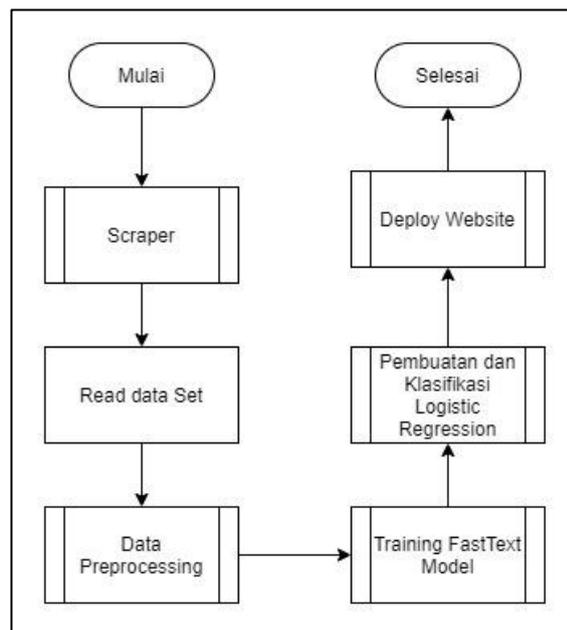


BAB 3

METODOLOGI PENELITIAN

3.1 Gambaran Umum Metodologi Penelitian

Gambaran umum metodologi penelitian akan menjelaskan bagaimana alur dari proses penelitian yang dilakukan dan spesifikasi perangkat yang digunakan. Gambaran umum dari metodologi penelitian ini dijabarkan menggunakan *flowchart* yang dilakukan pada penelitian Implementasi *Word Embedding* untuk klasifikasi berita disabilitas menggunakan Logistic Regression, mulai dari proses pengumpulan data di *website* yang telah ditentukan, *Data Preprocessing*, Pelatihan *pre-trained* model dari FastText, pengimplementasian Logistic Regression, hingga membuat *website* yang menggunakan model yang telah dilatih. Proses dari penelitian digambarkan dalam bentuk *flowchart* yang dapat dilihat pada Gambar 3.1.



Gambar 3.1 *Flowchart* Keseluruhan Proses Penelitian

Pada penelitian ini menggunakan perangkat keras dan lunak sebagai berikut.

Hardware:

1. Processor: AMD Ryzen 5 4600 (12 GPUs), ~3.0Hz
2. Graphics Processing Unit: NVIDIA GeForce GTX 1650 Ti
3. RAM: 16.00 GB

Software:

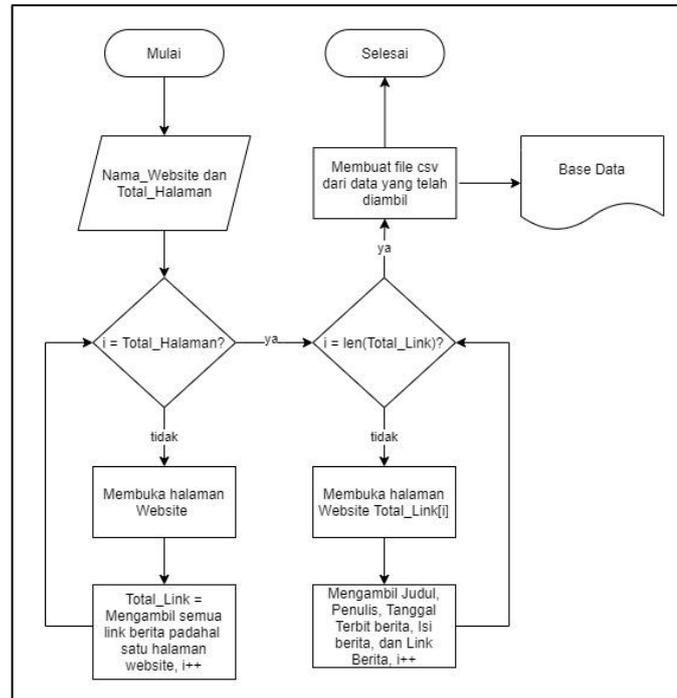
1. OS: Ubuntu 20.04.2.0 LTS
2. Google Chrome
3. Bahasa Pemrograman: Python 3
4. *Framework*: Flask
5. Visual Studio Code
6. Jupyter Notebook

3.2 Studi Literatur

Studi literatur adalah tahap untuk mencari sumber-sumber yang berhubungan dengan penelitian ini seperti Difabel, *Word Embedding*, Logistic Regression, dan *Confussion Matrix*. Tahap ini dilakukan agar dapat memahami secara mendalam teori-teori dan cara implementasi yang dibutuhkan dalam penelitian.

3.3 Pengumpulan Data

Proses pengumpulan data yang dilakukan menggunakan bahasa pemrograman *python* dengan bantuan *library* *urllib*, dan *beautifulsoup*. *Urllib* berguna untuk membuka halaman *website* yang diinginkan, sedangkan *beautifulsoup* untuk mengubah halaman *website* tersebut menjadi bentuk teks dan membaca semua isi dari konten HTML (*HyperText Markup language*) yang ada. Proses *scraping* untuk mengumpulkan data dapat dilihat pada Gambar 3.2.



Gambar 3.2 *Flowchart* Proses *Scraper*

Proses yang dilakukan adalah menentukan target dari *website* yang mau di-*scraping*, dimana pada penelitian ini menggunakan empat *website* yaitu liputan6, kompas, tempo, dan newsdifabel. Setelah menentukan *website* yang mau di-*scraping* selanjutnya menentukan jumlah halaman *website* yang akan diproses, seperti newsdifabel yang mempunyai 41 halaman dengan 10 berita disetiap halamannya. Proses pertama akan mengambil *link* dari semua berita dari setiap halaman yang dibuka. Proses kedua adalah membuka semua link yang telah dikumpulkan lalu mengambil Judul, nama penulis, tanggal berita diterbitkan, isi berita, dan Link berita tersebut. Setelah berhasil mengumpulkan semua data, maka data akan dijadikan *file excel*.

Data yang terkumpul dari proses *scraping* tidak semuanya mempunyai kategorinya seperti *website* kompas, liputan6, dan tempo sehingga data yang

diambil dari *website* tersebut harus melakukan proses pelabelan manual oleh pakar. Data yang digunakan adalah sebanyak 378 data, yang dapat dilihat pada Tabel 3.1.

Tabel 3.1 Dataset Berita Penyandang Disabilitas

Kategori	Jumlah Data Berita
Tokoh	56
International	55
Regional	43
Olahraga	42
Edukasi	38
Nasional	38
Tekno	36
Hukum	35
Lifestyle	35

Hasil *scraping* dibuat menjadi bentuk *file excel*, yang dapat dilihat pada Gambar 3.3.

	A	B	C	D	E	F
		Judul	Isi Berita	Kategori	Date	Link
2	0	Dunia Punya Stevie Wonder, Indonesia pernah lahirkan pedangdut difabel netra bers Tokoh			3 April, 2021	https://www.newsdifabel.com/2021/04/03/dunia-punya-stevie-wonder-indonesia-pernah-lahirkan-pedangdut-difabel-netra-bers-tokoh/
3	1	Dewi Endah Mayasari Tenaga Pendidik permenmendiknas nomor 70 tahun 2009 mewajibkan agar pe Tokoh			11 December, 2020	https://www.newsdifabel.com/2020/12/11/dewi-endah-mayasari-tenaga-pendidik-permenmendiknas-nomor-70-tahun-2009-mewajibkan-agar-pe-tokoh/
4	2	Shannon DeVido Menertawakan Kedi komedi sebagai pendobrak batas karena membuat difak Tokoh			11 March, 2021	https://www.newsdifabel.com/2021/03/11/shannon-devido-menertawakan-kedi-komedi-sebagai-pendobrak-batas-karena-membuat-difak-tokoh/
5	3	Tri Bagio Doktor Difabel Netra di Indo tiga belas tahun yang indah. penuh warna, sarat cerita. r Tokoh			10 February, 2021	https://www.newsdifabel.com/2021/02/10/tri-bagio-doktor-difabel-netra-di-indo-tiga-belas-tahun-yang-indah-penuh-warna-sarat-cerita-r-tokoh/
6	4	Pengalaman Rizwan Hermawan Sebag dalam kajian antropologis terungkap bahwa pada awaln Tokoh			22 January, 2021	https://www.newsdifabel.com/2021/01/22/pengalaman-rizwan-hermawan-sebag-dalam-kajian-antropologis-terungkap-bahwa-pada-awaln-tokoh/
7	5	Saber al-Ashkar: Pejuang Difabel Pale 'pertama, mereka mengambil tanahnya, lalu mereka me Tokoh			1 December, 2020	https://www.newsdifabel.com/2020/12/01/saber-al-ashkar-pejuang-difabel-pale-pertama-mereka-mengambil-tanahnya-lalu-mereka-me-tokoh/
8	6	Melalui Sastra, Opik Melukis Dunia jalan hidup, tuan-puan, memiliki arah yang tak bisa engl Tokoh			26 October, 2020	https://www.newsdifabel.com/2020/10/26/melalui-sastra-opik-melukis-dunia-jalan-hidup-tuan-puan-memiliki-arah-yang-tak-bisa-engl-tokoh/
9	7	Yuk, Kenalan dengan Ibu Neneng Sun [bandung, 28 oktober 2019] buku adalah jendela dunia. l Tokoh			28 October, 2019	https://www.newsdifabel.com/2019/10/28/yuk-kenalan-dengan-ibu-neneng-sun-bandung-28-oktober-2019-buku-adalah-jendela-dunia-l-tokoh/
10	8	Perspektif Daden Sukendar Calon Anj [bandung, 15 agustus 2019] pada tahun ini, komnas pere Tokoh			15 August, 2019	https://www.newsdifabel.com/2019/08/15/perspektif-daden-sukendar-calon-anj-bandung-15-agustus-2019-pada-tahun-ini-komnas-pere-tokoh/
11	9	Kang Wawa, Montir Motor Roda Tiga [bandung, 29 juli 2019] sejak tahun 2009, wawa gunawar Tokoh			29 July, 2019	https://www.newsdifabel.com/2019/07/29/kang-wawa-montir-motor-roda-tiga-bandung-29-juli-2019-sejak-tahun-2009-wawa-gunawar-tokoh/
12	10	Bapak Braille Indonesia seorang tunanetra dan huruf braille ibarat siang dan mal Tokoh			4 December, 2018	https://www.newsdifabel.com/2018/12/04/bapak-braille-indonesia-seorang-tunanetra-dan-huruf-braille-ibarat-siang-dan-mal-tokoh/
13	11	Tiga Aplikasi Ajaib Untuk Difabel Netr perkembangan teknologi untuk difabel netra bagai oase Tekno			30 March, 2021	https://www.newsdifabel.com/2021/03/30/tiga-aplikasi-ajaib-untuk-difabel-netr-perkembangan-teknologi-untuk-difabel-netra-bagai-oase-tekno/
14	12	Tongkat Canggih Bricane Terintegrasi di tengah suasana pandemi, syamsi dhuha foundation (s Tekno			22 February, 2021	https://www.newsdifabel.com/2021/02/22/tongkat-canggih-bricane-terintegrasi-di-tengah-suasana-pandemi-syamsi-dhuha-foundation-s-tekno/
15	13	Kepoin Screen Reader NVDA (Bagian l di artikel sebelumnya kita sudah berkenalan dengan jaw Tekno			21 February, 2021	https://www.newsdifabel.com/2021/02/21/kepoins-screen-reader-nvda-bagian-l-di-artikel-sebelumnya-kita-sudah-berkenalan-dengan-jaw-tekno/
16	14	Kepoin Screen Reader JAWS (Bagian l screen reader. apa, sih? jika diterjemahkan ke dalam ba Tekno			15 February, 2021	https://www.newsdifabel.com/2021/02/15/kepoins-screen-reader-jaws-bagian-l-screen-reader-apa-sih-jika-diterjemahkan-ke-dalam-ba-tekno/
17	15	Banting Tulangnya Difabel Netra Sebe di daerah perkotaan, pemandangan seorang difabel net Tekno			8 February, 2021	https://www.newsdifabel.com/2021/02/08/banting-tulangnya-difabel-netra-sebe-di-daerah-perkotaan-pemandangan-seorang-difabel-net-tekno/
18	16	Kiat Membeli Ponsel untuk Difabel N siapa yang tak mempunyai telepon pintar di zaman seka Tekno			28 January, 2021	https://www.newsdifabel.com/2021/01/28/kiat-membeli-ponsel-untuk-difabel-n-siapa-yang-tak-mempunyai-telepon-pintar-di-zaman-seka-tekno/
19	17	Peran Penting Dompot Digital Bagi Di di zaman serba canggih ini, sepertinya semua elemen se Tekno			15 April, 2021	https://www.newsdifabel.com/2021/04/15/peran-penting-dompot-digital-bagi-di-di-zaman-serba-canggih-ini-sepertinya-semua-elemen-se-tekno/
20	18	Kepoin Screen Reader JAWS (Bagian l screen reader. apa, sih? jika diterjemahkan ke dalam ba Tekno			15 February, 2021	https://www.newsdifabel.com/2021/02/15/kepoins-screen-reader-jaws-bagian-l-screen-reader-apa-sih-jika-diterjemahkan-ke-dalam-ba-tekno/
21	19	Banting Tulangnya Difabel Netra Sebe di daerah perkotaan, pemandangan seorang difabel net Tekno			8 February, 2021	https://www.newsdifabel.com/2021/02/08/banting-tulangnya-difabel-netra-sebe-di-daerah-perkotaan-pemandangan-seorang-difabel-net-tekno/
22	20	Kiat Membeli Ponsel untuk Difabel N siapa yang tak mempunyai telepon pintar di zaman seka Tekno			28 January, 2021	https://www.newsdifabel.com/2021/01/28/kiat-membeli-ponsel-untuk-difabel-n-siapa-yang-tak-mempunyai-telepon-pintar-di-zaman-seka-tekno/
23	21	Peran Penting Dompot Digital Bagi Di di zaman serba canggih ini, sepertinya semua elemen se Tekno			15 April, 2021	https://www.newsdifabel.com/2021/04/15/peran-penting-dompot-digital-bagi-di-di-zaman-serba-canggih-ini-sepertinya-semua-elemen-se-tekno/

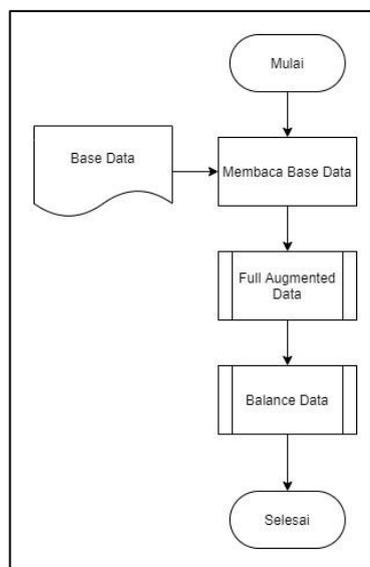
Gambar 3.3 Hasil *Scraping*

3.4 Data Preprocessing

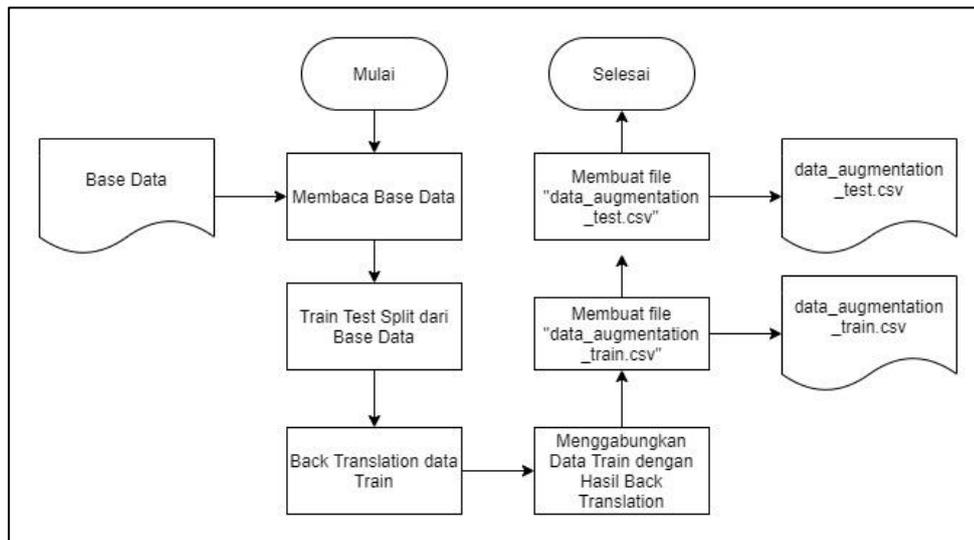
Data yang telah dikumpulkan akan diproses terlebih dahulu sebelum digunakan untuk *training* model klasifikasi. Pertama data harus “bersih” dengan cara menghapus isi konten berita yang merupakan angka, bukan alfabet, tanda baca, dan judul atau isi berita yang kosong. Kemudian mengubah seluruh huruf menjadi

lowercase. Setelah data bersih, maka data akan ditentukan kategorinya oleh pakar yang telah dipilih, setelah semua data mempunyai label kategori maka data akan dilakukan proses data augmentasi yaitu *back-translation*. Hingga data di-*stemming* untuk mengembalikan kata menjadi bentuk dasarnya seperti kerinduan menjadi rindu, proses *stemming* untuk Bahasa Indonesia akan menggunakan *library* Sastrawi yang hanya akan digunakan oleh model dari Logistic Regression. Selanjutnya data akan masuk proses *word embedding* menggunakan FastText dan model Logistic Regression.

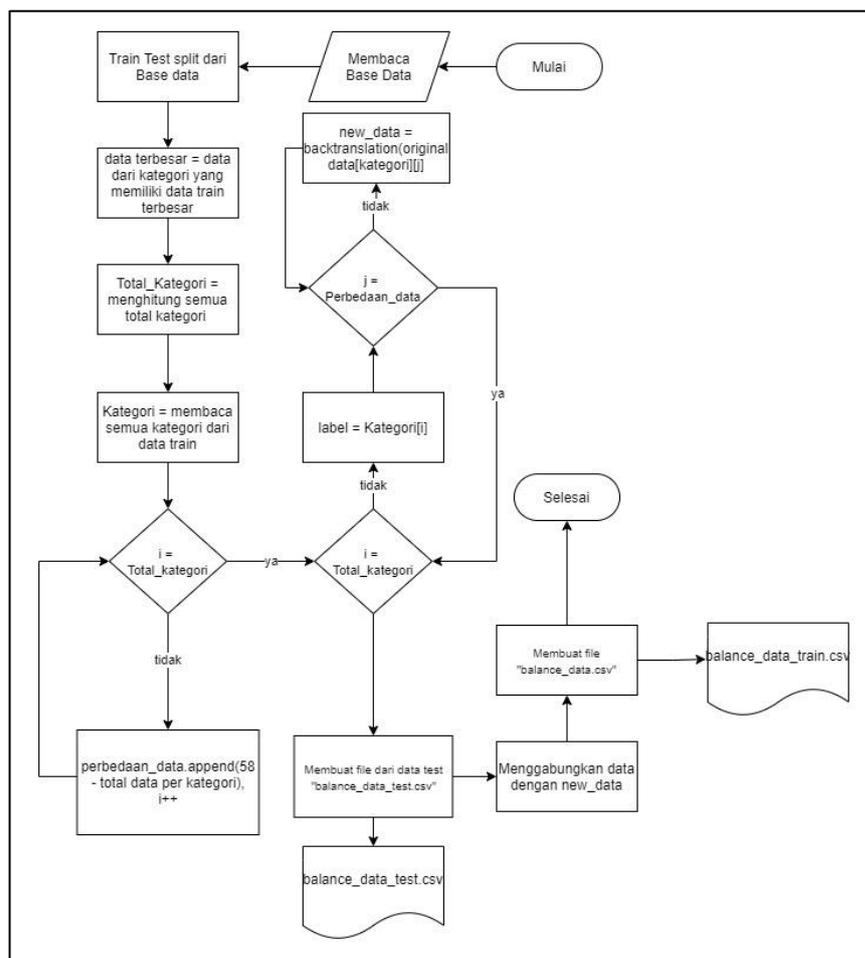
Pada penelitian ini akan menggunakan tiga skenario dataset yaitu *base data* yaitu data original dari proses *scraping*, data *augmentation* merupakan data yang dibuat dari proses *augmentation* semua data *trainnya*, dan *balance* adalah data hasil teknik *random oversampling* yang mengambil kategori dengan data paling dikit untuk di-*augmentation* hingga memiliki jumlah data yang sama dengan kategori dengan data terbanyak dst. Proses dari data *preprocessing* dapat dilihat pada Gambar 3.4 pembuatan *full augmented data* dapat dilihat pada Gambar 3.5, sedangkan *balance data* pada Gambar 3.6.



Gambar 3.4 Proses Data *Preprocessing*



Gambar 3.5 Flowchart dari Proses Full augmented data



Gambar 3.6 Flowchart dari Proses Balance Data

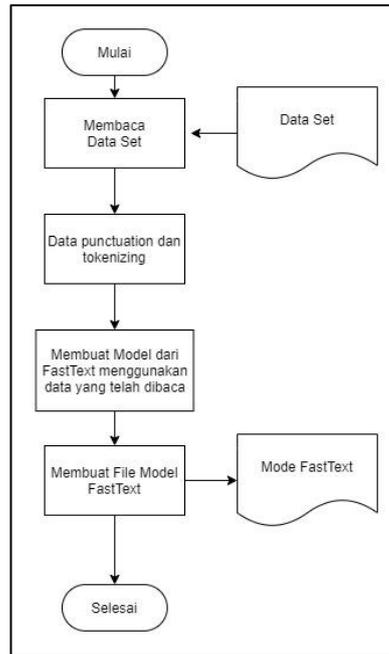
Setiap tipe jenis data *augmentation* dan data *balance* yang dibuat, akan ada enam data dari setiap jenisnya karena dalam penelitian ini menggunakan skenario rasio pembagian data *train* dan *test* 80:20, 75:25, dan 70:30. Jumlah enam file yang akan dibuat perjenisnya terdiri dari file *train* dan *test* dari setiap rasio pembagian datanya.

3.5 Pembuatan Model Pembelajaran Mesin

Pembuatan model pembelajaran mesin merupakan tahap untuk mengimplementasi dan mengevaluasi Metode yang digunakan. Mulai dari *training* model FastText dan Logistic Regression dengan data yang telah melalui tahap *preprocessing* dengan penjabaran *flowchart* sebagai berikut.

3.5.1 Flowchart FastText

Penelitian ini menggunakan beberapa skenario data yang telah diaugmentasi maupun tidak sehingga ada sembilan model FastText yang dibuat untuk setiap data yang digunakan. Walaupun menggunakan data yang berbeda-beda, FastText tetap memakai cara yang sama untuk mempelajari sebuah data. Proses membuat model FastText dapat dilihat pada Gambar 3.7.



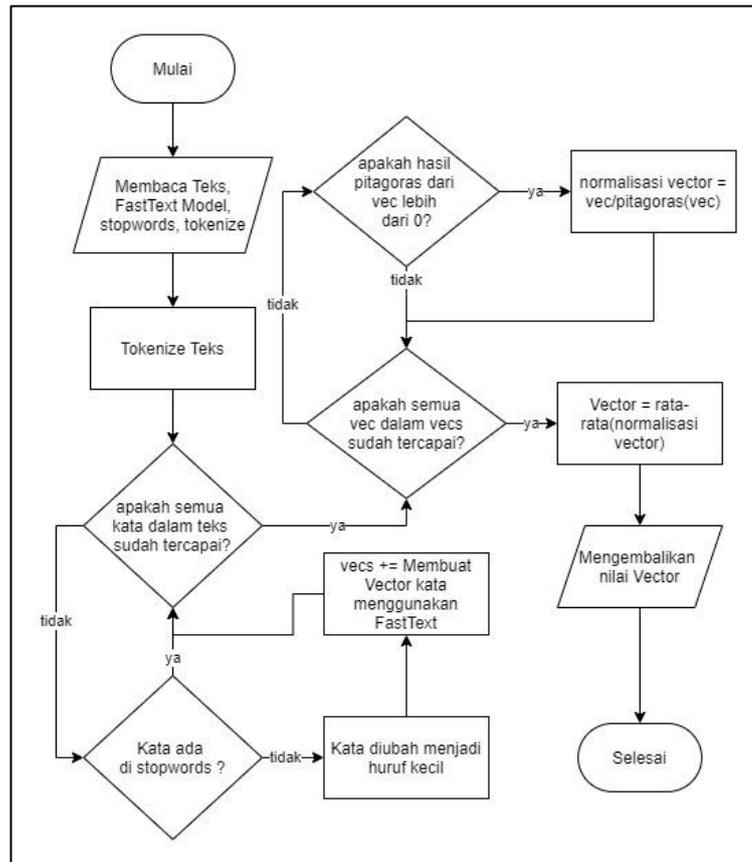
Gambar 3.7 *Flowchart* Proses Pembuatan Model FastText

Proses yang terjadi adalah membaca *dataset* yang akan dipelajari, lalu melakukan *text processing* yaitu menghilangkan tanda baca dan tokenizing. Pada data yang akan menjadi bahan untuk pembelajaran model FastText sengaja tidak di-*stemming* karena FastText mempunyai fitur dapat membuat sebuah *vector* dari sebuah kata-kata yang tidak ada pernah dipelajari dari *subword* dari kata yang pernah dipelajari, begitu juga dengan *stop words* karena semakin banyak kata-kata yang dipelajari oleh FastText semakin baik juga modelnya. Pada bagian deklarasi model, parameter, serta proses FastText mempelajari sebuah kata dalam dilakukan dalam satu proses. Setelah FastText selesai mempelajari data maka akan membuat *file* dari model FastText yang telah dibuat dengan penamaan *file* sesuai dengan nama data yang dipelajari.

3.5.2 Flowchart Normalisasi Sentence Vector

Pada proses pelatihan hingga klasifikasi menggunakan metode Logistic Regression akan dimulai dari membaca data yang akan digunakan, *stemming* data,

membaca model FastText yang akan mengubah data yang semula berbentuk *string* menjadi sebuah *vector*. Proses mengubah data menjadi *vector* diinisialisasi dengan nama normalisasi *sentence vector*, prosesnya dapat dilihat pada Gambar 3.8.



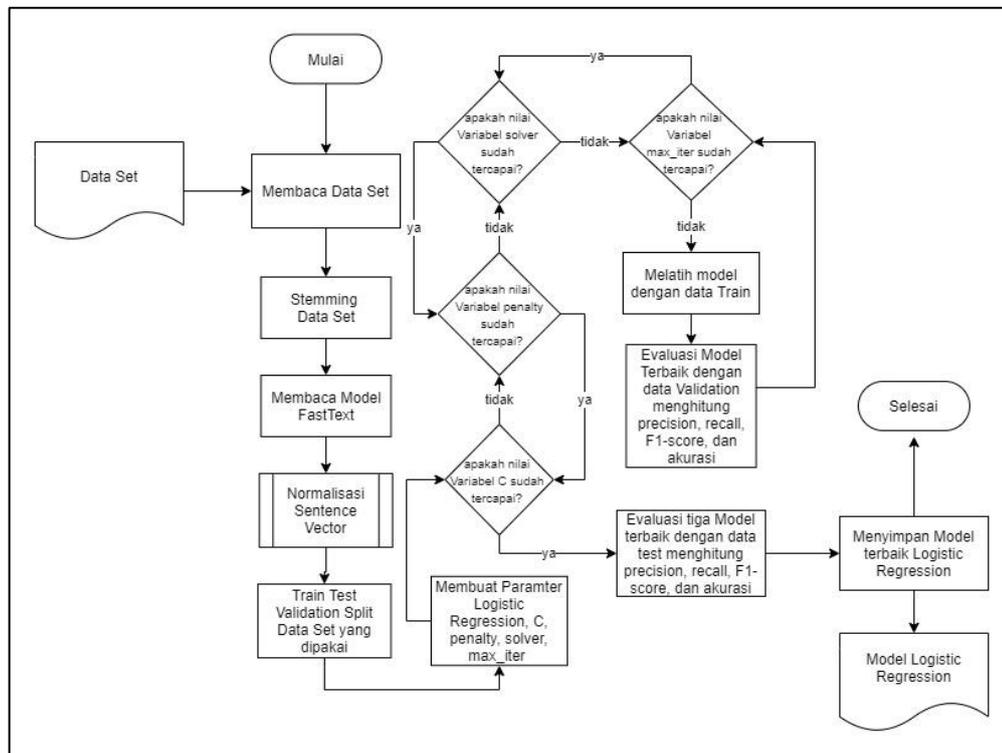
Gambar 3.8 Proses Normalisasi Sentence Vector

Proses yang dilakukan dalam fungsi normalisasi *sentence vector* adalah *Tokenize* dari setiap teks menjadi kata, lalu melakukan proses pengulangan dari setiap kata. Jika kata yang akan diproses termasuk di dalam *list* dari *stop words* maka akan dilewati, sebaliknya jika tidak ada di *list* maka kata kan diubah menjadi huruf kecil, lalu akan menggunakan FastText untuk mengubah kata tersebut menjadi *vector*. Setelah semua kata sudah diubah menjadi *vector*, maka *vector* dari sebuah teks akan dinormalisasi dengan cara *vector* dari teks di bagi dari hasil pitagorasnya, setelah berhasil dinormalisasi akan dihitung nilai rata-ratanya, proses

ini akan menghasilkan hasil *vector* dari data yang akan digunakan untuk diproses dengan metode Logistic Regression.

3.5.3 Flowchart Logistic Regression

Pada penelitian ini terdapat proses klasifikasi dengan Logistic Regression yang menggunakan proses *hyperparameter tuning* untuk mendapatkan paramter terbaik dari model Logistic Regression untuk *dataset* yang digunakan. Proses dari *hypterparamter tuning* dilakukan secara manual, yang dapat dilihat pada Gambar 3.9.



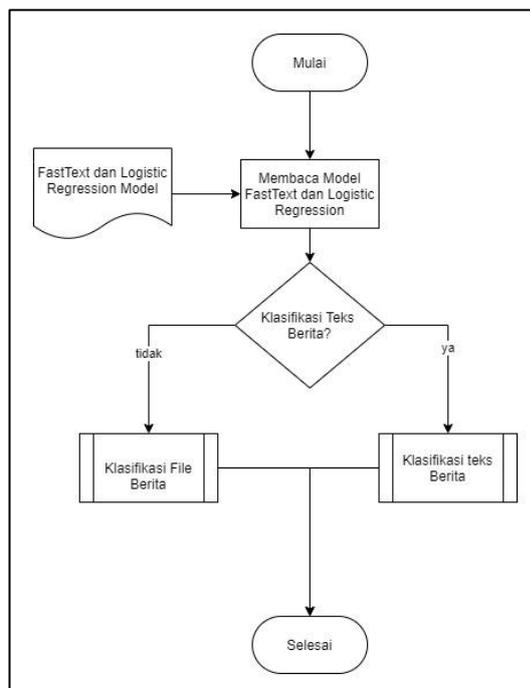
Gambar 3.9 *Flowchart* dari Proses Pembuatan Model Logistic Regression

Pada proses pembuatan model Logistic Regression proses *hyperparameter tuning* dengan cara manual, sehingga prosesnya akan dimulai dari membaca data, *Stemming* data, membaca model FastText, menjalankan fungsi normalisasi *sentence vector* untuk mengubah data menjadi vector, inialisasi paramter dari

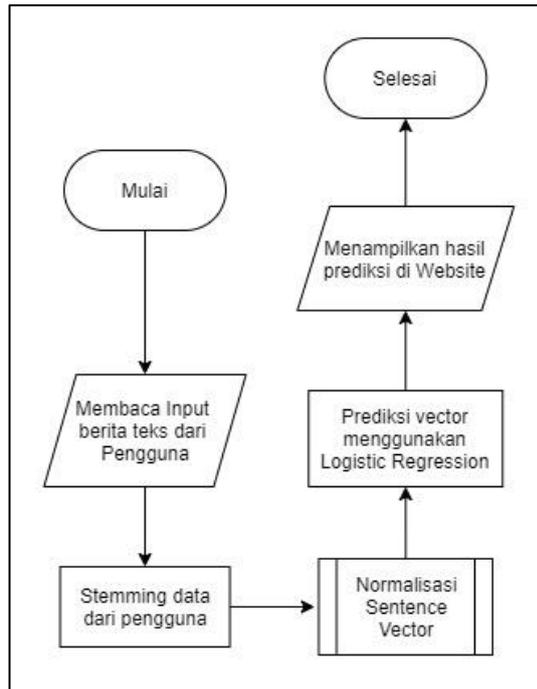
Logistic Regression, lalu melakukan proses pengulangan sesuai dengan jumlah dari paramter yang telah ditentukan, hingga proses pengulangan selesai dan mendapatkan hasil *precision*, recall, F1-score, dan *Accuracy* yang terbaik dan menyimpan model terbaik.

3.6 Perancangan Aplikasi

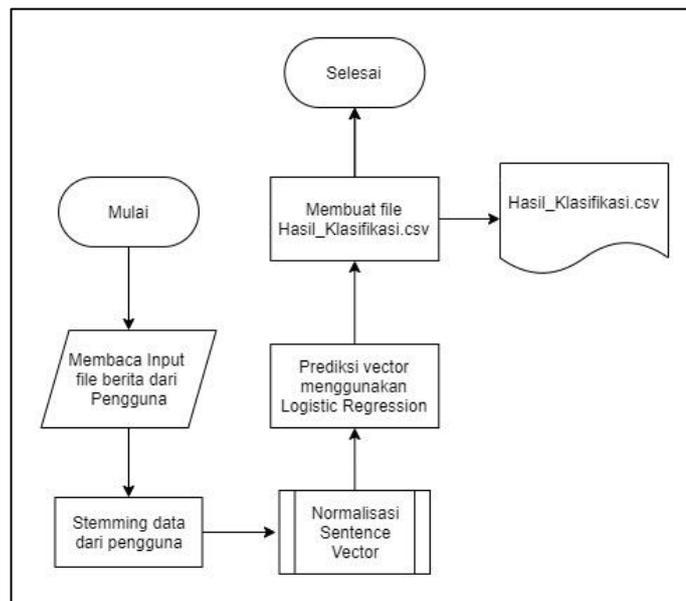
Proses perancangan aplikasi adalah proses menentukan *workflow* dari aplikasi yang akan dibuat mulai dari *website* yang akan terintegrasi dengan model pembelajaran mesin yang telah dibuat, hingga tampilan tatap muka sebuah *website*. Pemrograman *website* menggunakan bahasa pemrograman *python* dan *framework* Flask dan mempunyai dua fitur yaitu klasifikasi berita teks dan *file* yang berisikan kumpulan berita. sehingga mempunyai alur utama *website* yang dapat dilihat pada Gambar 3.10, alur klasifikasi teks berita yang dapat dilihat pada Gambar 3.11, dan klasifikasi file pada Gambar 3.12.



Gambar 3.10 Alur Utama *Website*



Gambar 3.11 Proses Klasifikasi Teks Berita Pada *Website*

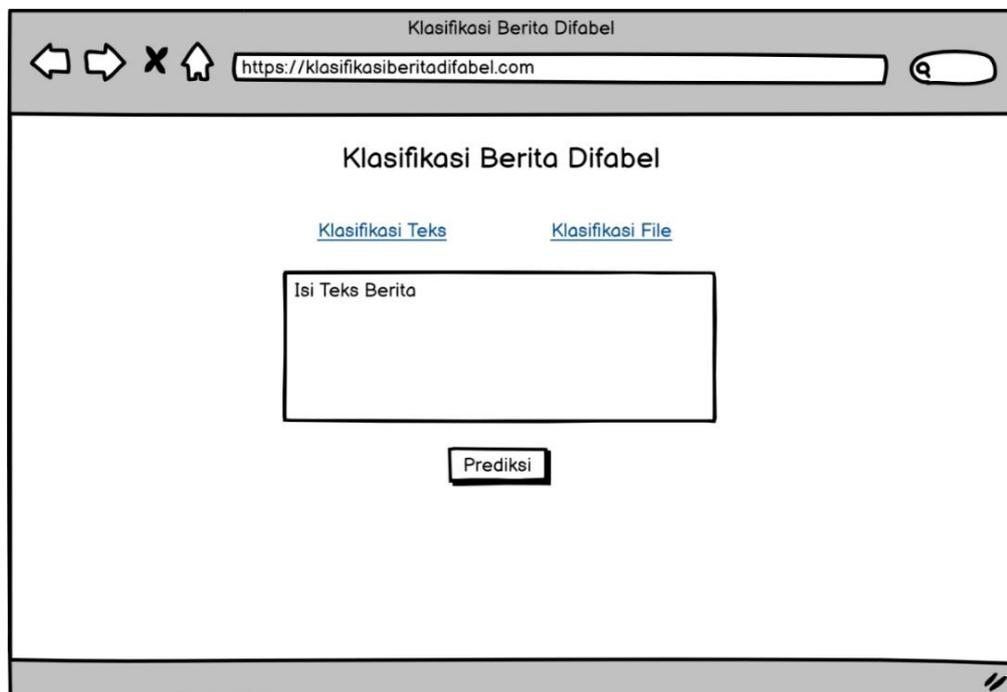


Gambar 3.12 Proses Klasifikasi File Berita Pada *Website*

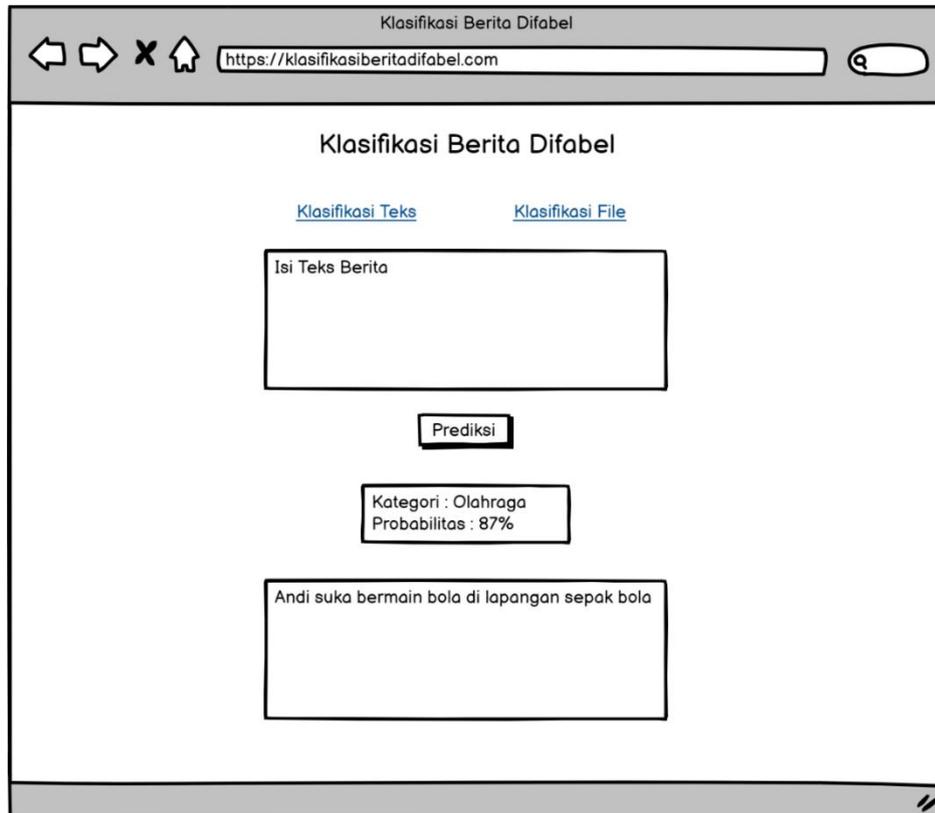
Proses Klasifikasi dari teks berita seperti pada Gambar 3.11, pertama akan membaca semua model FastText dan Logistic Regression yang telah ditentukan, membaca input dari pengguna, lalu akan *distemming*, data akan menggunakan diproses difungsi normalisasi *sentence vector* agar *sentence* berubah menjadi sebuah *vector* sehingga dapat diprediksi oleh model Logistic Regression lalu

ditampilkan ke halaman *website*. Pada proses klasifikasi *file* mempunyai alur yang sama, hanya saja hasil klasifikasi akan berupa file csv (Comma-separated values) bernama “Hasil_Klasifikasi”.

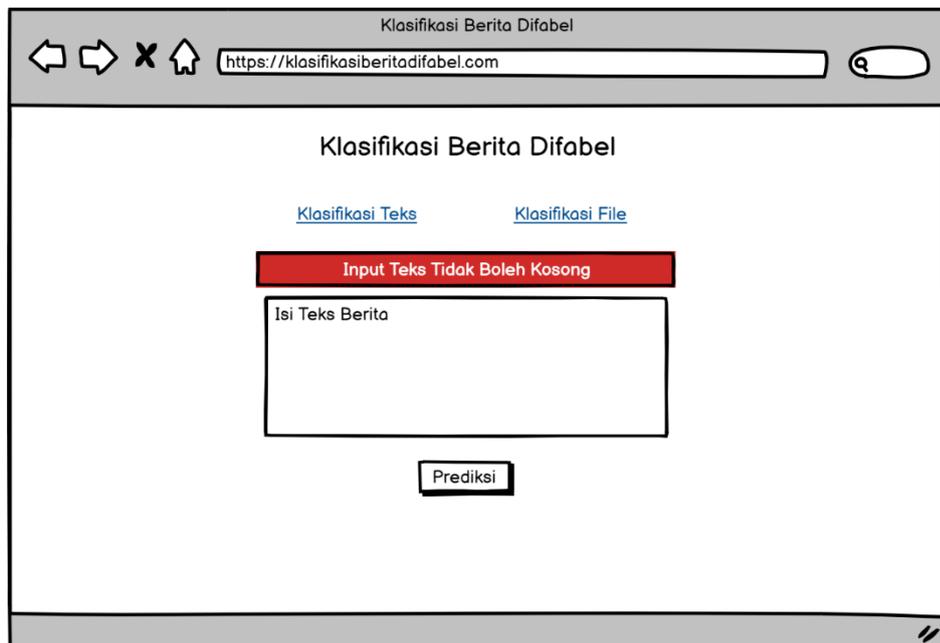
Tampilan *website* yang akan digunakan menggunakan HTML dan CSS (*Cascading Style Sheets*). *Website* yang digunakan dapat menerima *input* pengguna berupa teks dari berita yang akan diklasifikasi, setelah mengisi memasukkan teks berita maka pengguna dapat menekan tombol bertuliskan “prediksi” untuk melakukan proses klasifikasi dengan input teks. Perancangan Tampilan dapat dilihat pada Gambar 3.13 untuk Tampilan awal, Gambar 3.14 untuk tampilan setelah prediksi, dan Gambar 3.15 untuk tampilan jika pengguna tidak memasukkan apapun tetapi menekan tombol prediksi.



Gambar 3.13 Tampilan Sebelum Klasifikasi Teks Berita

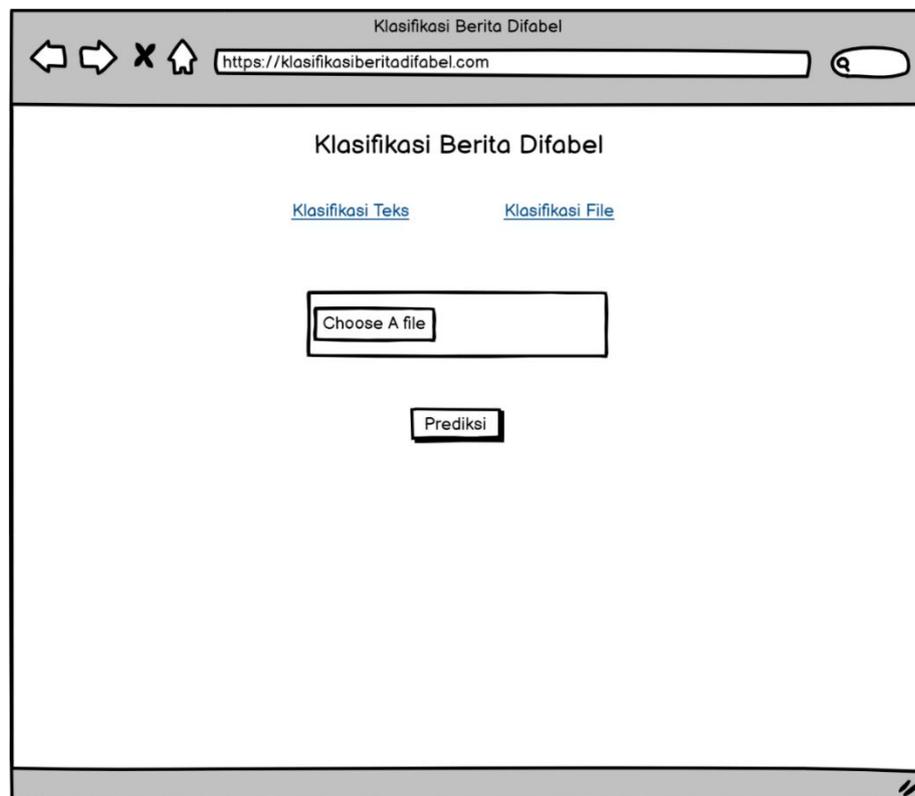


Gambar 3.14 Tampilan *Website* Setelah Mengklasifikasi Teks Berita

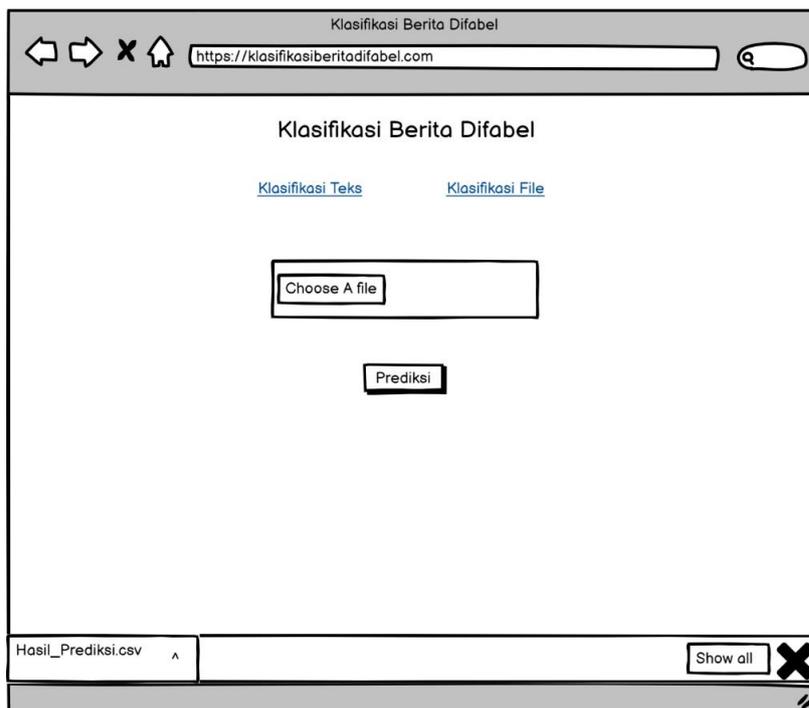


Gambar 3.15 Tampilan *Website* Jika Pengguna Tidak Memasukkan Teks Tetapi Menekan Tombol Prediksi

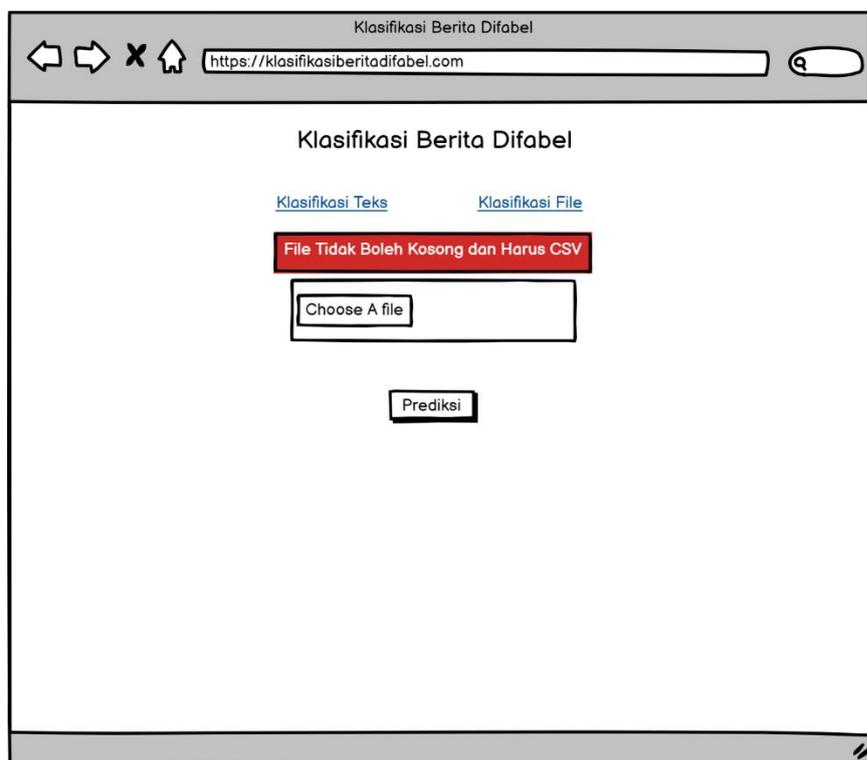
Disaat pengguna memasukkan teks yang akan diklasifikasi lalu menekan tombol prediksi, maka akan muncul hasil prediksi kategori dan probabilitas dari teks yang dimasukan dan akan dimunculkan kembali teks yang diklasifikasi, tetapi jika hanya menekan tombol prediksi tetapi tidak memasukkan teks apapun maka akan muncul pesan seperti pada Gambar 3.15. *Website* juga dapat menerima input berupa file, sehingga dapat mengklasifikasi banyak berita secara sekaligus, dan hasil dari klasifikasi dapat dilihat pengguna dari file yang akan otomatis diunduh dalam bentuk csv. Tampilan dari halaman awal dapat dilihat pada Gambar 3.16, tampilan halaman hasil klasifikasi dapat dilihat pada Gambar 3.17, dan tampilan dari halaman ketika pengguna tidak memasukkan file dapat dilihat pada Gambar 3.18.



Gambar 3.16 Tampilan Halaman *Website* untuk Mengklasifikasi *File* Berita



Gambar 3.17 Tampilan dari Halaman *Website* untuk Hasil Mengklasifikasi *File* Berita



Gambar 3.18 Tampilan Halaman Klasifikasi *File* Jika Tidak Ada Input

3.7 Pembuatan Aplikasi

Proses ini bertujuan untuk membuat aplikasi berbasis *website* yang telah terintegrasi dengan model yang telah buat. Aplikasi *website* akan dibuat menggunakan *framework* Flask yang diprogram dengan bahasa pemrograman Python. *Website* yang dibuat dapat menerima input berupa Judul dan Isi berita lalu akan memunculkan hasil prediksi dari model yang telah dibuat.

3.8 Penulisan Laporan dan Konsultasi

Membuat laporan dari penelitian yang berisikan dokumentasi penelitian mulai dari latar belakang hingga kesimpulan dan saran. Konsultasi berguna untuk dapat menulis laporan dengan baik dan terarah.