



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

METODOLOGI PENELITIAN DAN PERANCANGAN SISTEM

3.1 Metodologi Penelitian

Metodologi penelitian yang digunakan dalam penyusunan tugas akhir ini dibagi menjadi tujuh tahap, yaitu tahap studi literatur, pengumpulan data, analisa masalah, perancangan dan pembangunan web untuk Spam Filtering, implementasi algoritma yang digunakan, pengujian dan evaluasi, serta dokumentasi dan penyusunan laporan.

a. Studi literatur

Pada tahap ini dilakukan pembelajaran terhadap teori-teori dengan mencari jurnal, *paper*, serta artikel-artikel terpercaya di Internet yang berhubungan dengan sistem yang akan dibuat.

b. Pengumpulan data

Pada tahap ini dilakukan *scrapping user feedback* untuk mendapatkan *user feedback* terbaru dari aplikasi terkait. Selanjutnya dilakukan proses labelisasi dengan menggunakan metode *majority vote*. *Majority vote* adalah sebuah metode dimana sebanyak 5 orang akan melakukan pelabelan pada seluruh *dataset*. Selanjutnya pada tiap dokumen/kalimat akan dipilih 1 label yang dipilih sebanyak 3 kali atau lebih. Contohnya apabila kalimat A diberi label oleh 3 orang sebagai *spam*, maka kalimat A akan diberi label *spam*. Total data yang terkumpul sebanyak 900 data dengan komposisi 810 data ham dan 90 data spam.

c. Analisa masalah

Pada tahap ini dilakukan analisis untuk setiap informasi yang telah diperoleh agar mendapatkan pemahaman menyeluruh terhadap masalah dan metode yang digunakan untuk menyelesaikan permasalahan tersebut.

d. Perancangan dan pembangunan web untuk Spam Filtering

Pada tahap ini dilakukan perancangan dan pembangunan aplikasi dan sistem yang akan dibuat. Perancangan yang dimaksud meliputi pembuatan flowchart, dan rancangan antarmuka. Setelah itu dilakukan pembuatan rancangan web yang menerima *input* berupa list dari *user feedback* dan akan mengembalikan list dari *user feedback* yang telah diklasifikasi dalam bentuk file csv.

e. Implementasi algoritma yang digunakan

Pada tahap ini dilakukan implementasi dari algoritma yang digunakan pada sistem klasifikasi *user feedback* untuk *Spam Filtering*. Selain itu dilakukan juga implementasi dari algoritma-algoritma pendukung seperti TF/IDF, *text preprocessing*, dan lain-lain.

f. Pengujian dan evaluasi

Pada tahap ini akan dilakukan pengujian akurasi serta performa dari program. Evaluasi akan dilakukan dengan menghitung akurasi dan menggunakan *Confusion Matrix* untuk mengukur performa program dengan menghitung *precision*, *recall*, serta F1-Score. Selanjutnya akan dilakukan evaluasi berdasarkan hasil performa yang telah didapatkan untuk selanjutnya dilakukan perbaikan terhadap program.

g. Dokumentasi dan penyusunan laporan

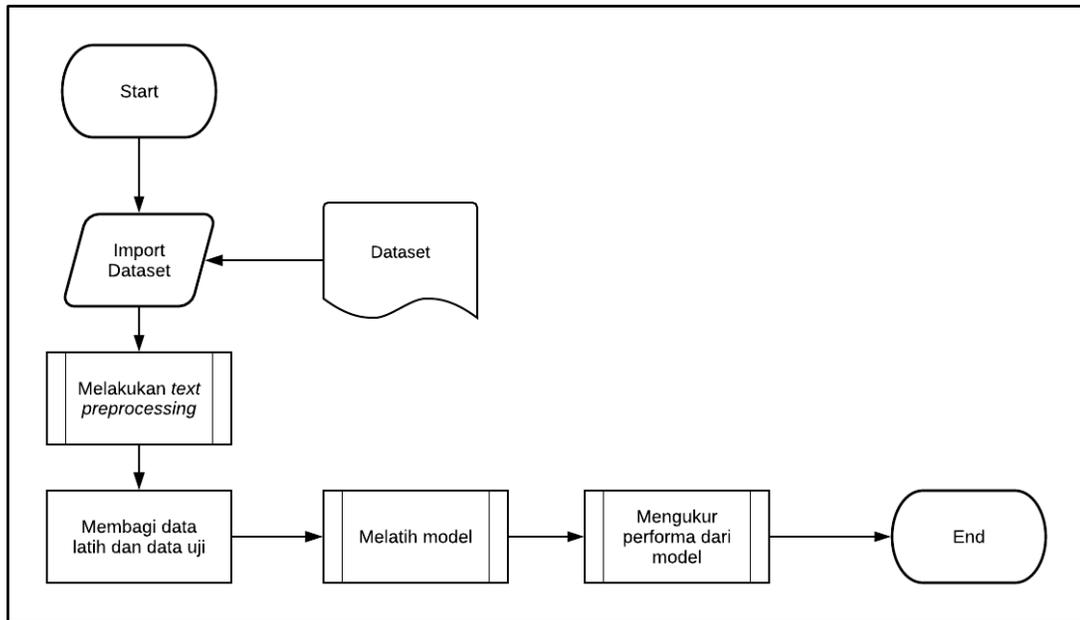
Penyusunan laporan dilakukan sejak awal penelitian dimulai. Sedangkan dokumentasi program dilakukan setelah program yang telah dibuat diperkirakan sudah memenuhi kriteria dan memiliki akurasi serta performa yang memuaskan.

3.2 Perancangan aplikasi

Perancangan aplikasi yang dilakukan meliputi pembuatan *flowchart*, dan rancangan antarmuka.

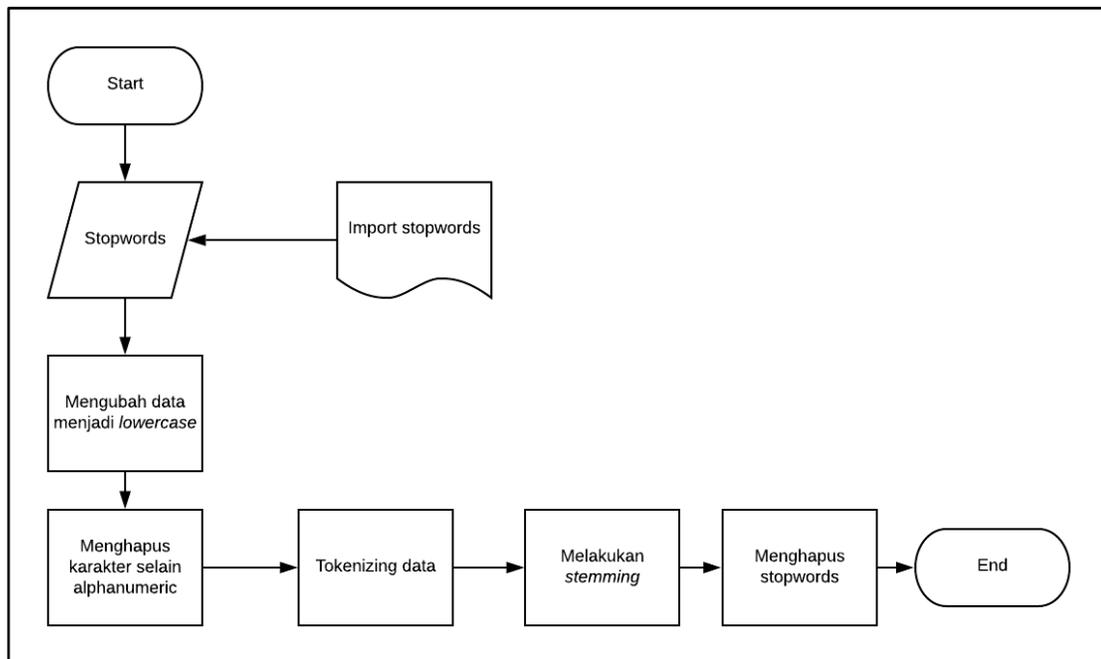
3.2.1 Flowchart

Flowchart merupakan diagram yang menunjukkan alur kerja dari program. Pada Gambar 3.1, ditunjukkan *flowchart* aktivitas *training* data yang dilakukan untuk mendapatkan model yang diinginkan. Langkah pertama yang dilakukan adalah dengan memasukkan *dataset* yang telah dilabeli *spam* atau *non-spam*. *Dataset* yang digunakan berjumlah 900 data. Langkah pertama yang dilakukan adalah melakukan proses *preprocessing text*. Setelah itu dilakukan pemisahan *dataset* menjadi data latih dan data uji. Selanjutnya dilakukan proses pemodelan data train, dan pengujian pada model yang telah dibuat.



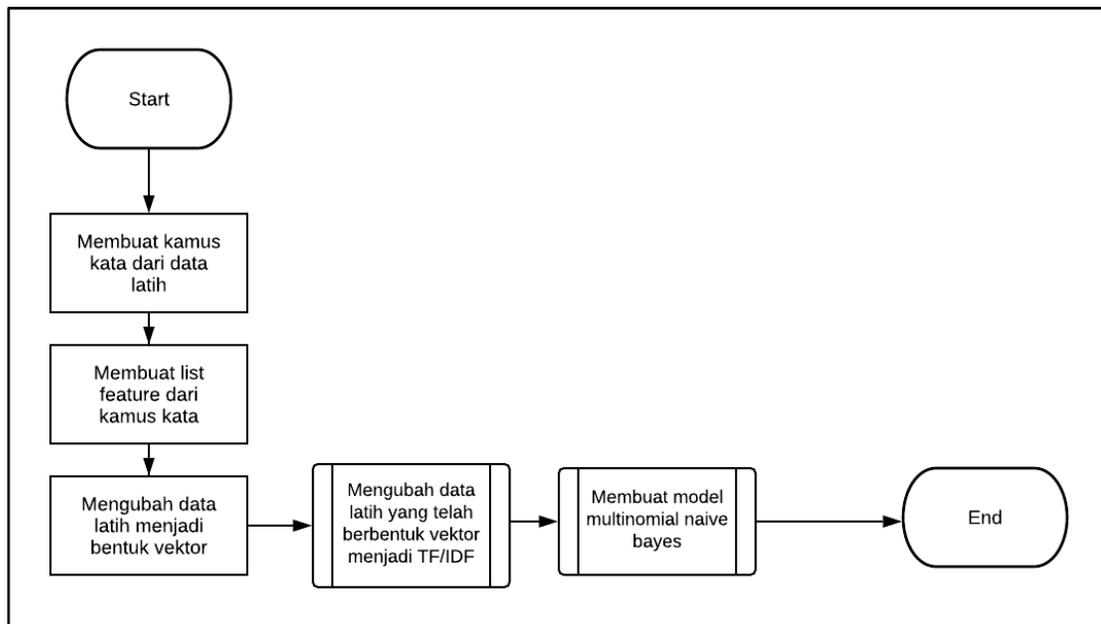
Gambar 3.1 Flowchart pembuatan model

Selanjutnya pada Gambar 3.2 dijelaskan proses *text preprocessing*. Proses *text preprocessing* dimulai dengan mengubah seluruh kata menjadi *lowercase*. Lalu dilakukan proses menghapus karakter selain alphanumerik. Setelah itu dilakukan proses *tokenizing* yaitu proses memisahkan kata per kata. Setelah proses *tokenizing* selesai dilakukan proses *stemming*, yaitu proses mengubah kata menjadi kata dasar. Terakhir dilakukan pembuangan kata-kata yang tidak dibutuhkan yaitu *stop words* yang berisi kata-kata yang tidak berpengaruh pada penelitian seperti saya, kamu, dan lain-lain.



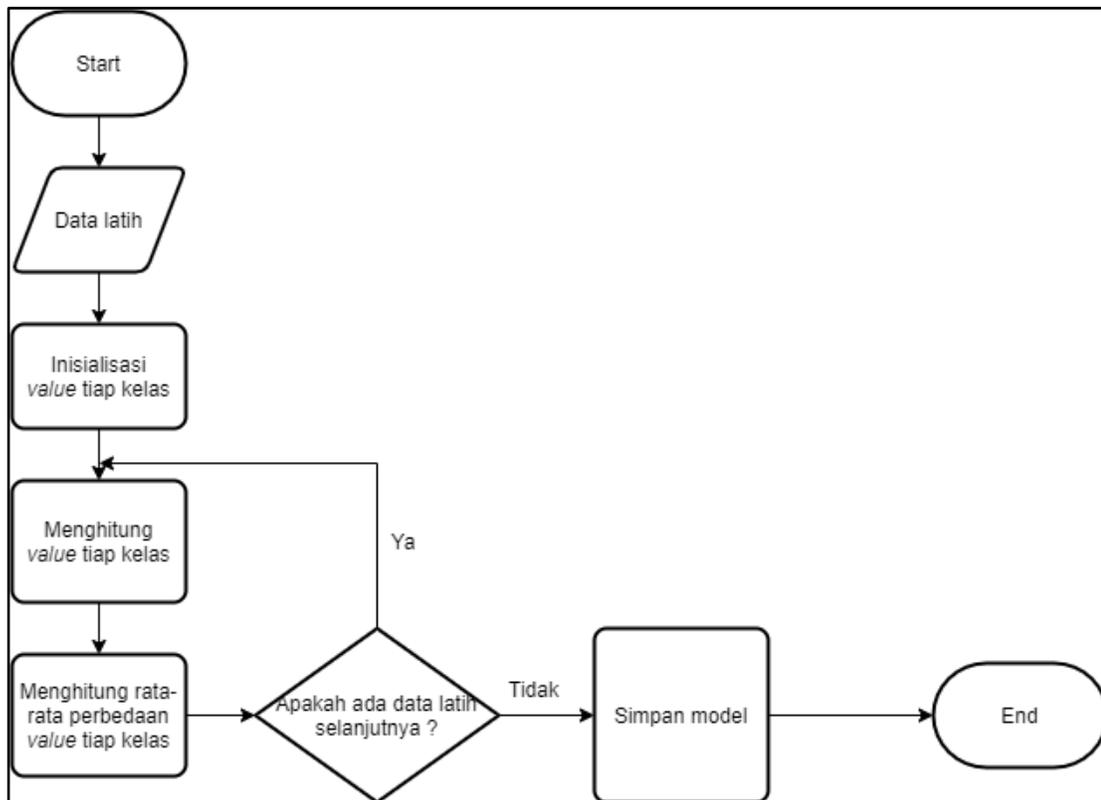
Gambar 3.2 Flowchart *text preprocessing*

Selanjutnya dilakukan pembuatan model untuk data latih seperti pada Gambar 3.3. Langkah pertama yang dilakukan adalah membuat kamus data dari data latih yang telah dibuat. Dari kamus kata tersebut, diambil kata-kata yang memiliki kemunculan lebih tinggi dari *cut off frequency* yang ditentukan untuk menjadi *features*. Setelah itu data latih akan diubah bentuknya menjadi bentuk vektor. Selanjutnya data yang telah berbentuk vektor akan diubah menjadi TF/IDF dengan menggunakan *library* TF/IDF Transformer. Langkah terakhir yang dilakukan adalah membuat model *multinomial naive bayes* dengan menggunakan data latih yang telah dibuat.



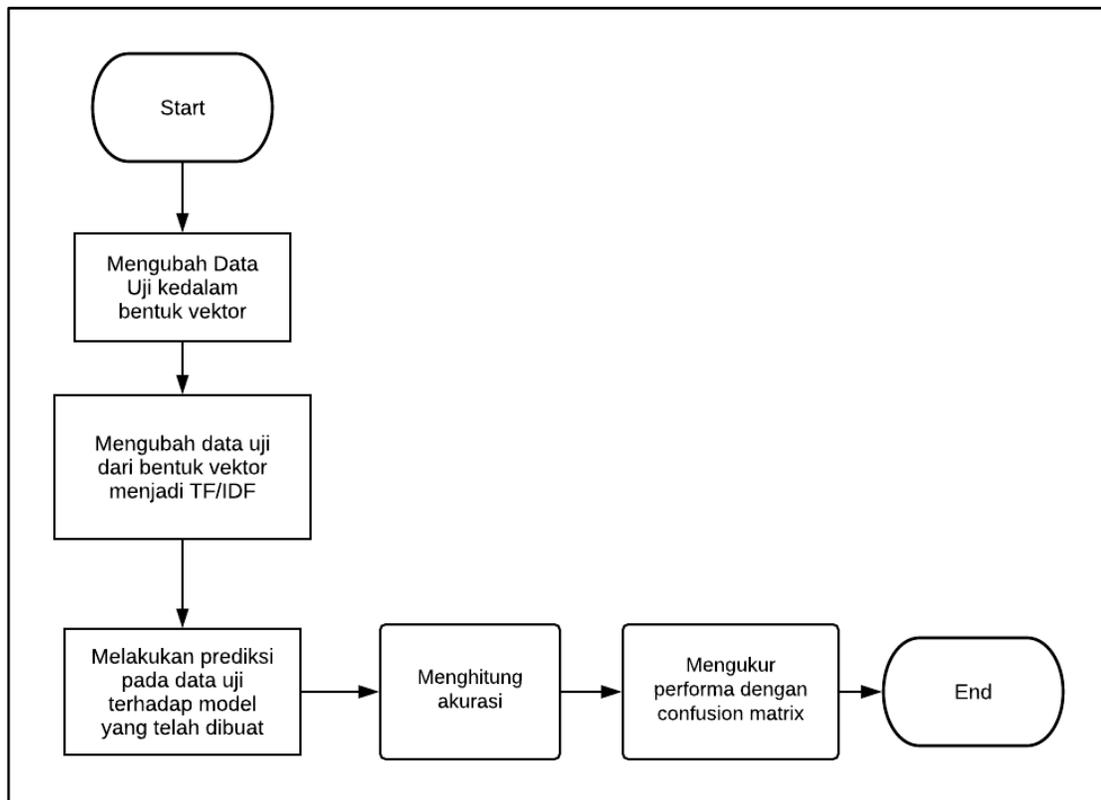
Gambar 3.3 Flowchart pemodelan data latih

Pada Gambar 3.4 dijabarkan proses pembuatan model *multinomial naive bayes* dengan menggunakan data latih yang telah dibuat sebelumnya. Langkah pertama yang dilakukan adalah melakukan inialisasi *value* tiap kelas. Setelah itu dilakukan perulangan terhadap data latih tersebut dan melakukan perhitungan *value* pada setiap perulangan. Setelah perulangan berakhir, model yang telah dibuat tersebut disimpan untuk selanjutnya digunakan dalam proses prediksi.



Gambar 3.4 Flowchart multinomial naive bayes

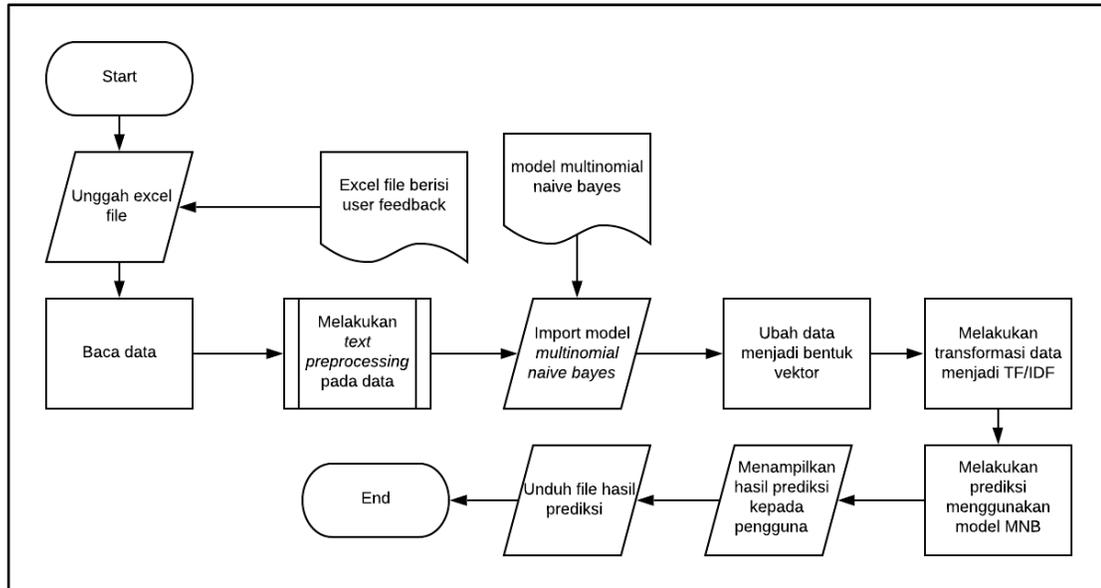
Pada Gambar 3.5 dijelaskan alur dari proses pengujian performa dari model yang telah dibuat. Langkah pertama yang dilakukan adalah mengubah bentuk data uji menjadi bentuk vektor terlebih dahulu. Selanjutnya data tersebut diubah menjadi TF/IDF dengan menggunakan *library* TF/IDF. Lalu dilakukan proses prediksi dengan menggunakan model *multinomial naive bayes* yang telah dibuat sebelumnya. Dari hasil prediksi tersebut, dilakukan proses perhitungan akurasi dari data latih maupun data uji. Setelah itu dapat dilakukan pembuatan *confusion matrix* yang dapat digunakan untuk mengukur performa dari model *multinomial naive bayes*.



Gambar 3.5. Flowchart pengujian pada model

Untuk memudahkan proses penggunaan, maka sistem klasifikasi dibuat kedalam aplikasi web. Flowchart dari aplikasi web yang dibuat dapat dilihat pada Gambar 3.6. Langkah pertama adalah pengguna mengunggah file berisi *user feedback* yang berbentuk file .xlsx atau .csv kedalam aplikasi. Setelah itu data yang diunggah akan dibaca oleh aplikasi. Selanjutnya dilakukan proses *text preprocessing* pada data tersebut. Selanjutnya dilakukan *import* model *multinomial naive bayes* yang telah dibuat sebelumnya. Langkah selanjutnya adalah mengubah data yang sebelumnya telah dilakukan *text preprocessing* tersebut kedalam bentuk vektor. Lalu data tersebut akan di-transformasi menjadi TF/IDF dengan menggunakan *library* TF/IDF Transformer. Selanjutnya dilakukan prediksi dengan menggunakan model *multinomial naive bayes*.

Setelah prediksi selesai, maka hasil prediksi akan ditampilkan pada halaman web agar dapat dilihat oleh pengguna. Terakhir, pengguna dapat mengunduh hasil klasifikasi tersebut yang berbentuk file dengan ekstensi .csv.

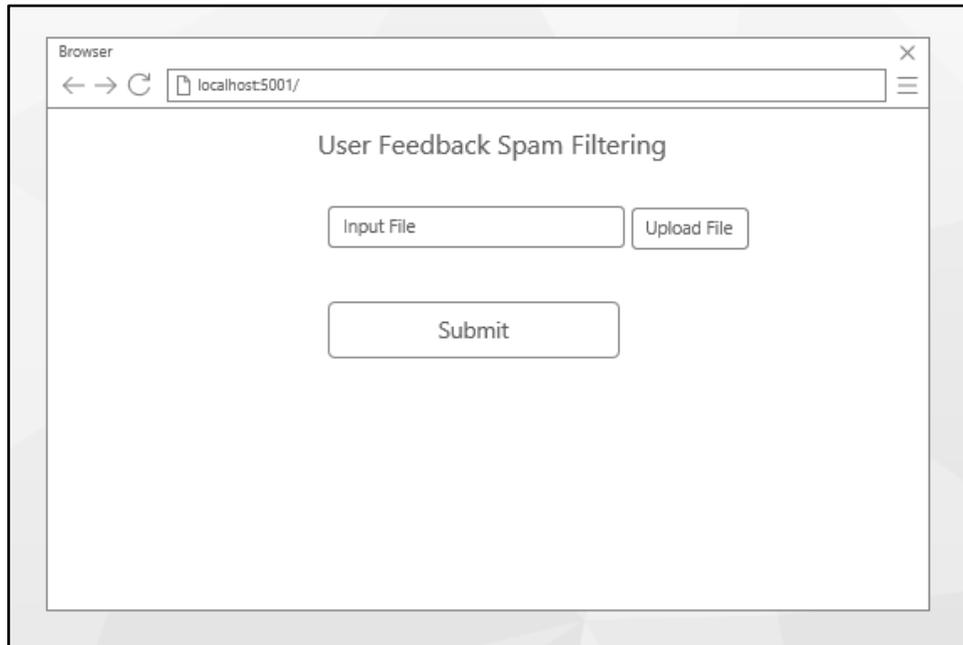


Gambar 3.6. Flowchart klasifikasi dengan aplikasi web

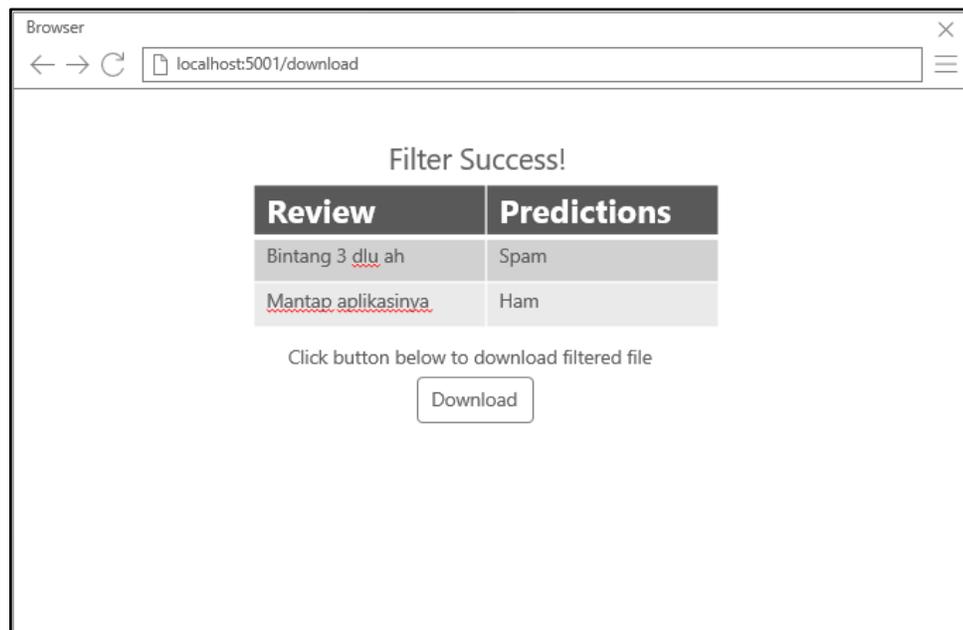
3.2.2 Rancangan Antarmuka

Pada perancangan antarmuka ini, digunakan add-ons pada Microsoft Powerpoint yaitu PowerMockup untuk pembuatan rancangan antarmuka. Gambar 3.7 merupakan rancangan halaman saat pengguna akan mengunggah file berisi *user feedback* yang ingin dilakukan *spam filter*. Pengguna hanya bisa mengunggah file berekstensi csv. Setelah pengguna menekan tombol Submit, maka program akan berjalan dan melakukan *spam filtering*. Selanjutnya pengguna akan diarahkan ke halaman *download* atau unduh seperti pada Gambar 3.8. Pada halaman tersebut pengguna dapat mengunduh hasil klasifikasi yang telah dilakukan. Hasil klasifikasi

tersebut merupakan file dengan ekstensi csv. Selain itu pengguna juga dapat melihat hasil klasifikasi dari data yang telah diinput sebelumnya.



Gambar 3.7 Mockup halaman upload file



Gambar 3.8 Mockup halaman download hasil klasifikasi