BAB 3

METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Metodologi yang diterapkan dalam penelitian ini terdiri dari beberapa tahap, antara lain sebagai berikut:

1. Studi literatur

   Studi literatur dilakukan dengan tujuan memperkuat pemahaman terhadap teori-teori yang digunakan serta sebagai panduan dalam melakukan penelitian, sehingga setiap yang dilakukan dalam penelitian tidak menyimpang dari teori yang ada. Studi literatur yang dilakukan yaitu terhadap pemahaman kesenian tradisional Indonesia, Named Entity Recognition, serta Conditional Random Field. Adapun sumber referensi untuk studi literatur ini diperoleh dari berbagai sumber seperti jurnal, buku, serta internet.

2. Pengumpulan data

   Data yang digunakan berasal dari artikel digital Kompas.com yang di-*preprocessing* dan diberikan label secara manual. Selanjutnya data tersebut dibagi untuk data *training* dan data *testing*. Selain itu, terdapat juga data *testing* eksternal berasal dari artikel digital Wikipedia.com yang disusun ulang dalam bentuk pdf.

3. Perancangan sistem

   Perancangan dilakukan agar aplikasi dibentuk sesuai dengan kebutuhan dan terstruktur. Tahapan ini mencakup penyusunan *flowchart*.

4. Implementasi

   Implementasi dilakukan dengan melakukan *coding* dalam menyusun sistem sesuai dengan *flowchart* yang telah dibuat sebelumnya.

5. Pengujian model dan pengukuran akurasi

   Tahap pengujian dilakukan untuk memastikan program berjalan dengan lancar dan menghasilkan akurasi yang baik. Adapun pengukuran akurasi didasarkan pada ketepatan sistem memproses data training dan mengeluarkan label yang sesuai.

6. Penyusunan laporan

   Penyusunan laporan dilakukan sebagai tahap akhir dari penelitian, dengan tujuan melaporkan tahap dan hasil dari penelitian. Laporan disusun sesuai dengan panduan yang berlaku dari program studi Informatika Universitas Multimedia Nusantara.

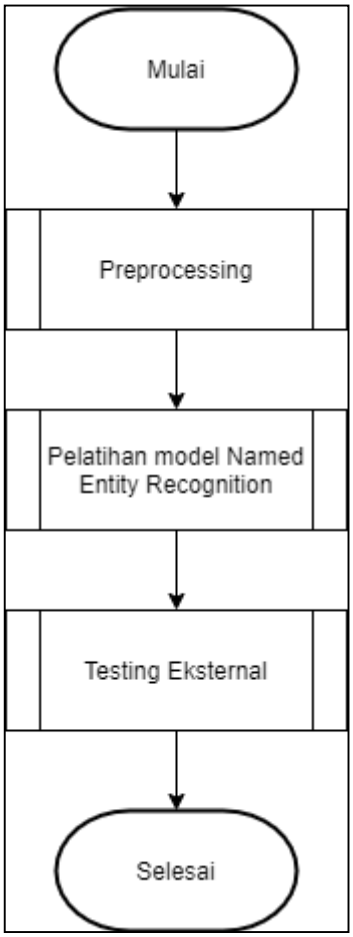
3.2 Perancangan Aplikasi

Perancangan aplikasi direpresentasikan dalam bentuk *flowchart*. Secara keseluruhan, alur aplikasi digambarkan dalam *flowchart* utama. *Flowchart* utama berisi beberapa modul yang dijelaskan kembali secara mendetail.

3.2.1 Flowchart Utama

Alur pada *flowchart* dapat dilihat pada Gambar 3.1. Dimulai dari tahapan *preprocessing* yang bertujuan untuk mengambil teks dari setiap artikel digital yang telah disusun *link*-nya menjadi per kata dalam csv yang baru. Selanjutnya *dataset* yang telah diberikan label akan masuk ke dalam proses pelatihan model *Named*

Entity Recognition. Dari proses tersebut dihasilkan model NER yang selanjutnya digunakan dalam modul *Testing.*

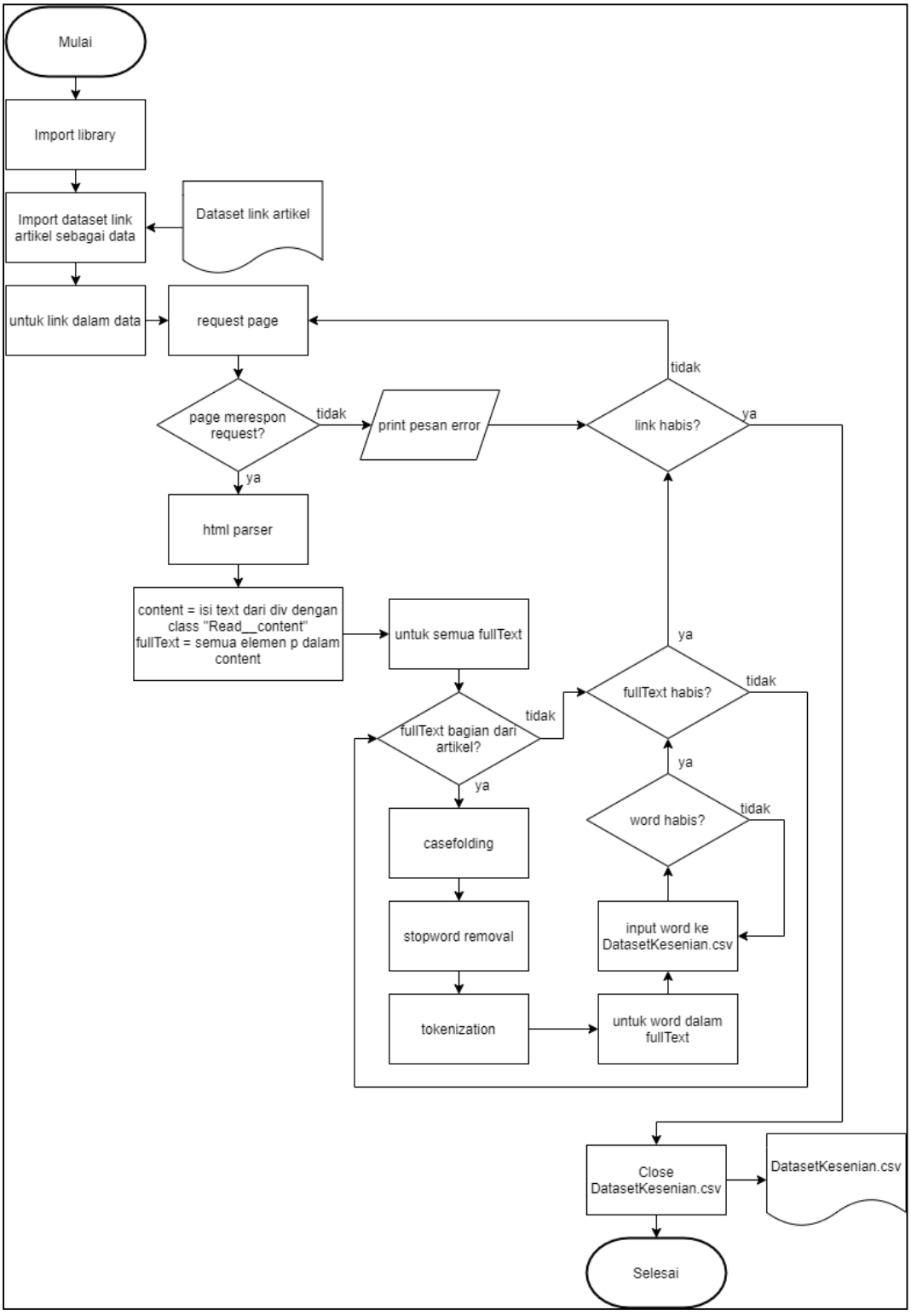


Gambar 3.1 Flowchart Utama

3.2.2 Flowchart Preprocessing

Preprocessing dilakukan untuk mengolah seluruh artikel digital yang sudah disiapkan menjadi sebuah dataset yang siap diberi label. Sebelum melakukan *preprocessing*, dilakukan pengambilan daftar *link* artikel dan melakukan *request page* terhadap setiap *link* yang ada. Pada hasil *request* yang berbentuk HTML, dicari bagian konten dari artikel tersebut. Pada Kompas.com, seluruh konten artikel dimuat pada div dengan kelas bernama "read_content". Di dalam div tersebut, konten artikel disusun per kalimat dan dibungkus dalam elemen p. Oleh karena itu, tahapan inti dari *preprocessing* dilakukan untuk setiap elemen p.

Iterasi dilakukan untuk setiap elemen p. Dilakukan pengecekan apakah elemen p tersebut memiliki isi dan isinya merupakan konten. Tahapan dari *preprocessing* berupa *casefolding* (membuat seluruh karakter menjadi huruf kecil dan menghilangkan karakter lain selain alfabet), *stopword removal* yang bertujuan untuk menghilangkan kata-kata tanpa makna sesuai dengan library Sastrawi, serta *tokenization*. Hasil dari *tokenization* yaitu setiap kalimat akan terpecah menjadi per kata. Selanjutnya setiap kata disimpan menjadi *row* baru pada DatasetKesenian.csv.

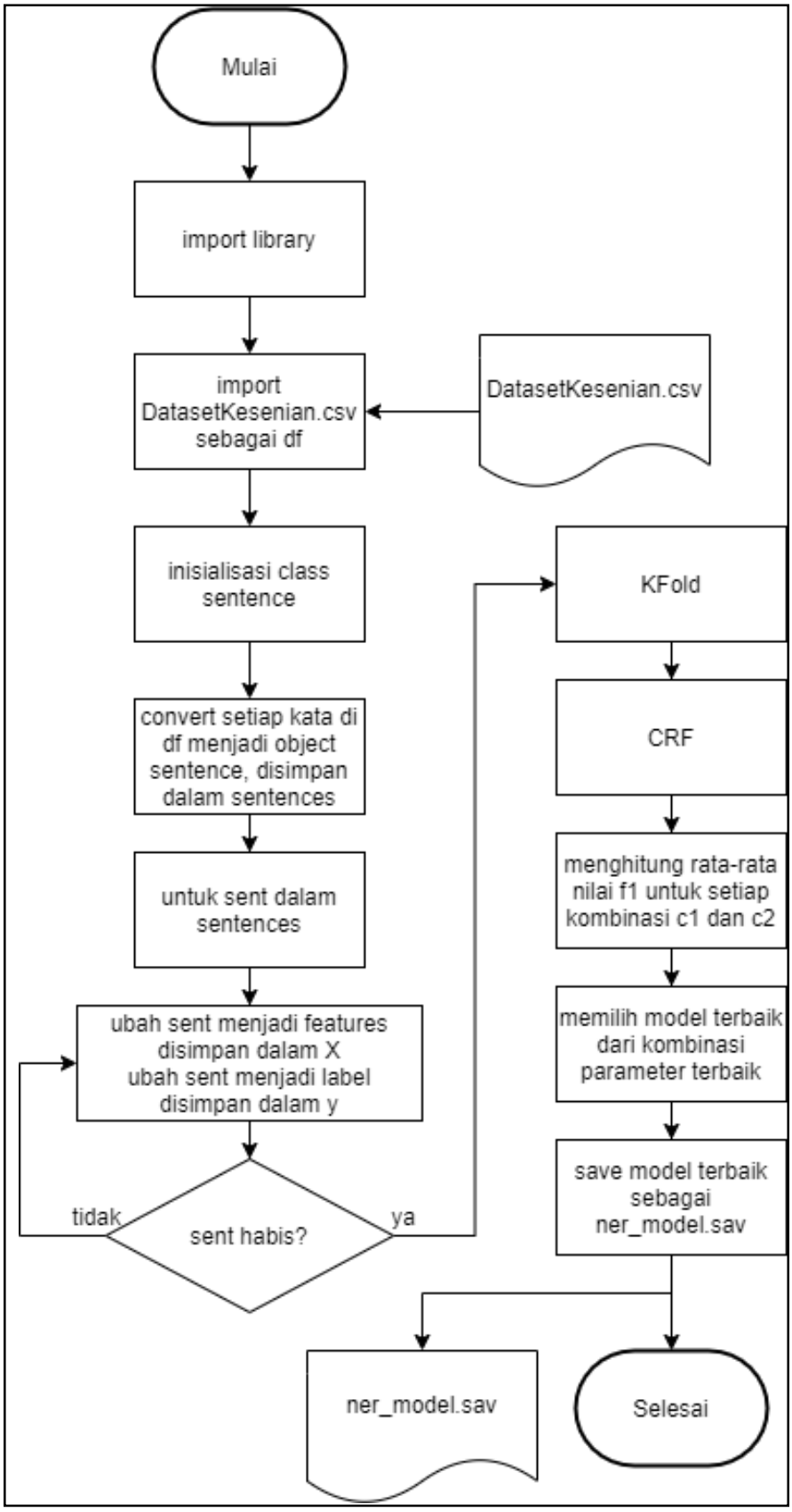


Gambar 3.2 Flowchart Preprocessing

3.2.3 Flowchart Pelatihan Model Named Entity Recognition

Tahapan ini bertujuan untuk menyusun model NER dan melakukan evaluasi guna mendapatkan hasil berupa f1-*score*. Data masukan untuk tahap ini adalah dataset yang berisi konten artikel dari Kompas.com dan telah dilabeli secara manual dalam bentuk DatasetKesenian.csv. Setiap baris dari dataset diproses menjadi objek *sentence* dan dikelompokkan berdasarkan id *sentence*. Selanjutnya, objek dikonversi menjadi *features* dan *labels* menggunakan *default function* dari *library* sklearn-crfsuite. *Features* selanjutnya disimpan sebagai X, sedangkan *labels* sebagai y.

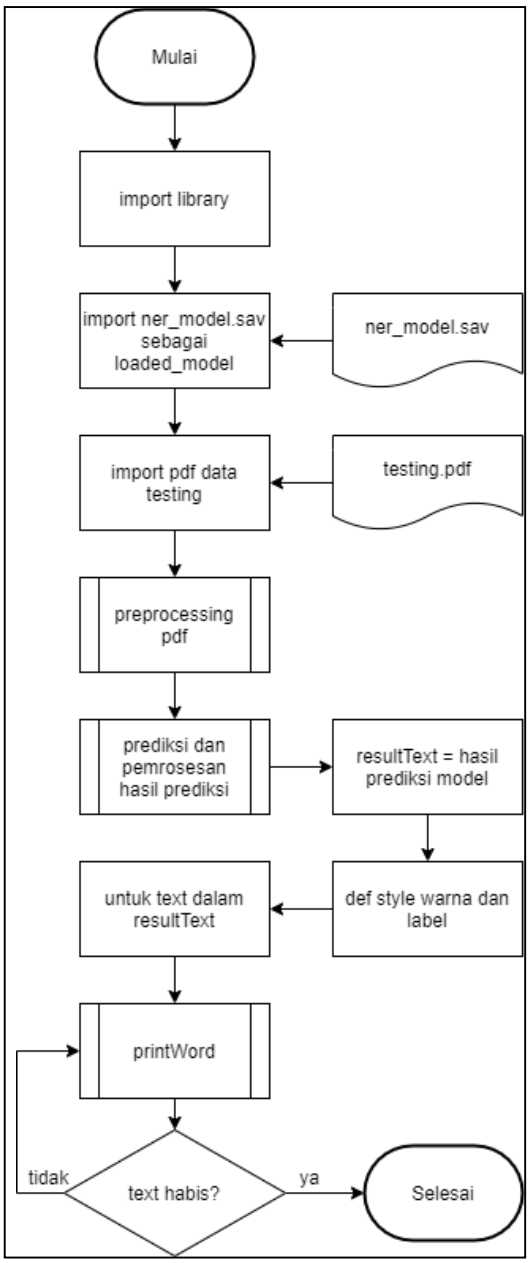
Selanjutnya dilakukan metode K-Fold menggunakan *library* sklearn.model_selection.KFold untuk *hyperparameter tuning*, di setiap iterasinya dilakukan *training* model CRF menggunakan data *training*, *testing* menggunakan data *testing*, dan perhitungan f1-score serta *print* keseluruhan *report*. Untuk model dengan f1-score terbaik disimpan sebagai ner_model.sav untuk digunakan selanjutnya.



Gambar 3.3 Flowchart pelatihan model Named Entity Recognition

3.2.4 Flowchart Testing Eksternal

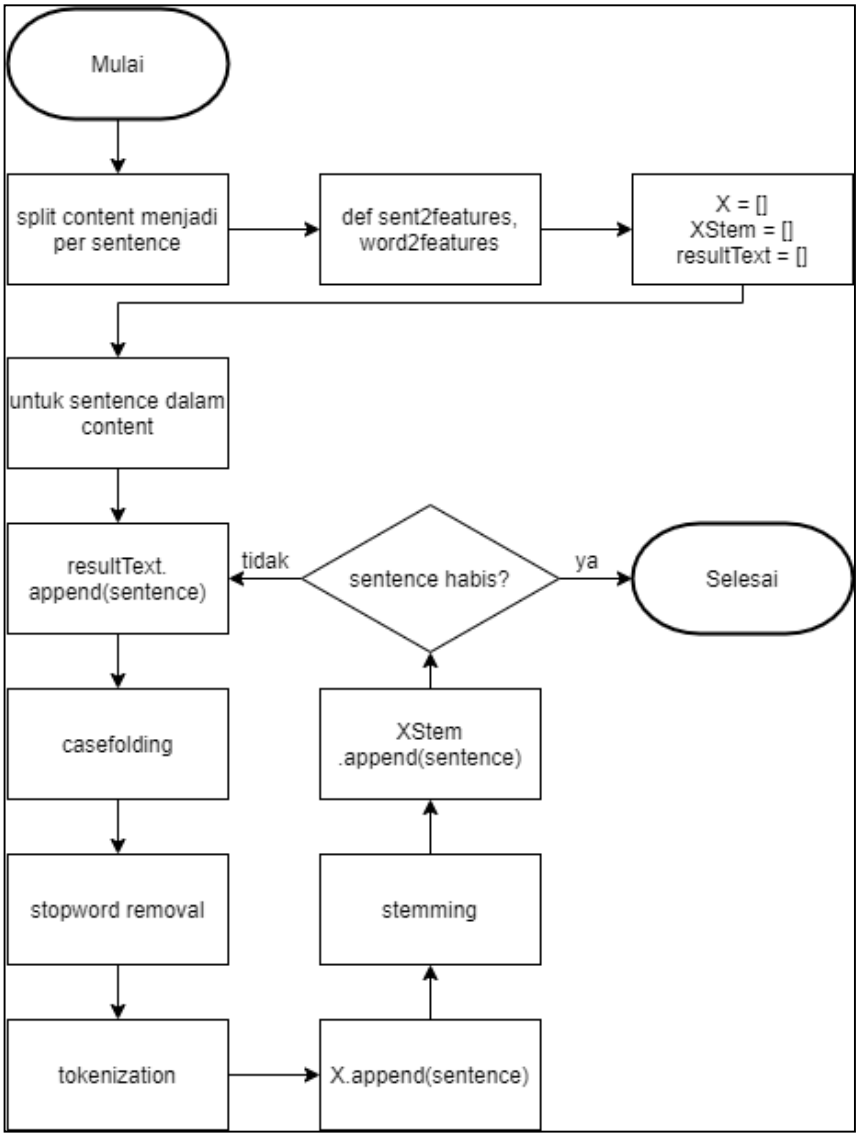
Tujuan dari tahapan ini adalah untuk menerapkan model NER terhadap artikel di luar dataset, serta menampilkan hasil penerapannya. Data yang digunakan merupakan konten artikel digital kesenian bersumber Wikipedia.com yang disusun ulang hanya dalam bentuk pdf. Selanjutnya dilakukan *preprocessing* terhadap pdf serta dilakukan prediksi oleh model NER, dan hasilnya ditampilkan.



Gambar 3.4 Flowchart Testing Eksternal

3.2.5 Flowchart Preprocessing PDF

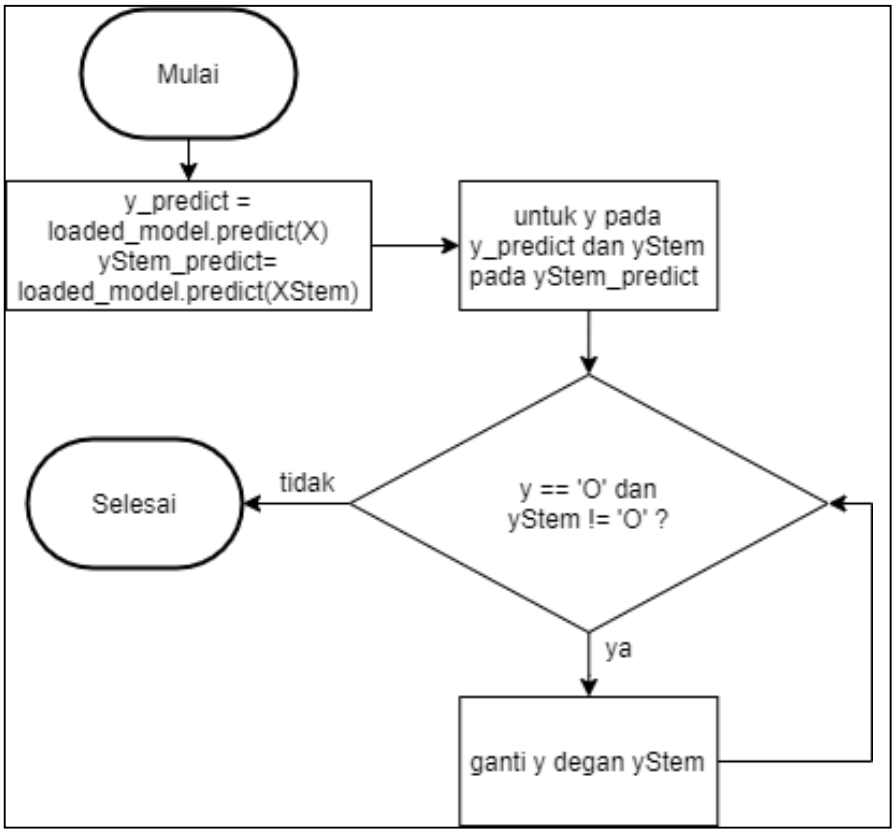
Tahapan *preprocessing* terhadap *file* pdf diawali dengan membagi keseluruhan konten menjadi per kalimat menggunakan library nltk.tokenize. Setiap kalimat selanjutnya diproses dan disimpan ke dalam dua variable berbeda, yaitu X dan XStem. X digunakan untuk menampung hasil *preprocessing* yang melalui tahapan *casefolding, stopword removal,* dan *tokenization* saja; sedangkan XStem untuk menampung hasil *preprocessing* yang dikenakan proses *stemming* juga. Tujuannya agar entitas yang ditulis dengan disertai imbuhan tetap dapat terdeteksi.



Gambar 3.5 Flowchart Preprocessing pdf

3.2.6 Flowchart Predict Testing

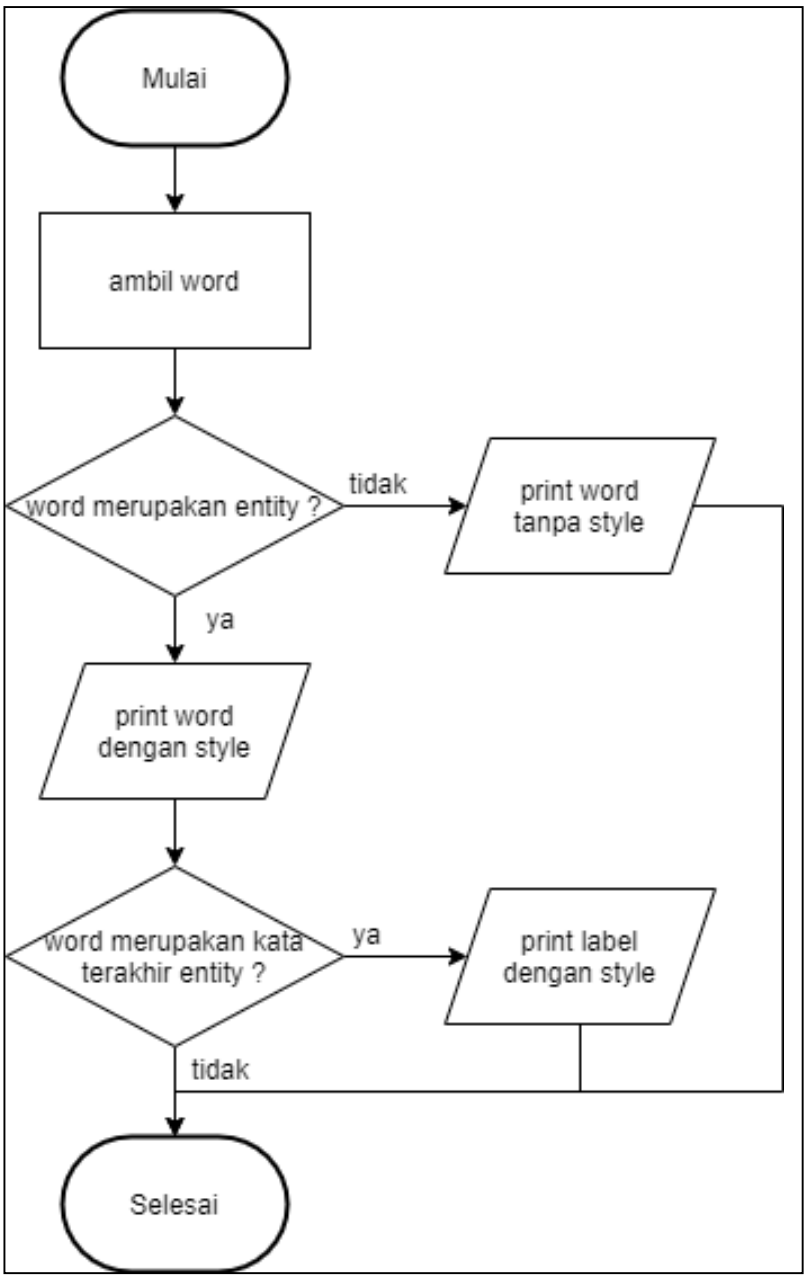
Pada tahap ini, dilakukan prediksi oleh model NER terhadap X dan XStem yang telah dihasilkan. Hasil prediksi disimpan sebagai y_predict dan yStem_predict. Selanjutnya, untuk setiap y_predict, dilakukan pengecekan, apabila nilainya “O” namun nilai yStem_predict untuk index yang sama tidak “O”, maka nilai y_predict untuk index tersebut akan digantikan dengan nilai yStem_predict-nya. Sebagai pengecualian, apabila untuk sebuah index, nilai y_predict dan yStem_predict tidak “O” namun keduanya memiliki nilai berbeda, maka y_predict tidak akan diubah. Dengan demikian, entitas dengan imbuhan dapat dikenali, tetapi tidak mengubah jenis entitas yang memang menggunakan imbuhan (contohnya: kata “penari”, apabila langsung dilakukan *stemming* akan menjadi kata “tari” dan merupakan entitas dengan jenis yang berbeda).



Gambar 3.6 Flowchart Predict Testing

3.2.7 Flowchart Print Word

Untuk setiap kata yang sudah di-*predict* oleh model NER maka akan ditampilkan. Kata yang bukan merupakan entitas akan di-*print* tanpa style apapun. Kata yang merupakan entitas akan di-*print* dengan background sesuai warna yang sudah ditetapkan. Apabila kata tersebut merupakan kata terakhir dari sebuah entitas, maka akan diikuti dengan tampilan labelnya juga.



Gambar 3.7 Flowchart Print Word