BAB 3
METODOLOGI PENELITIAN

3.1 Metode Penelitian

Tahapan metode yang dilaksanakan untuk melakukan penelitian ini adalah sebagai berikut.

3.1.1 Studi Literatur

Studi literatur dilakukan untuk memahami teori terkait implementasi algoritma Support Vector Machines dalam memprediksi tipe kepribadian MBTI berdasarkan konten. Berikutnya, tahap yang diperlukan yaitu mempelajari teknik-teknik *preprocessing* yang perlu diterapkan dalam klasifikasi teks. Selain itu, evaluasi dengan penilaian akurasi, *F1 score*, *precision*, dan *recall* juga perlu dipahami untuk mengetahui efektivitas model.

3.1.2 Pencarian Dataset

Dataset yang digunakan merupakan *dataset* dari Kaggle, sebuah anak perusahaan Google. Kaggle merupakan komunitas *online* dari *data scientists* dan *machine learning*. Kaggle juga menyediakan kumpulan *dataset* dan memungkinkan pengguna untuk menerbitkan *dataset*, mencari dan membangun model pada lingkungan *data science*, serta bekerja bersama *data scientists* lainnya. *Dataset* yang diambil berisi kumpulan konten pengguna dalam berbagai *platform* sosial media dan label yang menunjukkan tipe kepribadian MBTI pengguna tersebut.

3.1.3 Perancangan Model

Pada tahap perancangan model, akan dilakukan perancangan alur kerja untuk memprediksi tipe kepribadian MBTI berdasarkan konten dengan algoritma Support Vector Machines. Perancangan alur kerja dibuat dalam bentuk diagram alur atau *flowchart*. Selama tahap implementasi model, perancangan model dapat kembali disesuaikan.

3.1.4 Implementasi

Implementasi dilakukan dengan membuat *source code* dengan bahasa pemrograman Python. Beberapa proses yang terlibat dalam implementasi antara lain sebagai berikut.

1. Preprocessing

   Tahap ini bertujuan untuk menemukan fitur yang penting pada data. Pada proses ini juga bertujuan untuk mengurangi jumlah variabel dan data yang tidak dibutuhkan. Sehingga, hasilnya yaitu data yang telah diperbaharui sesuai format yang lebih sesuai untuk diproses lebih lanjut. Langkah *preprocessing* yang diterapkan dalam penelitian ini yaitu mengecek data yang kosong, menghapus duplikasi data, menghapus URL atau *hyperlink*, *case folding*, menghapus *stop words*, menghapus simbol, menghapus tipe kepribadian MBTI, serta menghapus kata yang duplikat.

2. Text Mining

   Setelah tahap *preprocessing*, metode yang digunakan yaitu Term Frequency-Inverse Document Frequency (TF-IDF). Metode ini digunakan untuk mencari informasi dalam mengukur kepentingan atau relevansi dari kata dalam suatu dokumen. Pada penelitian ini, data yang akan diukur dengan metode TF-IDF yaitu kolom posts.

3. Handle Imbalance Dataset

   Setiap kelas pada *dataset* yang digunakan akan dikelompokkan dan dihitung. Ketika ditemukan data yang tidak seimbang, *dataset* yang digunakan harus diseimbangkan terlebih dahulu. Teknik yang akan digunakan yaitu *oversampling* dengan Synthetic Minority Oversampling Technique (SMOTE). *Oversampling* dengan SMOTE dilakukan terhadap data posts hasil dari ekstraksi fitur TF-IDF dan data label tipe MBTI.

3.1.5 Pengujian dan Evaluasi

Setelah merancang aplikasi, model akan diuji untuk mengetahui nilai akurasinya serta tingkat kelayakan penggunaan. Evaluasi dilakukan dengan 4 nilai yaitu True Positive, True Negative, False Positive, dan False Negative. Bila hasil

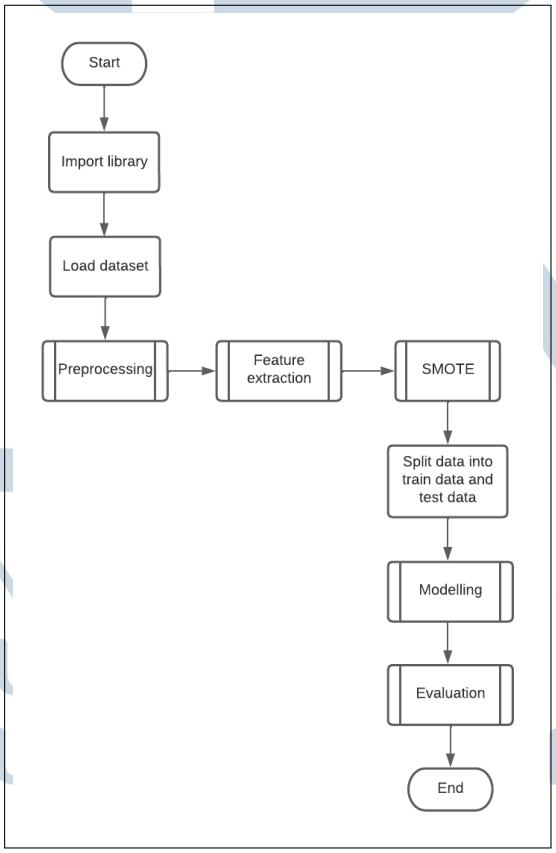
akurasinya kurang baik, data akan kembali dilatih dan memungkinkan adanya perbaikan atau revisi proses pembuatan sistem.

3.1.6 Penulisan Laporan

Penulisan laporan dilakukan sebagai dokumentasi dan penjelasan terhadap teknik-teknik yang digunakan dalam pembuatan model. Selain itu, penulisan laporan juga dapat digunakan sebagai sarana pembelajaran maupun ide untuk penelitian selanjutnya. Penulisan laporan dilakukan bersamaan dengan proses penelitian.

3.2 Perancangan Program

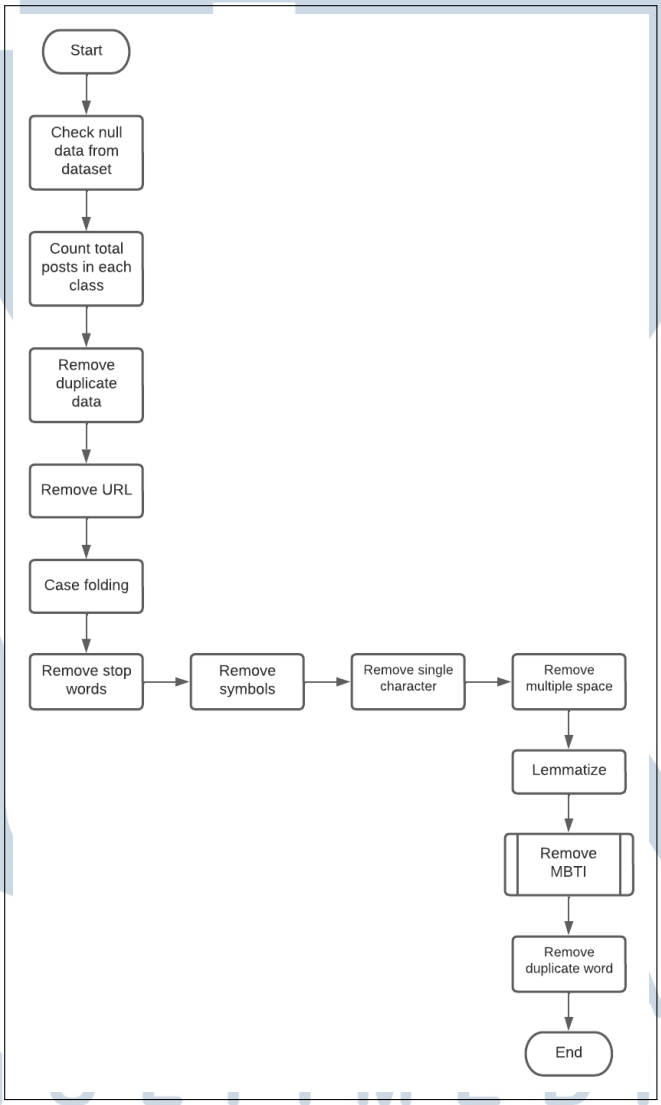
Tahap perancangan program akan menjabarkan alur kerja dari program/model yang dibuat. Alur kerja yang dimaksud akan digambarkan dalam *flowchart*. Berikut merupakan *flowchart* utama dari sistem prediksi MBTI berdasarkan konten media sosial dengan algoritma Support Vector Machines.



Gambar 3.1. *Flowchart* utama

Sistem memiliki enam proses utama yaitu *preprocessing*, *feature extraction*, SMOTE, *data splitting*, *modelling*, dan *evaluation*. Alur kerja sistem dimulai dengan meng-*import library* dan *package* yang akan digunakan selama pelatihan model. Kemudian, tahap berikutnya yaitu memuat *dataset* yang akan digunakan.

3.2.1 Preprocessing

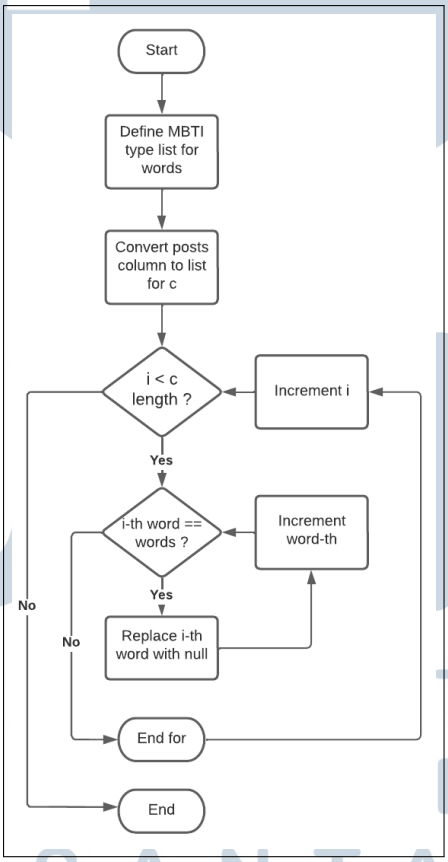


Gambar 3.2. *Flowchart preprocessing*

Gambar 3.2 di atas menunjukkan *flowchart preprocessing* yang terdiri dari beberapa tahap. Tahap awal yaitu mengecek data kosong. Apabila terdapat data kosong, maka data tersebut harus diproses terlebih dahulu. Berikutnya yaitu

menghitung total posts yang terdapat pada setiap kelas. Hal ini berguna untuk mengetahui persebaran data untuk diketahui keseimbangannya. Apabila data tidak seimbang, maka data perlu diproses dengan *undersampling* dan/atau *oversampling*.

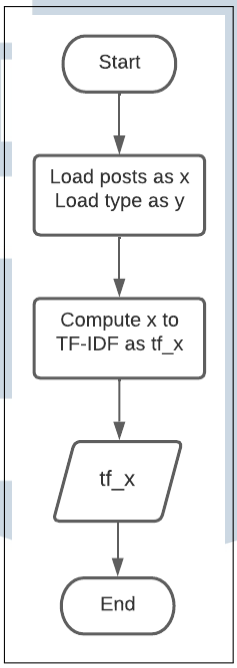
Tahap selanjutnya yaitu menghapus data yang duplikat. Berikutnya, URL akan dihapus dari data, menerapkan *lowercase* pada seluruh posts, menghapus *stop words* (kata-kata yang sering muncul) dalam Bahasa Inggris, menghapus simbol dan emotikon, menghapus karakter tunggal dan beberapa spasi yang tersisa dari hasil *preprocessing* sebelumnya. Setelah itu, *lemmatization* diterapkan pada data posts untuk dikembalikan ke bentuk akar katanya. Kemudian, pada seluruh posts, kata-kata yang merupakan tipe kepribadian MBTI seperti ”INFJ”, ”INTJ”, dan lain-lain akan dihapus dengan *flowchart* seperti pada Gambar 3.3. Tahap terakhir pada *preprocessing* yaitu menghapus kata-kata yang duplikat hasil dari proses *preprocessing* sebelumnya.



Gambar 3.3. *Flowchart remove* MBTI

3.2.2 Feature Extraction

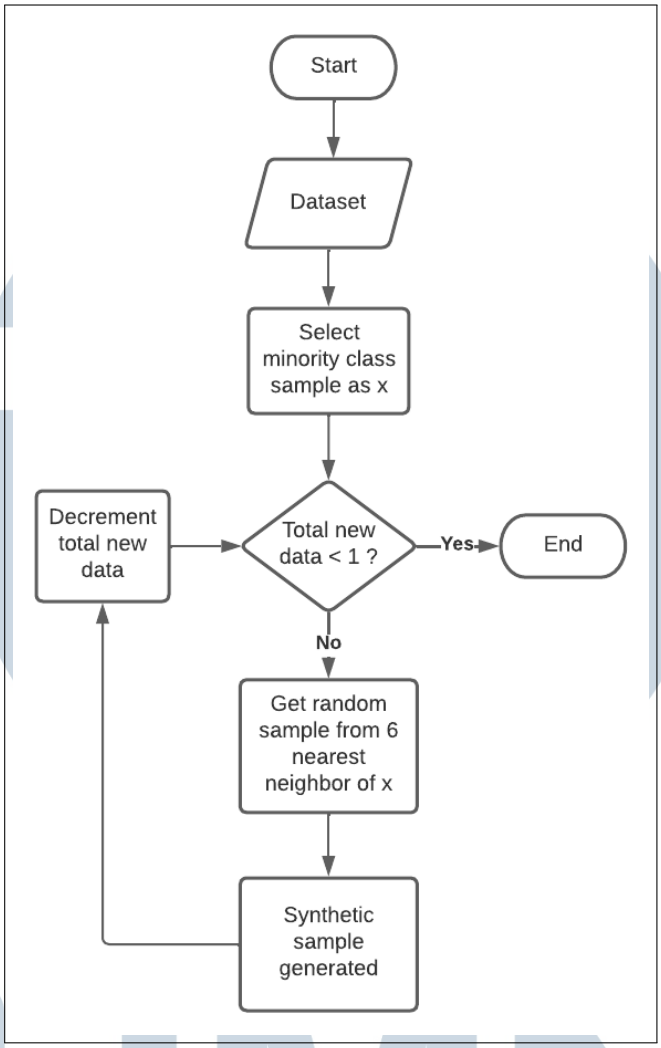
Setelah tahap *preprocessing*, data posts akan diinisiasikan ke variabel x dan data type akan diinisiasikan ke variabel y. Kemudian, x akan ditransformasikan oleh fungsi TfidfVectorizer. Hasil transformasi ini kemudian ditampung ke variabel tf_x. Gambar 3.4 berikut merupakan *flowchart* dari *feature extraction* dengan TF-IDF.



Gambar 3.4. *Flowchart feature extraction*

3.2.3 SMOTE

Tahap berikutnya yaitu *oversampling* dengan SMOTE. Pada tahap ini, kelas minoritas akan dipilih dari *dataset*. Data yang terpilih akan disebut sebagai x. Kemudian, sebanyak total data yang dibutuhkan akan diciptakan *synthetic sample* dari enam tetangga terdekat x secara acak. *Synthetic sample* dihasilkan dari jarak antara sampel acak dan x, yang kemudian dikalikan dengan angka acak antara 0 hingga 1 dan ditambahkan dengan data kelas minoritas tersebut. Berikut merupakan *flowchart* dari SMOTE.



Gambar 3.5. *Flowchart* SMOTE

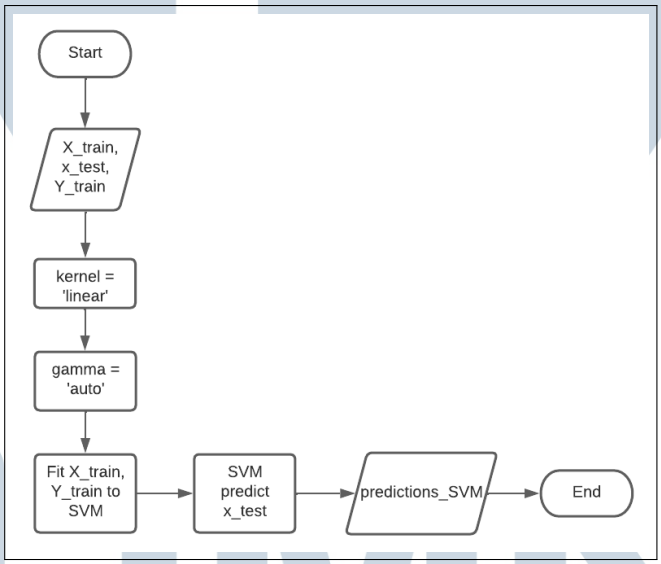
3.2.4 Data Splitting

Pada tahap *data splitting*, semakin besar data latih yang dimiliki, maka performa model akan meningkat, tetapi permodelan dengan algoritma akan mengonsumsi lebih banyak waktu [34]. Akan tetapi, pada suatu titik, setiap penambahan data latih, performa model tidak meningkat, sedangkan waktu pelatihan data terus meningkat. Dengan begitu, peningkatan performa yang didapat tidak sebanding dengan berkurangnya efisiensi. Sehingga, meningkatkan data latih secara berlebihan akan menyebabkan model tidak efisien dan tidak efektif. Berdasarkan penelitian [34] yang meneliti performa model untuk mengklasifikasikan dua dokumen yang berbeda dengan K-fold *evaluation*, menunjukkan bahwa tidak terdapat perbedaan hasil dengan penggunaan *simple holdout case* dengan proporsi 80% data latih dan

20%data uji. Sehingga pada penelitian ini, pembagian data yang dilakukan yaitu dengan 80% data latih dan 20% data uji.

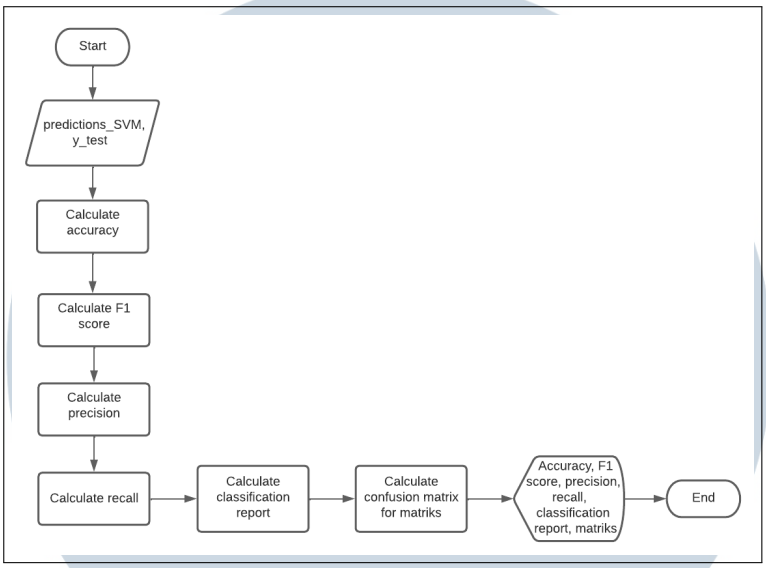
3.2.5 Modelling

Tahap berikutnya yaitu *modelling*. Tahap *modelling* akan menggunakan data X_train, x_test, dan Y_train yang dihasilkan dari proses *data splitting* sebelumnya. Dengan menggunakan tipe kernel linear dan gamma auto, *fitting model* ke SVM dilakukan dengan parameter X_train dan Y_train. Kemudian, SVM akan memprediksi x_test dan ditampung pada variabel predictions_SVM. Gambar 3.6 di bawah ini merupakan *flowchart* dari pelatihan model dengan SVM.



Gambar 3.6. *Flowchart modelling*

3.2.6 Evaluation



Gambar 3.7. *Flowchart evaluation*

Tahap terakhir yaitu tahap *evaluation*. Pada tahap ini, dengan menggunakan hasil sebelumnya yaitu predictions_SVM dan y_test akan dikalkulasikan nilai *accuracy*, *f1 score*, *precision*, *recall*, *classification report*, dan *confusion matrix*. Nilai-nilai ini kemudian akan ditampilkan sebagai bentuk evaluasi dari model. Gambar 3.7 di atas merupakan *flowchart* dari tahap *evaluation*.