Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, 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.

Perpustakaan Univeristas Multimedia Nusantara

BAB III

METODOLOGI PENELITIAN DAN PERANCANGAN SISTEM

3.1 Metodologi Penelitian

Dalam penelitian Implementasi Opinion Mining Komentar Channel Video Pelayanan Pemerintah Di Youtube Menggunakan Naïve Bayes ini akan menggunakan metode penelitian sebagai berikut.

3.1.1 Studi Literatur

Melakukan studi literature dengan cara mempelajari dan membaca penelitian-penelitian sebelumnya mengenai pelayanan public, *text mining* dengan NGram, *bigram* dan TF-IDF, analisis sentimen, *naïve bayes classifier*, *precision*, *recall* dan *accuracy*.

3.1.2 Pengumpulan Dataset

Dataset yang digunakan adalah dataset yang telah diolah oleh penulis penelitian sebelumnya yaitu sebanyak 911 data dari *channel* KemenPU, data tersebut terdiri dari 587 baris data berlabel positif, 214 data berlabel netral, dan 110 data berlabel negatif. Proses pengambilan data yang dilakukan diawali dengan penarikan data komentar pada *channel* video pelayanan pemerintah pada YouTube dengan menggunakan fasilitas *Application Programming Interface* (API), kemudian disimpan.

3.1.3 Pengembangan Sistem

Tahap pengembangan sistem yang dilakukan adalah *training* dataset, *coding model*, memodifikasi model. Tahap pengembangan sistem dilakukan atas pembuatan alur kerja untuk penerapan NGram, TF-IDF, dan *Gaussian Naïve Bayes*.

3.1.4 Pengujian dan Evaluasi

Tahap pengujian terhadap performa ini dilakukan untuk menguji apakah sistem yang dibangun sudah berjalan sesuai dengan seharusnya atau belum, serta melakukan perbaikan jika terjadi ketidak sesuaian atau adanya kesalahan.

Pada tahap evaluasi, proses dilakukan menggunakan *confusion matrix.* Proses evaluasi dilakukan pada hasil klasifikasi dengan menghitung tingkat *precision*, *recall*, *accuracy*, dan F1 *score* dari hasil klasifikasi teks yang dilakukan.

3.1.5 Penulisan Laporan

Tahap terakhir dalam penelitian ini adalah menulis laporan yang berisikan hasil keseluruhan penelitian. Penulisan dilakukan secara terstruktur sesuai dengan kaidah penulisan laporan ilmiah. Penulisan laporan dilakukan secara paralel sembari melakukan penelitian, sehingga proses dapat berjalan sesuai rencana dan tidak ada yang terlewatkan dalam penulisan laporan. Penulisan laporan juga dilakukan secara bertahap, dimulai dari pendahuluan, latar belakang, kesimpulan, dan saran.

3.2 Perancangan Sistem

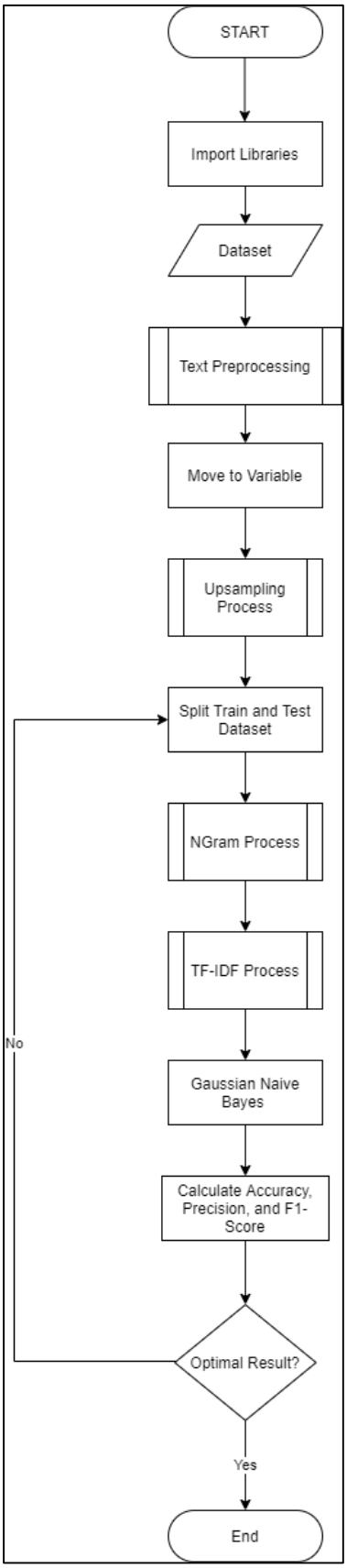
Sebelum dilakukan penelitian analisis sentimen, dibuat perancangan sistem agar dapat memahami alur kerja sistem tersebut secara keseluruhan. Perancangan sistem yang dibangun dijelaskan dalam *flowchart* yang dimulai dengan *preprocessing* sampai dengan pembuatan model, serta menjelaskan tentang tahapan dalam penggunaan *Gaussian Naïve Bayes* untuk analisis sentiment.

3.2.1 Flowchart Proses Klasifikasi

3.2.1.1 Flowchart Utama Sistem

Pada Gambar 3.1 merupakan gambar *flowchart* atau diagram alir dari proses utama klasifikasi sentimen menggunakan Gaussian Naïve Bayes secara umum. Langkah pertama pada proses klasifikasi adalah *import library* yang dibutuhkan untuk mendukung proses penelitian. Kemudian proses selanjutnya adalah *import* Dataset bernama "DataSet" dalam bentuk *file csv* untuk dapat diolah. *File* dataset yang digunakan berisi ulasan pengguna kepada *channel video* Pemerintah di Youtube. *File* tersebut dibaca menggunakan *library* Python yaitu *pandas*. Kemudian masuk pada proses selanjutnya yaitu proses *Preprocessing*. Proses ini dilakukan untuk mengolah data mentah menjadi data yang berkualitas. Pada proses ini dilakukan proses pengubahan teks pada Dataset menjadi Lowercase, menghilangkan angka, menghilangkan tanda baca, menghilangkan emoji dan memindahkan data ke dalam *variable*. Lalu setelah proses *Preprocessing* selesai dilakukan, proses selanjutnya adalah proses *Upsampling*. Proses ini digunakan untuk menggandakan jumlah data yang berada pada label negatif dan netral untuk menyeimbangi jumlah data pada label positif. Data label positif dijadikan acuan dalam proses *Upsampling* karena label positif memiliki jumlah data terbanyak (*majority)*. Setelah proses *Upsampling*, data akan memasuki proses *bigram*. Proses Ngram digunakan untuk memprediksi kata berikutnya yang mungkin dari kata N-1 sebelumnya. Model ini memberikan probabilitas kemungkinan pada kata berikutnya yang mungkin dapat digunakan untuk melakukan proses kemungkinan penggabungan pada keseluruhan kalimat. Model ini mengelola semua teks dan menggerakkan satu kata maju ke depan untuk membangkitkan Ngram berikutnya.

Kemudian proses selanjutnya adalah Term Frequency – Inverse Document Frequency (TF-IDF). Proses ini merupakan suatu metode algoritma yang berguna untuk menghitung bobot setiap kata yang telah memasuki proses Ngram. Metode ini akan menghitung nilai *term* atau seberapa sering suatu *token* atau kata yang muncul di dalam data atau dokumen. Setelah itu, proses yang akan dilakukan adalah proses dari *Gaussian Naïve Bayes*. Algoritma ini digunakan dengan *library* Python yaitu *Naïve Bayes*. Metode klasifikasi ini digunakan untuk memperkirakan distribusi data dengan mengasumsikan penyederhanaan nilai atribut bebas bersyarat ketika diberikan nilai *output*. Proses ini digunakan untuk menghitung probabilitas untuk nilai input pada setiap kelas menggunakan frekuensi dan untuk mengetahui apakah hasil sudah optimal atau belum, maka dilakukan proses perhitungan akurasi dan menampilkan hasil prediksi dan aktual menggunakan *library* Python yaitu *numpy* dan *pandas* dalam bentuk tabel yang berikut dengan analisa hubungan antar *variable* dalam baris dengan *variable* dalam kolom dari hasil prediksi dan aktual setiap label.

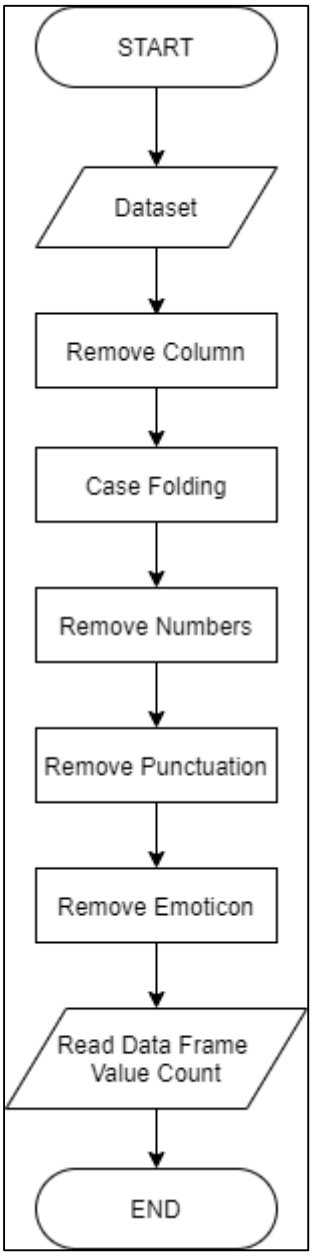


Gambar 3.1 Flowchart Proses Secara Umum

3.2.1.2 Flowchart Proses Preprocessing

Pada proses selajutnya setelah proses Dataset dibaca, Dataset akan masuk ke dalam tahap *Preprocessing*. Dalam tahap *Preprocessing*, terdapat beberapa tahap yang dilakukan seperti yang tertera pada Gambar 3. 2, berikut adalah tahapan proses pada *Preprocessing*:

1. *Remove Column*, tahap ini dilakukan untuk membuang kolom yang tidak diperlukan dalam proses analisis sentimen. Kolom yang diperlukan hanya kolom Komentar dan kolom Label hasil *voting*. Kolom yang dibuang adalah kolom Judul, Waktu, Nama, Nilai 1, Nilai 2, dan Nilai 3.
2. *Case Folding*, tahap ini dilakukan untuk mengubah setiap huruf besar (Uppercase) pada kalimat menjadi huruf kecil (Lowercase).
3. *Remove Numbers*, tahap ini dilakukan untuk menghapus semua angka yang ada pada kalimat.
4. *Remove Punctuation*, tahap ini dilakukan untuk menghilangkan tanda baca yang ada pada kalimat.
5. *Remove Emoticon*, tahap ini dilakukan untuk menghilangkan emoji yang ada pada kalimat.
6. Kemudian tahap selanjutnya adalah membaca *value data* yang telah diproses.

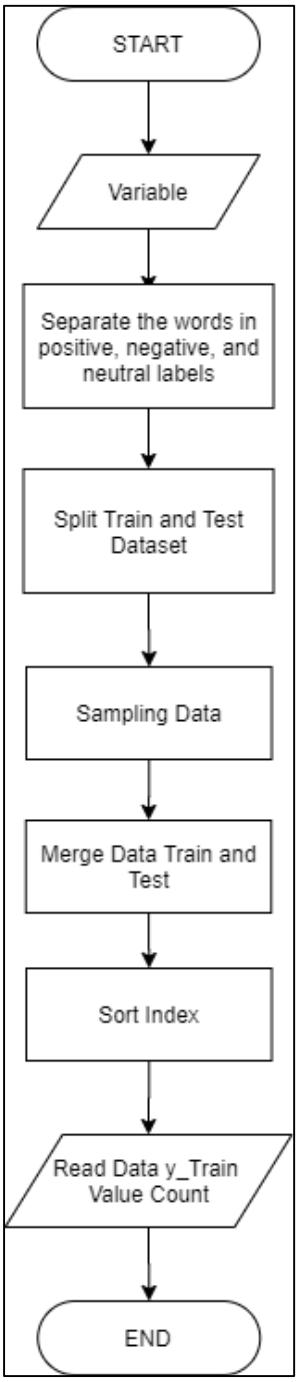


Gambar 3.2 Flowchart Preprocessing

3.2.1.3 Flowchart Proses Upsampling

Setelah proses *Preprocessing* dilakukan, Dataset yang telah mengalami proses tersebut dipindahkan ke dalam *variable* untuk diproses ke tahap selanjutnya. Tahap selanjutnya adalah proses Upsampling. Proses Upsampling digunakan untuk

menggandakan jumlah data yang berada pada label negatif dan netral untuk menyeimbangi jumlah data pada label positif. Pada proses Upsampling ini, label positif dijadikan acuan dalam proses Upsampling, hal ini dikarenakan jumlah data yang berlabel positif adalah jumlah data yang paling banyak (majority). Pada Gambar 3. 3 merupakan alur proses dari tahap Upsampling. Proses awal dari tahap Upsampling adalah data dibagi kedalam tiga label, yaitu positif, negatif dan netral. Kemudian masing-masing data di split ke dalam data train dan data test berdasarkan labelnya dengan rasio masing-masing sebesar 80% pada data train dan 20% pada data test. Lalu proses selanjutnya setelah data di split adalah proses sampling, pada proses ini menggunakan data label positif yang dijadikan acuan karena data label positif merupakan data label yang memiliki jumlah data paling banyak. Lalu proses selanjutnya adalah penggabungan data Train dan data test untuk masuk ke tahap selanjutnya. Setelah proses penggabungan ke masing-masing label pada data train dan data test, data akan memasuki proses *sort* untuk mengurutkan data agar berurutan.

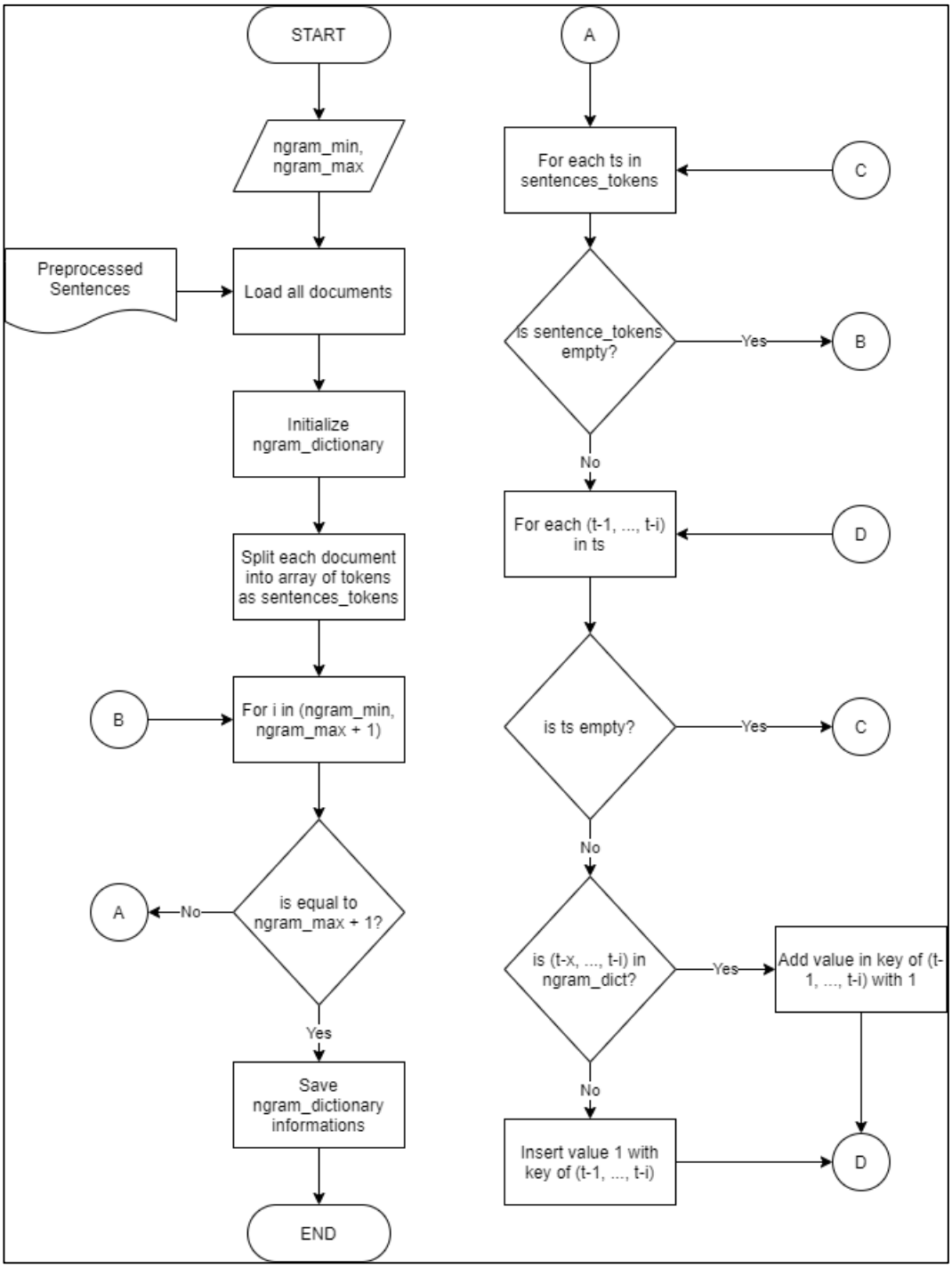


Gambar 3.3 Flowchart Upsampling

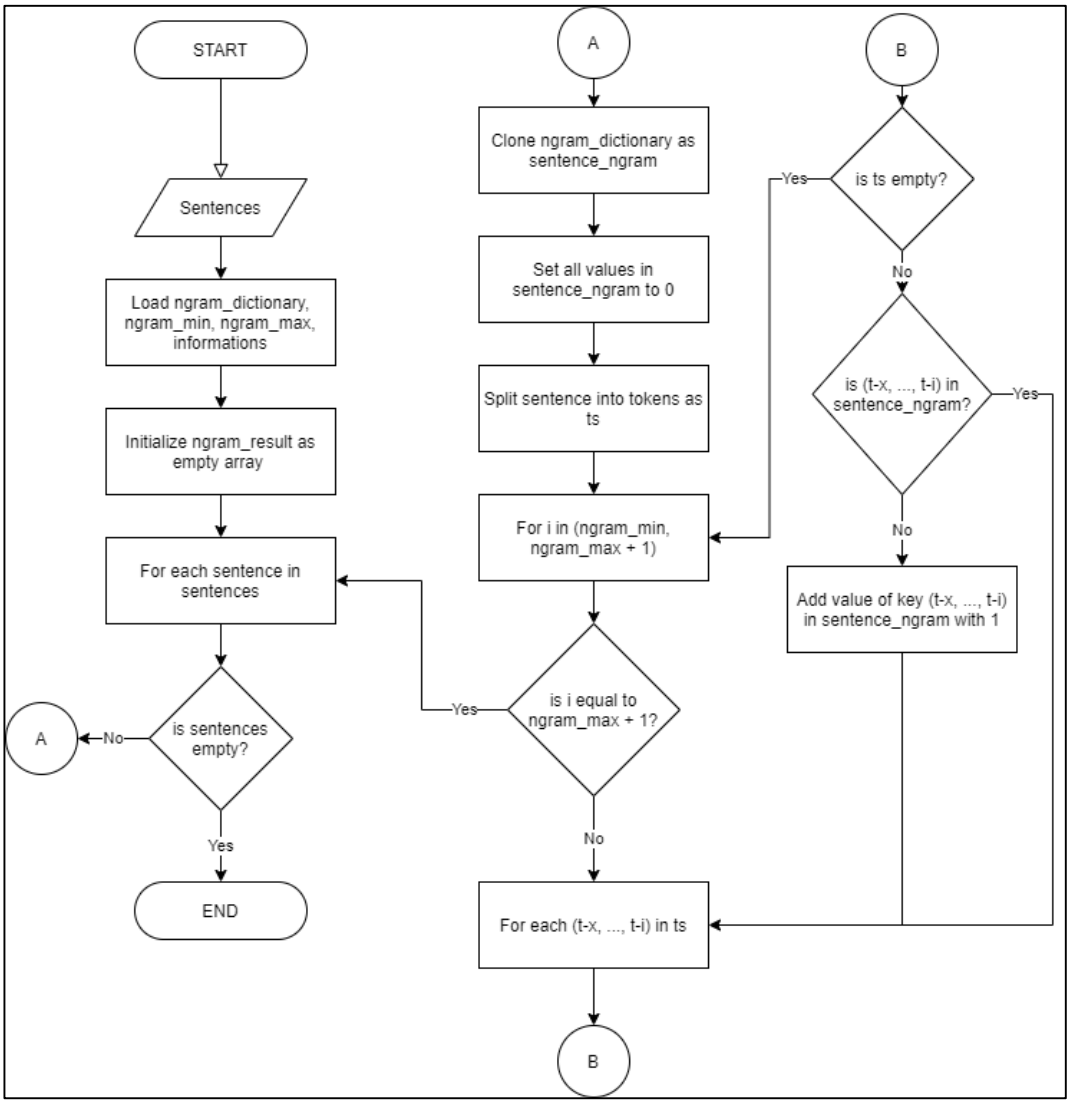
3.2.1.4 Flowchart Proses Ngram

Gambar 3. 4 merupakan proses pertama dari proses Ngram yaitu sampai dengan proses *fit* data, dan Gambar 3. 5 merupakan gambar flowchart dari proses Ngram bagian data *transform*. Proses pertama dilakukan dengan menerima *input* nilai minimal dan maksimal dari Ngram. Kemudian, semua data yang telah mengalami proses *Preprocessing* diambil. Proses selanjutnya adalah membagi setiap kata yang ada di dalam data. Jika kata tersebut tidak ada di dalam *dictionary*, maka akan ditambahkan ke dalam *dictionary* dengan *value* 1. Jika data tersebut sudah ada dalam *dictionary*, maka *value* dari *key* data tersebut akan bertambah 1. Proses penambangan *value* pada *key* dilakukan sepanjang data dimulai dari nilai Ngram minimum sampai dengan nilai maksimum + 1 dari Ngram.

Setelah proses *fit* data dilakukan proses kedua yaitu data *transform*. Proses *transform* dilakukan dengan menyesuaikan model dengan data yang sudah mengalami proses *fit*.



Gambar 3.4 Flowchart Proses Ngram

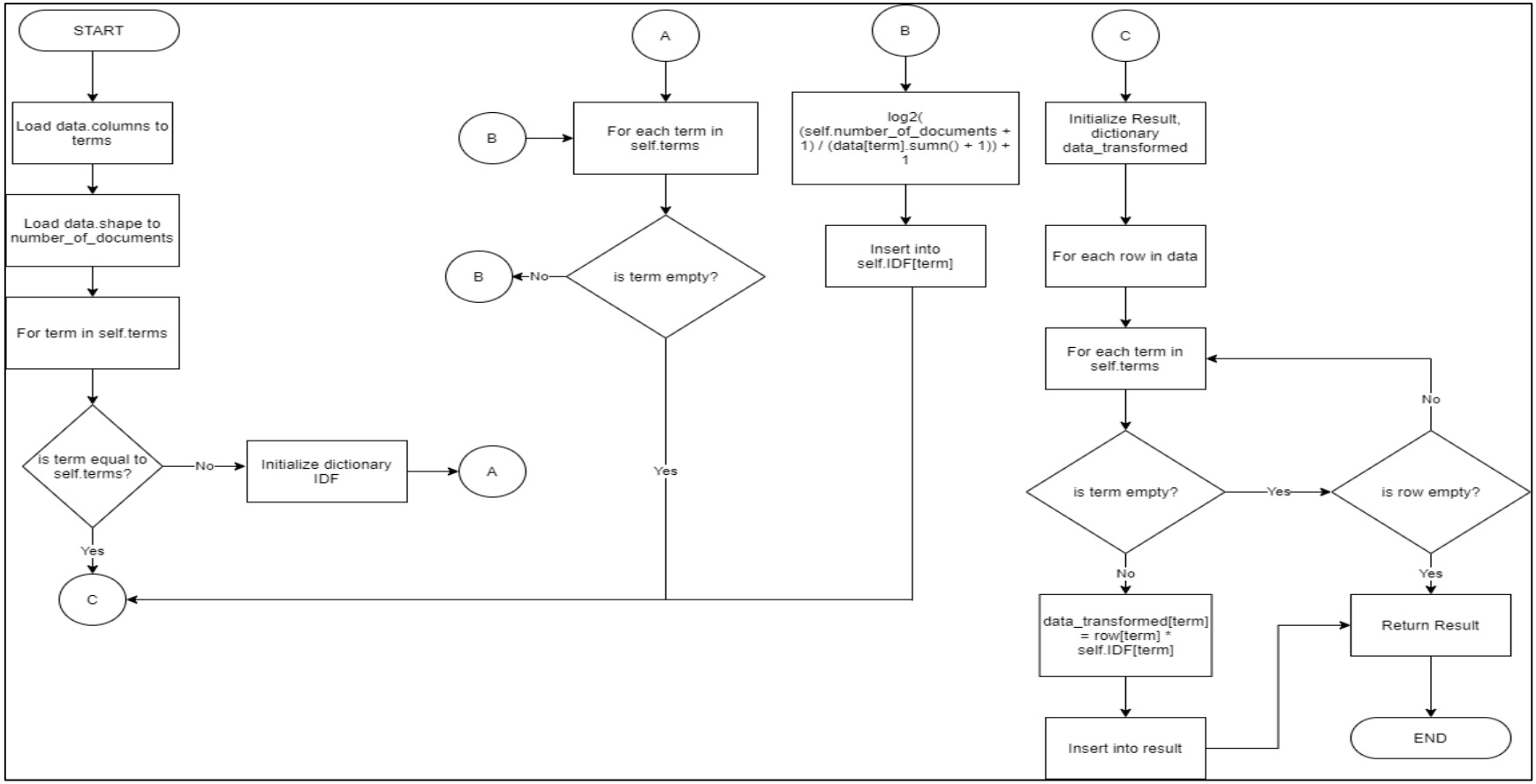


Gambar 3.5 Proses Flowchart Ngram Lanjutan

3.2.1.5 Flowchart Proses TF-IDF

Gambar 3. 6 merupakan flowchart dari proses TF-IDF. Proses ini diawali dengan memproses data.columns ke data terms. kemudian, data tersebut memproses data.shape ke dalam number_of_documents. Untuk setiap term pada data dilakukan proses perhitungan IDF dengan rumus self.IDF[term] = math.log2((self.number_of_documents + 1) / (data[term].sum() + 1)) + 1. Proses

selanjutnya dilakukan proses *transform* untuk setiap data term dengan rumus data_transformed[term] = row[term] * self.IDF[term].



Gambar 3.6 Flowchart Proses TF-IDF