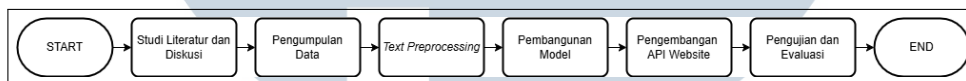


## BAB 3 METODOLOGI PENELITIAN

### 3.1 Metode Penelitian

Metode penelitian yang digunakan pada penelitian ini adalah metode kuantitatif yang merupakan analisis sistematis terhadap suatu permasalahan dengan mengukur data yang sudah dikumpulkan menggunakan berbagai teknik seperti teknik statistik, matematis, maupun komputasi [25]. Pelaksanaan penelitian dengan metode kuantitatif dilakukan dengan tahapan penelitian yang terstruktur dan sistematis sesuai dengan tahapan penelitian ilmiah [26]

### 3.2 Tahapan Penelitian



Gambar 3.1. *Flowchart* Metode Penelitian

Gambar 3.1 menunjukkan alur penelitian yang dimulai dari studi literatur dan diskusi, pengumpulan data, *text preprocessing*, pembangunan model, pengembangan *API Website*, dan pengujian serta evaluasi. Penjelasan tiap tahapan penelitian adalah sebagai berikut.

#### 3.2.1 Telaah Literatur dan Diskusi

Pada tahap ini, dilakukan telaah literatur berupa riset secara teoritis berdasarkan berbagai sumber seperti jurnal ilmiah untuk mencari algoritma yang sesuai untuk kasus pendeteksian peluluhan kata. Selain itu, dilakukan juga diskusi bersama dosen pembimbing untuk mencari *library* yang dapat digunakan untuk mempermudah pembangunan model. Kedua proses ini dilakukan untuk mencari dan memilih algoritma serta pendekatan lainnya yang paling efektif dalam pembangunan model untuk mendeteksi kesalahan peluluhan kata.

#### 3.2.2 Pengumpulan Data

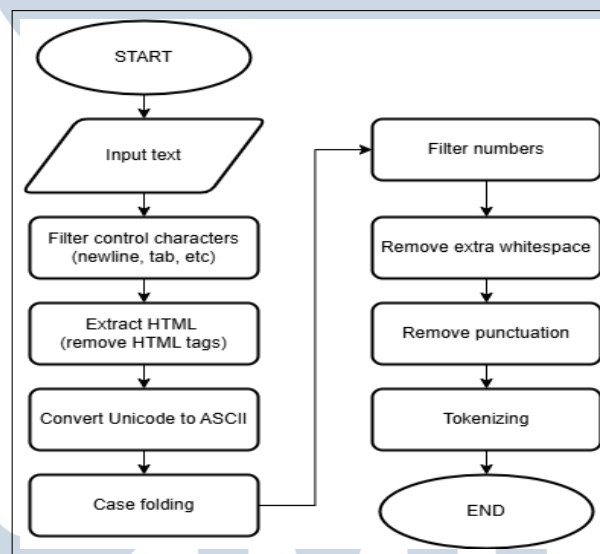
Tahap pengumpulan data ini dilakukan dengan mengumpulkan seluruh bentuk peluluhan kata yang benar melalui *crawling* terhadap KBBI, baik kata

berkonsonan ganda maupun tunggal. Adapun kata yang dikumpulkan harus berawalan dengan *prefix me-* dan *pe-*. Data tersebut kemudian akan disimpan dalam ekstensi *.xlsx* untuk digunakan pada model.

Selain itu, dikumpulkan juga data berupa artikel berita dari *TribunNews* sebagai dataset pengujian sistem. Dataset dengan ekstensi *.pdf* dan *.docx* ini dikumpulkan oleh mahasiswa jurusan Jurnalistik dan disimpan dalam folder *Google Drive* yang dapat diakses oleh tim peneliti U-Tapis. Pada penelitian ini, sebanyak 450 artikel berita digunakan sebagai dataset pengujian sistem untuk menguji kemampuan deteksi dan koreksi sistem.

### 3.2.3 Text Preprocessing

Kemudian, dilakukan *preprocessing* pada data *input* yang berasal dari teks berita *Tribun News*. Tahapan ini dapat dilihat melalui Gambar 3.2 berikut.



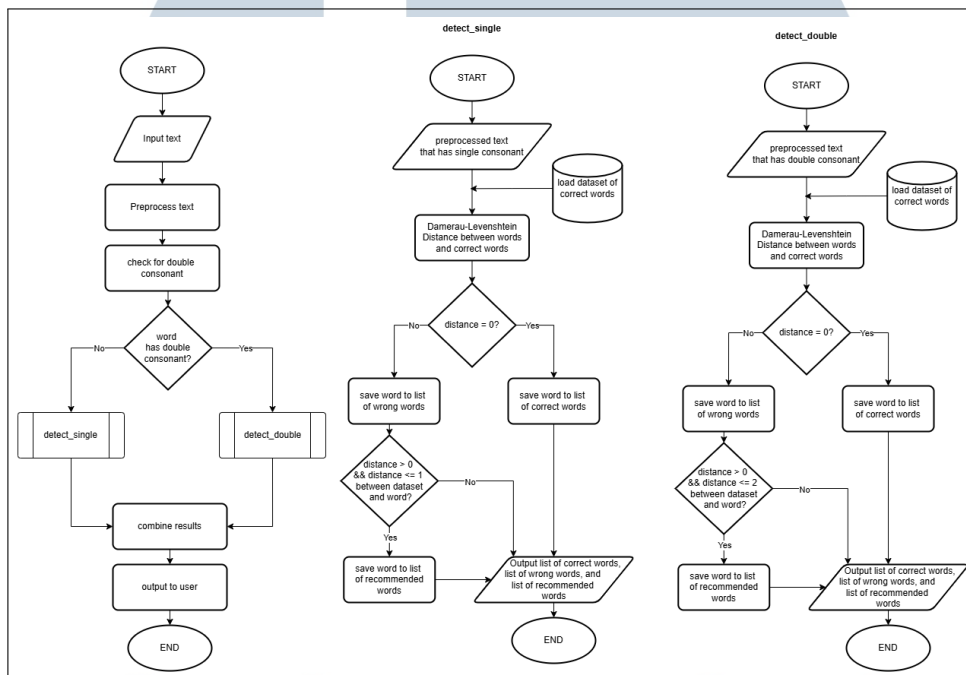
Gambar 3.2. Flowchart Text Preprocessing

Pada tahap ini, *input* berita akan melalui pra-proses mulai dari penghapusan karakter kontrol, penghapusan *tag HTML*, konversi karakter *Unicode* menjadi ASCII, *case folding*, penghapusan numerik, penghapusan spasi berlebih, penghapusan tanda baca, dan *tokenizing*.

### 3.2.4 Pembangunan Model

Pada tahap pembangunan model, digunakan algoritma *Damerau-Levenshtein Distance (DLD)* untuk mendeteksi kesalahan penulisan peluluhan

kata dalam teks berita yang telah di-*preprocess* sebelumnya. Algoritma tersebut kemudian akan menghitung jarak antara dua *string* berdasarkan empat operasi dasar: penghapusan (*deletion*), penyisipan (*insertion*), substitusi (*substitution*), dan *transposition*, yang pada konteks ini, algoritma akan membandingkan kata-kata yang telah di-*preprocess* dengan kumpulan kata-kata yang benar berdasarkan *dataset*.



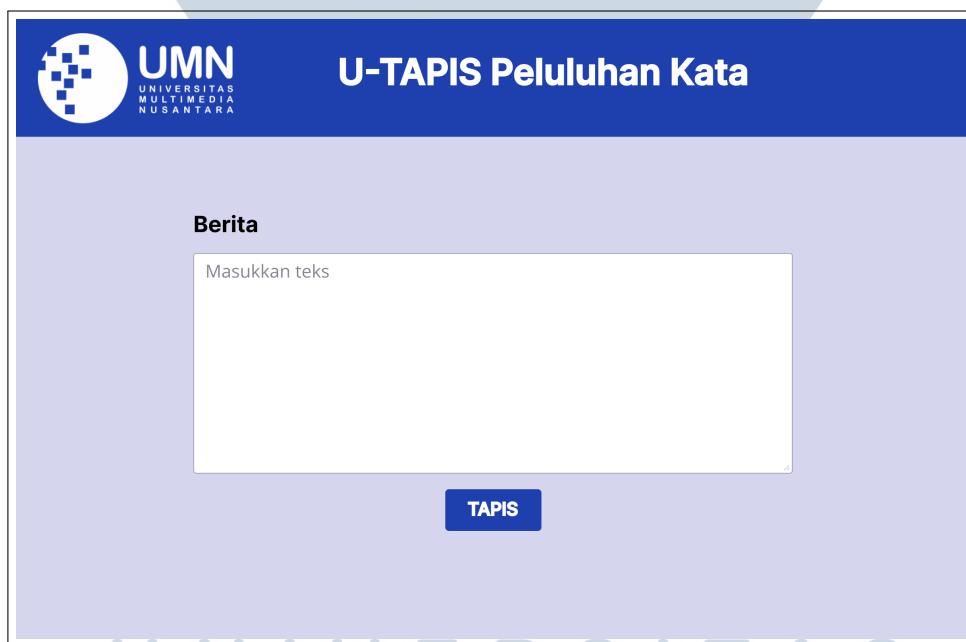
Gambar 3.3. Flowchart Pembangunan Model

Gambar 3.3 menunjukkan alur pembangunan model secara keseluruhan, yang dimulai dari *input* teks berita. Setelah melalui tahap pra-proses, teks kemudian akan dicek apakah memiliki konsonan ganda atau tidak. Jika memiliki konsonan ganda, maka berita akan diproses dengan fungsi *detect\_double* dan jika tidak memiliki konsonan ganda, maka berita diproses menggunakan fungsi *detect\_single*. Pada fungsi *detect\_single* dan *detect\_double*, pertama-tama akan dibandingkan jarak *Damerau-Levenshtein Distance* antara kata-kata pada berita dengan kumpulan kata benar pada dataset Excel. Jika jarak DLD-nya adalah 0, maka kata akan disimpan ke daftar kata benar, tetapi jika jarak DLD tidak sama dengan 0, maka kata tersebut akan disimpan ke daftar kata salah. Perbedaan antara fungsi *detect\_single* dengan fungsi *detect\_double* adalah pada perbandingan jarak sebagai dasar penilaiannya. Pada fungsi *detect\_single*, jarak lebih besar dari 0 dan lebih kecil sama dengan 1 digunakan sebagai aturan untuk memasukkan kata dari dataset ke daftar kata rekomendasi untuk perbaikan. Sedangkan pada fungsi *detect\_double*, jarak yang

digunakan sebagai aturan adalah jika jarak DLD lebih besar dari 0 dan lebih kecil sama dengan 2. Hasil kata yang diperoleh dari kedua fungsi ini kemudian dikembalikan sebagai *list* kata benar, *list* kata salah, dan *list* kata rekomendasi.

### 3.2.5 Pengembangan API Website

Setelah model selesai dibangun dan memiliki akurasi yang baik, maka dikembangkan *website* U-Tapis yang telah ada sebelumnya dengan menambahkan fitur baru khusus untuk implementasi model ini. Untuk mengimplementasikan model, maka dibuat API yang diprogram menggunakan bahasa pemrograman *Python* dan *framework Flask* yang akan mengembalikan hasil model berdasarkan *input* pengguna dalam format JSON. Bagian antarmuka pengguna akan menampilkan kesalahan penulisan yang telah dideteksi beserta perbaikannya maupun kata luluh yang sudah benar yang dapat dilihat melalui *prototype* berikut.



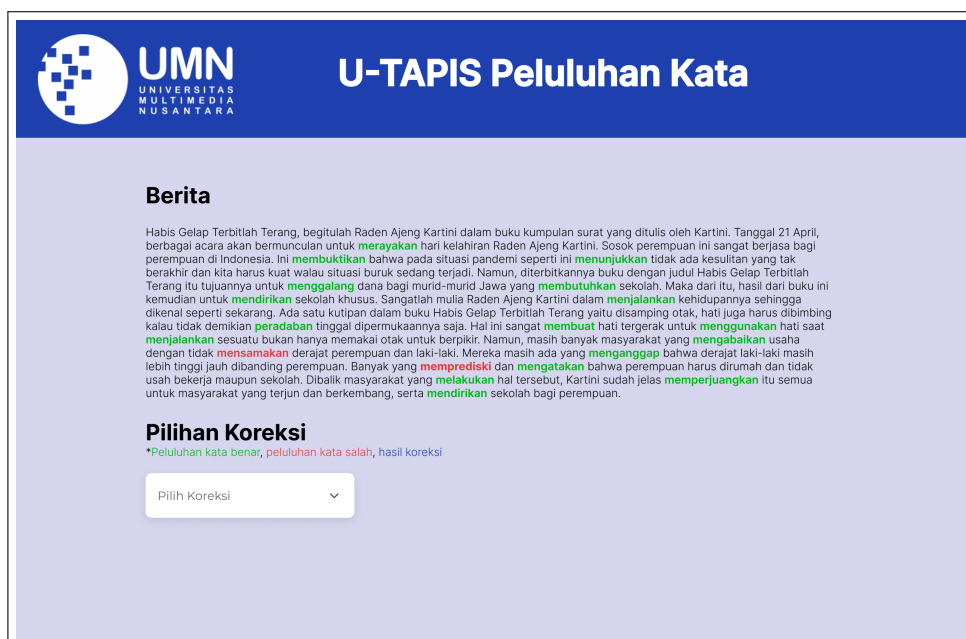
Gambar 3.4. Tampilan Awal Web

Gambar 3.4 menunjukkan tampilan awal program. Pada halaman ini, terdapat sebuah *text area* dan sebuah tombol bernama “Tapis”. Pengguna dapat memasukkan teks berita sebagai *input* dan menekan tombol “Tapis” untuk mengecek peluluhan kata yang terdapat pada teks seperti pada Gambar 3.5 berikut.



Gambar 3.5. Tampilan Isi Berita

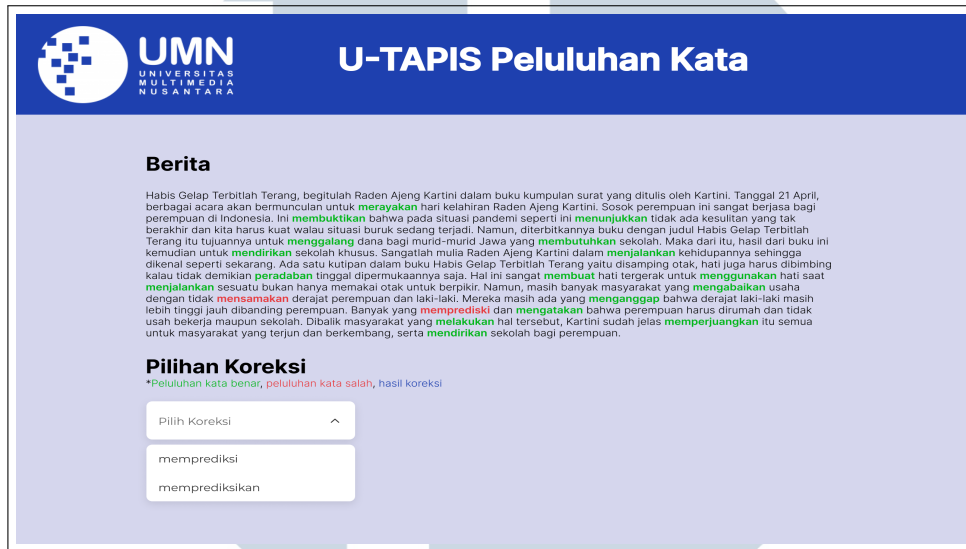
Setelah pengguna menekan tombol “Tapis”, maka sistem akan memproses berita tersebut dan menampilkan kembali teks berita beserta hasil deteksinya seperti pada Gambar 3.6 berikut.



Gambar 3.6. Tampilan Hasil Deteksi

Warna hijau menunjukkan bahwa kata terdeteksi sudah memiliki bentuk luluh yang benar sesuai KBBI dan warna merah menunjukkan bahwa kata terdeteksi

tersebut memiliki bentuk luluh yang salah. Jika kata berwarna merah yang berarti memiliki bentuk luluh yang salah ditekan, maka pengguna dapat menekan pilihan atau *dropdown* untuk melihat daftar perbaikan kata tersebut seperti pada Gambar 3.7 berikut.



Gambar 3.7. Tampilan Pilihan Perbaikan

Pengguna dapat memilih salah satu dari opsi perbaikan yang diberikan sistem. Opsi ini dapat terdiri dari satu atau lebih pilihan tergantung kata dasarnya, seperti pada contoh kata “memprediksi” yang memiliki dua bentuk luluh yang benar, yaitu “memprediksi” dan “memprediksikan”.



Gambar 3.8. Tampilan Setelah Perbaikan

Setelah pengguna memilih salah satu opsi perbaikan, maka secara otomatis

kata yang salah akan diganti dengan opsi tersebut seperti pada Gambar 3.8. Warna biru pada teks menunjukkan bahwa kata tersebut merupakan kata yang sebelumnya memiliki bentuk luh kata yang salah dan kini digantikan dengan hasil koreksi sistem.

### 3.2.6 Pengujian dan Evaluasi

Tahap ini dilakukan menggunakan 450 berita dari *Tribun News* dengan berbagai skenario yaitu 100, 150, dan 200 berita. Ketika hasil pengujian diperoleh, maka akan ditentukan *confusion matrix*-nya yang kemudian dapat digunakan untuk perhitungan akurasi, presisi, *f1-score*, dan *recall*. Selain itu, dicari juga akurasi rekomendasi yang diberikan sistem dengan membagi total koreksi yang benar dengan total peluluhan kata yang salah.

