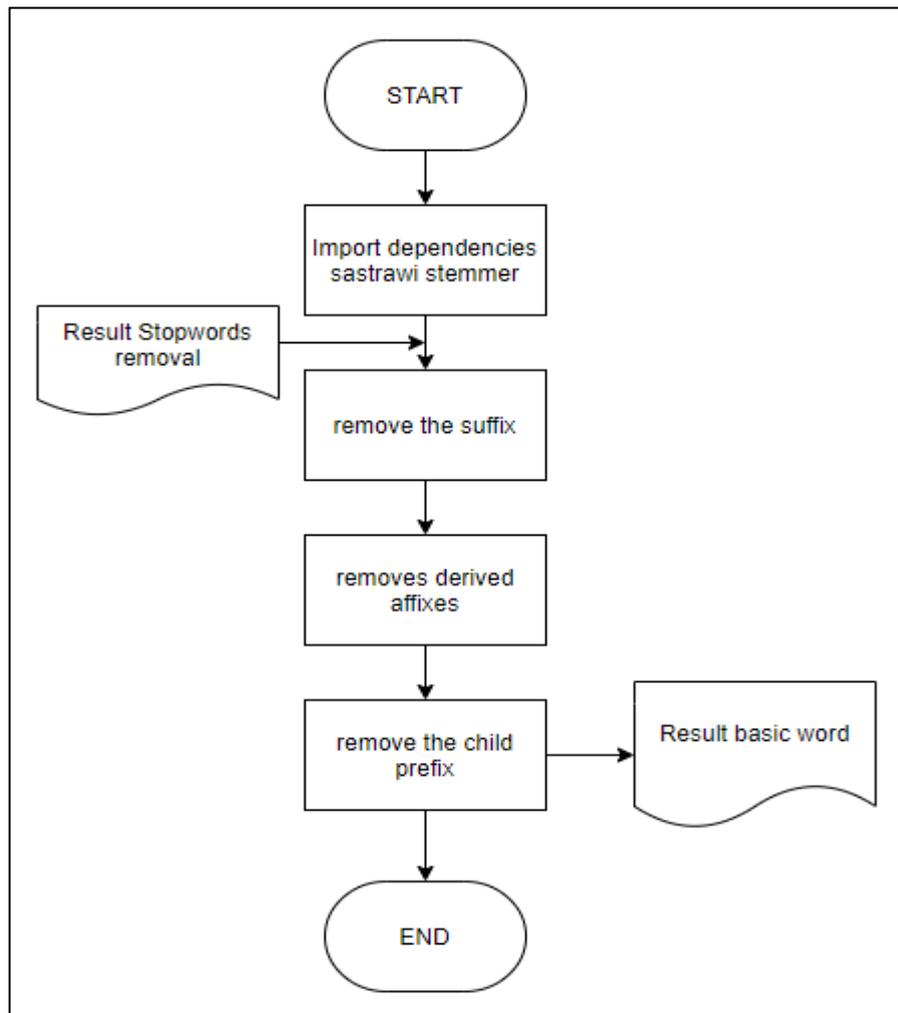


Gambar 3.6 Flowchart Modul Tokenizing

Flowchart modul Tokenisasi dimulai dengan mengimpor library dependencies word_tokenze, yang kemudian proses untuk melakukan pemisahan kata atau pemotongan string input dari seluruh data dari hasil proses case folding. Ketika proses sudah menyelesaikan pemisahan kata, maka akan menghasilkan data baru hasil tokenisasi dan proses modul selesai.

3.3.7 Flowchart Modul Stemming

Gambar 3.7 merupakan alur proses dari modul stemming, pada proses modul ini juga merupakan bagian dari data preparation.



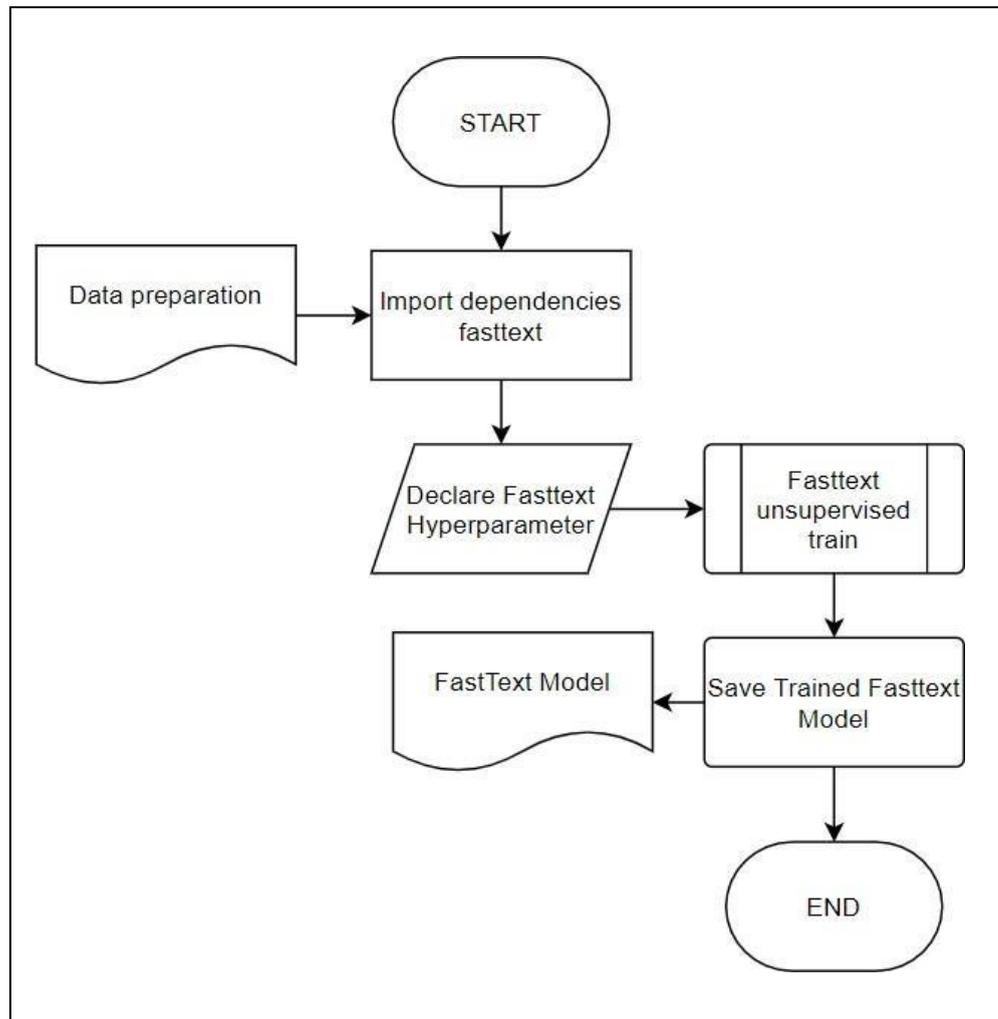
Gambar 3.7 Flowchart Modul Stemming

Flowchart dimulai dengan mengimpor sastrawi stemmer. Sastrawi merupakan salah satu library stemming bahasa Indonesia terbaik yang dapat diakses oleh umum. Kemudian proses berlanjut dengan menghilangkan sufiks yang ada pada data yang merupakan hasil dari proses stopwords removal.

Setelah itu proses akan menghilangkan seluruh imbuhan turunan pada data, lalu proses berlanjut ke langkah menghilangkan awalan turunan dan proses akan menghasilkan data yang baru yaitu kata dasar dalam bentuk txt. Kemudian proses alur modul stemming selesai begitu juga dengan alur keseluruhan preparation data selesai.

3.3.8 Flowchart Modul FastText Training Model

Setelah melewati data preparation, alur sistem memasuki ke tahap proses fastText training Model. Gambar 3.8 merupakan alur dari proses fastText training Model yang terdiri dari memuat data preparation dalam bentuk txt.

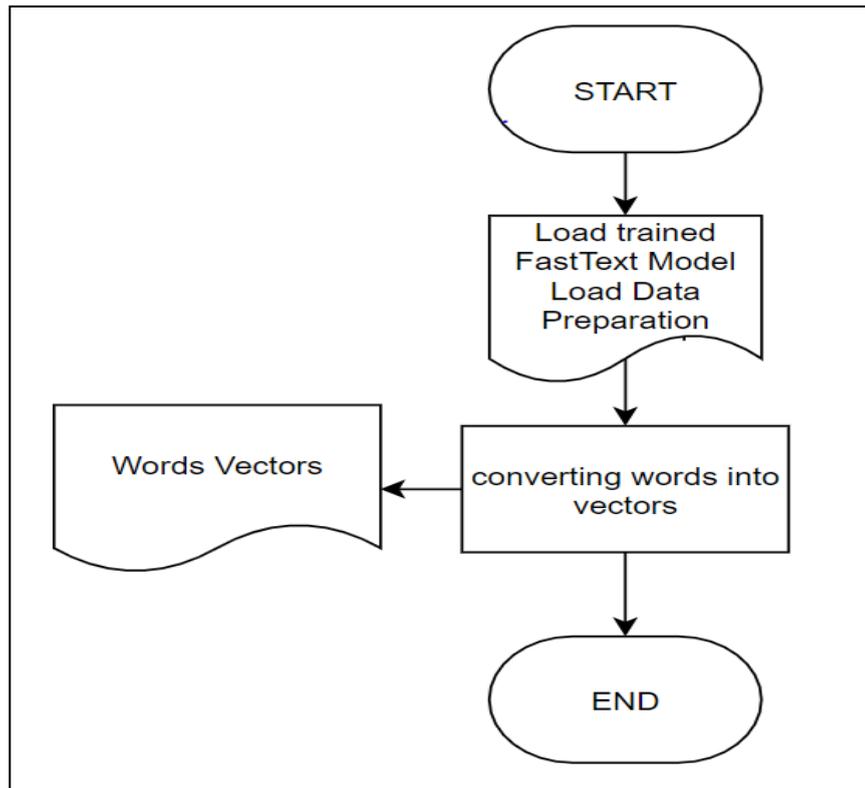


Gambar 3.8 Flowchart Fasttext Training Model

Setelah memuat file dari data preparation, proses selanjutnya adalah mengimpor library dependencies fastText. Kemudian proses selanjutnya adalah deklarasi fasttext hyperparameter seperti minCount, dim dan bucket. Setelah itu proses dilanjutkan dengan fastText unsupervised train. Setelah dilakukannya fastText unsupervised train, proses dilanjutkan dengan menyimpan hasil model yang telah dilatih dan hasil model tersebut akan menghasilkan file model fastText yang berupa bin dan proses modul fastText training model telah selesai.

3.3.9 Flowchart Modul Vectorized Word Embedding

Setelah melakukan proses fastText Training Model, alur sistem memasuki tahap proses Vectorized Word Embedding pada Gambar 3.9.

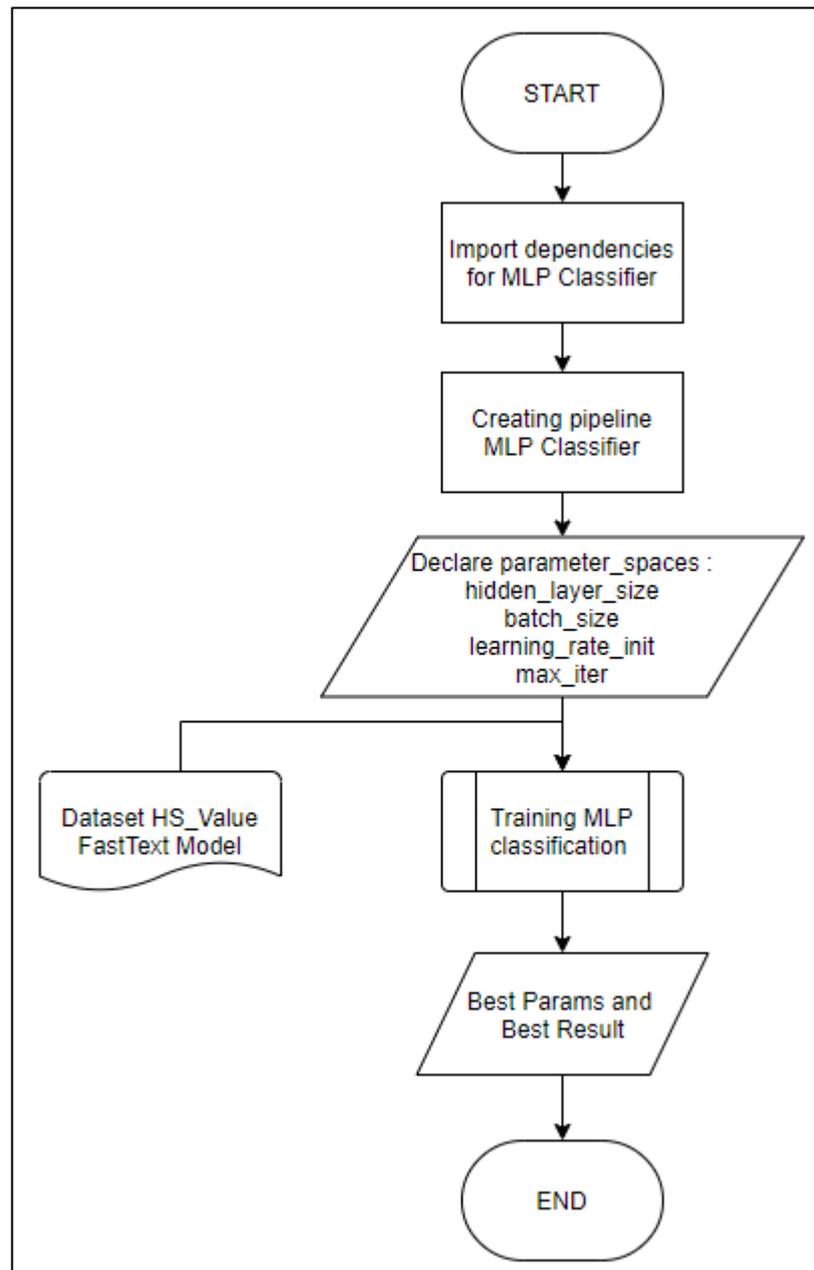


Gambar 3.9 Flowchart Vectorized Word Embedding

Pada tahapan Vectorized Word Embedding, data yang sudah di-training oleh fastText akan diloat terlebih dahulu. Kemudian dataset yang telah dilakukan proses preprocessing akan diubah menjadi vektor. Setelah proses dilakukan, maka proses ini akan menghasilkan word vectors yang akan digunakan untuk proses training model multilayer perceptron classifier dan proses modul Vectorized Word Embedding sudah selesai.

3.3.10 Flowchart Modul Training Multilayer Perceptron Classification

Gambar 3.10 merupakan flowchart yang menjelaskan proses training multilayer perceptron classification.



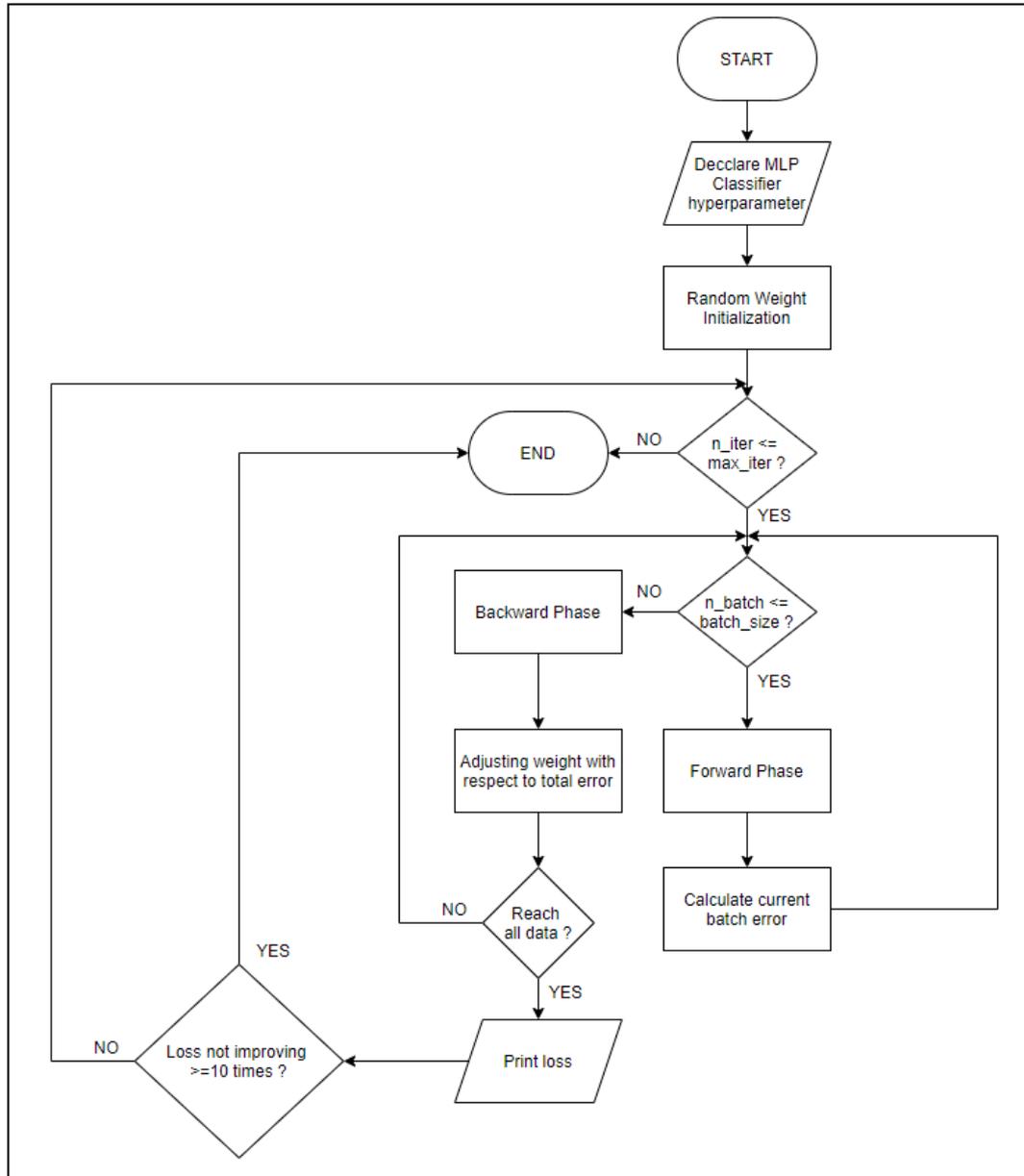
Gambar 3.10 Flowchart Modul Training Multilayer Perceptron

Classification Pertama dilakukannya proses memuat seluruh dependencies library yang digunakan untuk Multilayer Perceptron Classification. Setelah itu

dilakukannya proses untuk membuat pipeline dengan scaler 0 sampai 1. Kemudian dilakukannya proses pembuatan `parameter_space` atau deklarasi hyperparameter dengan menggunakan `hidden_layer_space`, `batch_size`, `learning_rate`, `init`, `max_iter`. Setelah itu proses dilanjutkan dengan training model multilayer perceptron classification yang akan menghasilkan hasil terbaik dan parameter terbaik dari `parameter_spaces` dan modul training multilayer perceptron classification sudah selesai.

3.3.11 Flowchart Modul Multilayer Perceptron

Gambar 3.11 merupakan flowchart yang menjelaskan tentang proses lebih detail untuk klasifikasi menggunakan MLP Classifier.



Gambar 3.11 Flowchart Modul Multilayer Perceptron

Proses pertama dimulai dengan inialisasi hyperparameter yang digunakan untuk proses klasifikasi. Setelah itu bobot pada jaringan syaraf akan diinisialisasi dengan bilangan kecil secara acak. Setelah itu dilanjutkan dengan Forward phase dimana pada langkah pelatihan model ini dilakukan untuk menghitung semua

keluaran di unit tersembunyi dan juga menghitung semua keluaran jaringan di unit keluaran dengan hanya meneruskan input ke model dengan mengkalikannya dengan bobot pada setiap layer dan menemukan output yang dihitung model. Tahapan ini juga akan menghasilkan akumulasi error yang berdasarkan batch size yang sudah ditentukan pada saat inisialisasi hyperparameter.

Setelah batch size yang sudah ditentukan terpenuhi tahap selanjutnya adalah Backward phase yaitu tahap penyesuaian atau perbaikan bobot koneksi berdasarkan hasil eror akumulatif yang didapat pada Forward phase dalam jaringan sehingga meminimalkan ukuran perbedaan antara vektor output aktual dari net dan vektor output yang diinginkan. Setelah backward phase dilakukan maka kemudian akan kembali dilakukan Forward Phase untuk batch data berikutnya lalu Backward Phase lagi hingga seluruh data ter-cover.

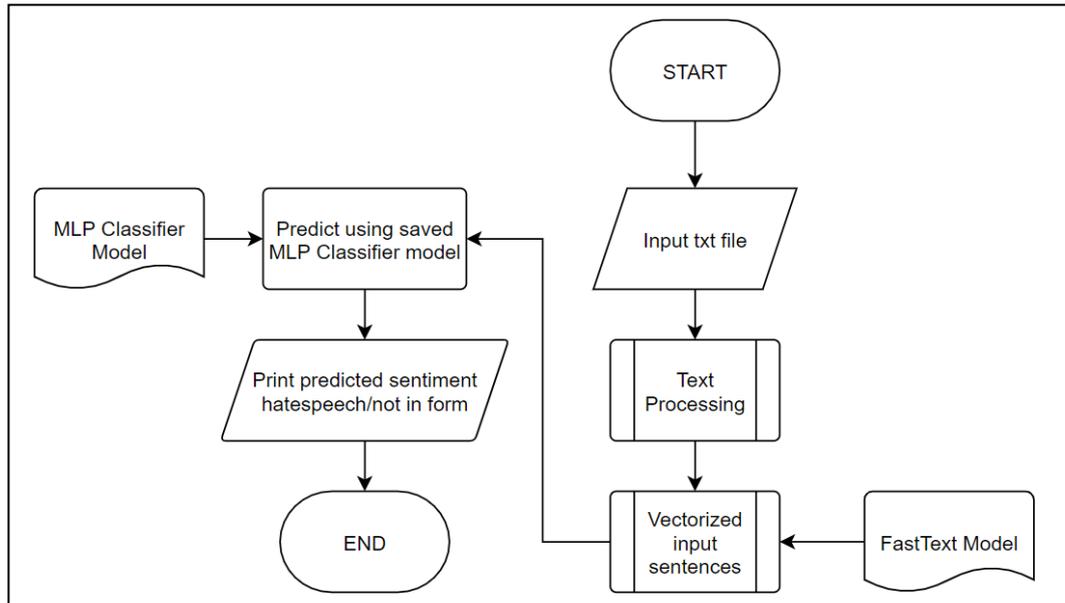
Seluruh proses sebelumnya akan dilakukan sebanyak jumlah epoch atau iterasi yang ditentukan pada saat inisialisasi hyperparameter untuk menemukan hasil loss hingga mendekati 0. Jika loss dalam pengulangan epoch tidak ada perbaikan selama 10 iterasi berturut turut maka proses training akan dianggap selesai.

3.3.12 Flowchart Aplikasi Hatespeech Classification

Model klasifikasi training yang telah dilakukan akan dievaluasi dan dijalankan dalam aplikasi website agar lebih mudah dipahami. Dalam aplikasi web ini akan ada dua jenis input yang bisa digunakan untuk menggambarkan hasil prediksi dari algoritma Multilayer Perceptron Classifier yaitu dengan memasukan kalimat ke dalam textbox dan berupa file .txt yang telah diunggah ke dalam

website menggunakan form input file yang akan menghasilkan sentimen hatespeech atau non hatespeech.

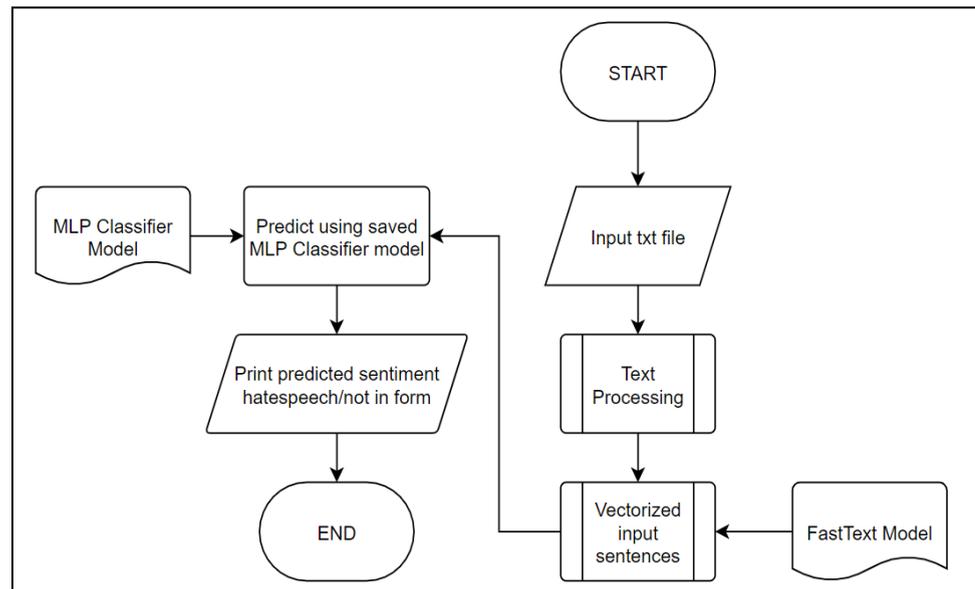
A. Flowchart Prediksi Input Kalimat Tweet



Gambar 3.12 Flowchart Prediksi Input Kalimat Tweet

Alur flowchart yang ditunjukkan pada Gambar 3.12 menjelaskan bahwa proses untuk memprediksi input dari kalimat tweet yang diberikan melalui text box pada aplikasi website. Setelah memasukan kalimat kemudian menekan tombol predict. Setelah itu aplikasi web akan menjalankan proses yang sama seperti pada proses klasifikasi yang telah dijelaskan sebelumnya tanpa melalui proses training karena proses prediksi telah menggunakan model dari hasil proses training algoritma multilayer perceptron classifier yang telah dilakukan. Setelah itu halaman website akan menampilkan hasil dari prediksi kalimat tweet tersebut hatespeech atau non hatespeech.

B. Flowchart Prediksi File Kalimat Tweet

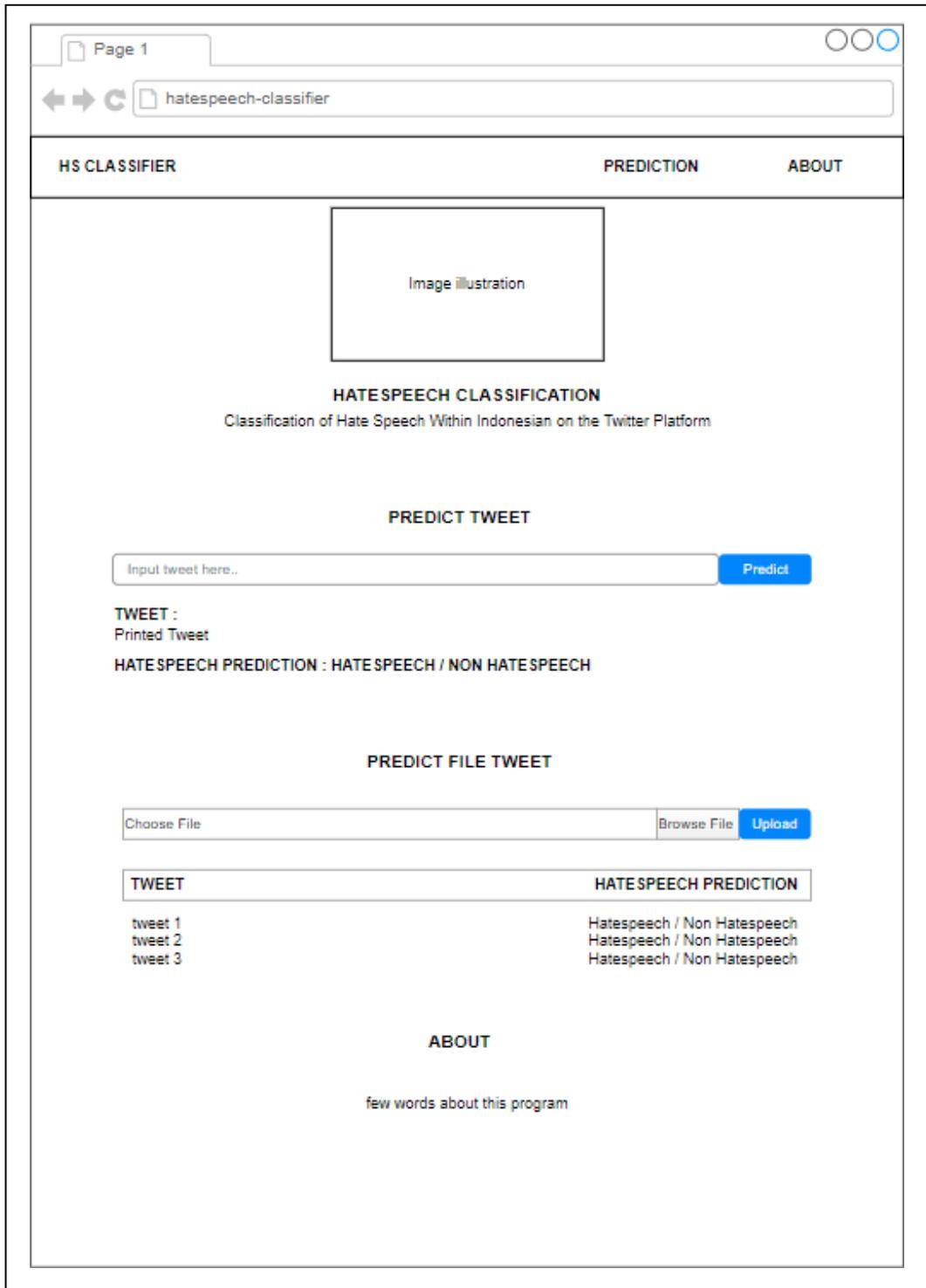


Gambar 3.13 Flowchart Prediksi File Kalimat Tweet

Alur diagram pada Gambar 3.13 adalah proses dari prediksi dengan memasukkan file .txt yang diunggah ke dalam aplikasi website. Ketika kita telah selesai mengunggah file tersebut, kemudian tekan tombol upload. Setelah itu aplikasi web akan menjalankan proses yang sama seperti pada proses klasifikasi yang telah dijelaskan sebelumnya tanpa melalui proses training karena proses prediksi telah menggunakan model dari hasil proses training algoritma multilayer perceptron classifier yang telah dilakukan. Setelah itu halaman web akan menampilkan hasil dari prediksi dari file txt yang berisi kalimat tweet tersebut dalam bentuk table form dan proses selesai.

3.4 Rancangan Antarmuka Aplikasi Website Hatespeech Classifier

Gambar 3.14 menunjukkan rancangan antarmuka pada aplikasi website hatespeech classifier.



Gambar 3.14 Rancangan Antarmuka Aplikasi Website Hatespeech Classification

Rancangan antar muka akan dibuat untuk single web page app, dimana semua bagian dari aplikasi dijadikan dalam satu halaman. Dimana pada tampilan tersebut terdapat beberapa pilihan menu, diantara lain adalah prediction dan about. Rancangan antarmuka aplikasi hatespeech classifier hanya dibuat dengan satu halaman. Pada awal tampilan pengguna akan diperlihatkan menu dari aplikasi hatespeech yaitu prediction dan about. Lalu akan diperlihatkan pula banner ilustrasi hatespeech dan penjelasan singkat tentang aplikasi web.

Ketika menu prediction ditekan, pengguna akan diperlihatkan tampilan untuk prediksi kalimat tweet hatespeech atau bukan. Pada tampilan ini terdapat dua jenis prediksi diantara lain adalah prediksi dengan cara memasukan kalimat tweet dan prediksi dengan cara mengunggah file txt ke dalam form yang ada. Apabila pengguna memilih untuk memasukan kalimat tweet, maka pengguna akan diperlihatkan tampilan form yang berupa text box dan tombol predict. Ketika pengguna memasukan kalimat tweet ke dalam text box dan menekan tombol predict, maka aplikasi web akan langsung menampilkan hasil dari prediksi hatespeech atau non hatespeech berdasarkan kalimat tweet yang user masukan.

Namun ketika user memilih untuk memasukan dokumen txt, pengguna dapat mengunggah dokumen berisi .txt. Setelah tombol upload ditekan, maka aplikasi web akan menampilkan isi dari dokumen kumpulan tweet serta prediksi hatespeech atau non hatespeech dari masing-masing kumpulan tweet tersebut. Apabila menu about pengguna tekan, maka pengguna akan ditampilkan tentang beberapa kata mengenai deskripsi singkat tentang aplikasi website hatespeech classifier ini