



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.

BAB III

METODE DAN PERANCANGAN SISTEM

3.1 Metodologi Penelitian

Sistem yang dikembangkan akan berfungsi untuk mendapatkan ringkasan dari suatu berita. Penelitian ini dilakukan dengan menggunakan metode TextRank. Sistem yang dirancang akan memproses data masukan yang berupa berita, kemudian sistem akan menampilkan hasil dari proses tersebut. Berikut tahapan-tahapan yang akan dilakukan dalam penelitian ini.

1. Studi Literatur

Melakukan studi mengenai teori-teori dan konsep yang berhubungan dengan penelitian yang dilakukan, seperti teori mengenai *text preprocessing*, *text summarization*, dan juga algoritma TextRank. Selain itu juga akan dilakukan pengumpulan data lain yang dapat mendukung penelitian. Referensi yang digunakan dalam penelitian ini adalah buku, jurnal, internet, dan sumber-sumber lainnya yang dapat membantu proses penelitian.

2. Perancangan Sistem

Melakukan perancangan awal terhadap sistem yang dibangun seperti *flowchart diagram*. Diawali dengan sistem menerima *input* berupa berita dan *output* berupa hasil ringkasan dari berita tersebut.

3. Implementasi

Melakukan pembangunan aplikasi dengan menggunakan algoritma TextRank. Selanjutnya dilakukan uji coba terhadap sistem untuk mengetahui kinerja sistem.

4. Uji Coba dan Evaluasi

Melakukan uji coba terhadap sistem yang telah dikembangkan dan melakukan evaluasi terhadap sistem berdasarkan hasil uji coba yang dilakukan. Dalam melakukan evaluasi terhadap sistem, hasil ringkasan yang diperoleh dari sistem dibandingkan dengan ringkasan manual yang sebelumnya telah didapatkan dari seorang pakar Bahasa Indonesia lalu dinilai sesuai dengan metode yang telah disebutkan sebelumnya.

3.2 Perancangan Sistem

Dalam proses pembuatan sistem akan dilakukan perancangan awal terlebih dahulu sehingga memudahkan untuk mengetahui alur kerja sistem. Rancangan sistem yang dibuat adalah sebagai berikut.

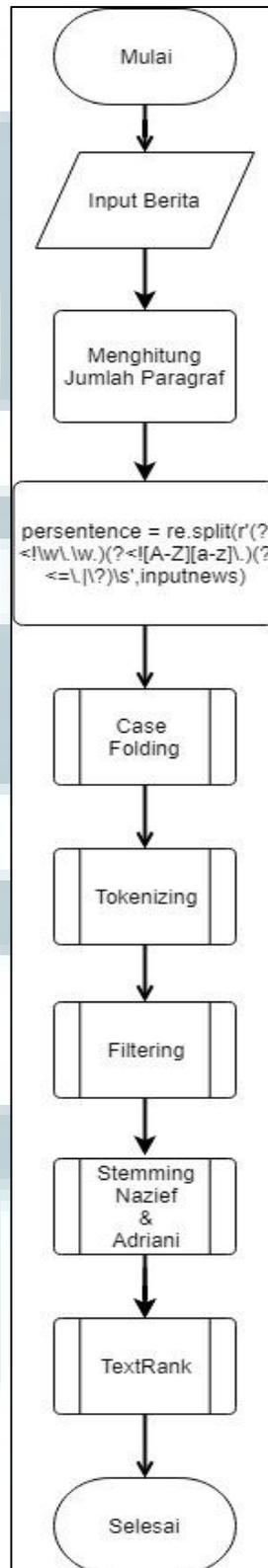
3.2.1 Flowchart

Flowchart berfungsi untuk menggambarkan alur kerja sistem secara keseluruhan. Alur kerja dari sistem yang dikembangkan adalah sebagai berikut.

A. Flowchart Sistem Peringkasan Berita

Sistem akan menerima *input* berupa berita. Setelah itu, berita yang telah diterima akan diproses dengan 7 tahap yaitu menghitung jumlah paragraph yang digunakan sebagai acuan hasil sistem, pemecahan kalimat yang diuraikan menggunakan *function split()* untuk memecah kalimat-kalimat pada satu kesatuan berita tersebut, *case folding*, *tokenizing*, *filtering*, *stemming* Nazief & Adriani, dan

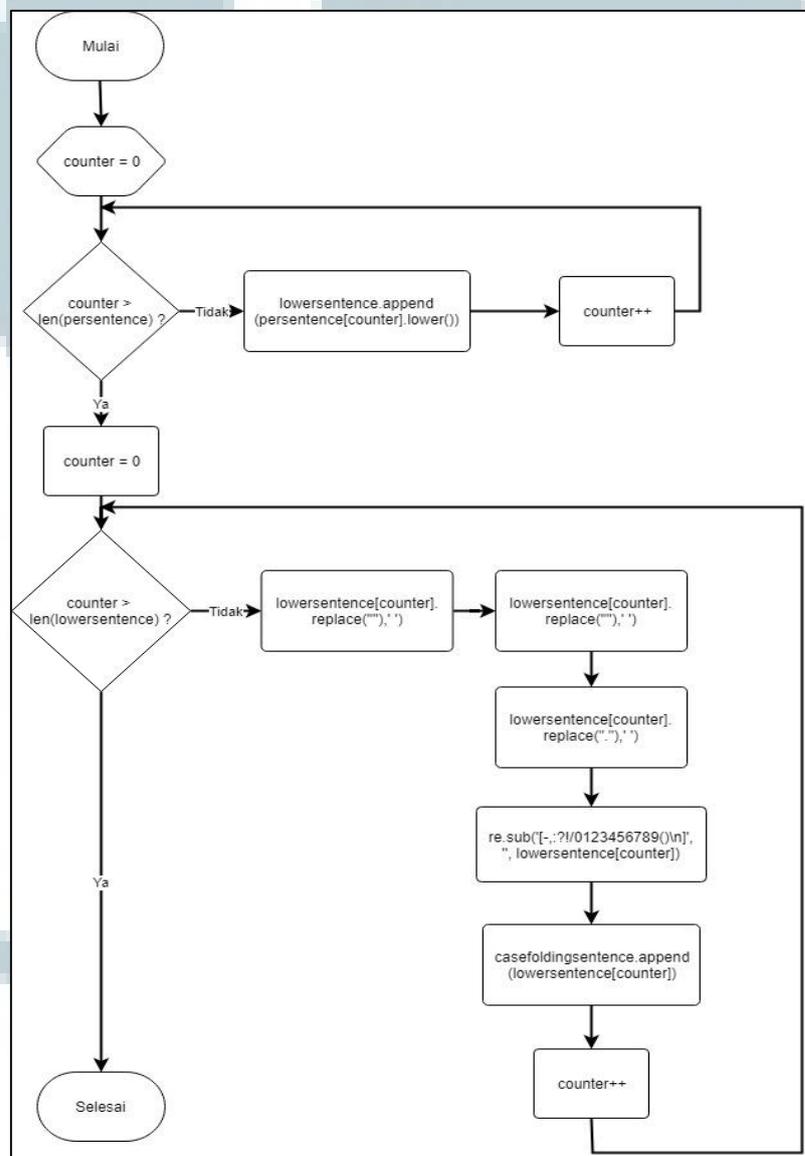
TextRank. Hasil perhitungan tersebut akan digunakan untuk mendapatkan *output* dari sistem. Gambar 3.1 merupakan *flowchart* sistem.



Gambar 3.1 Flowchart Sistem

B. Flowchart Case Folding

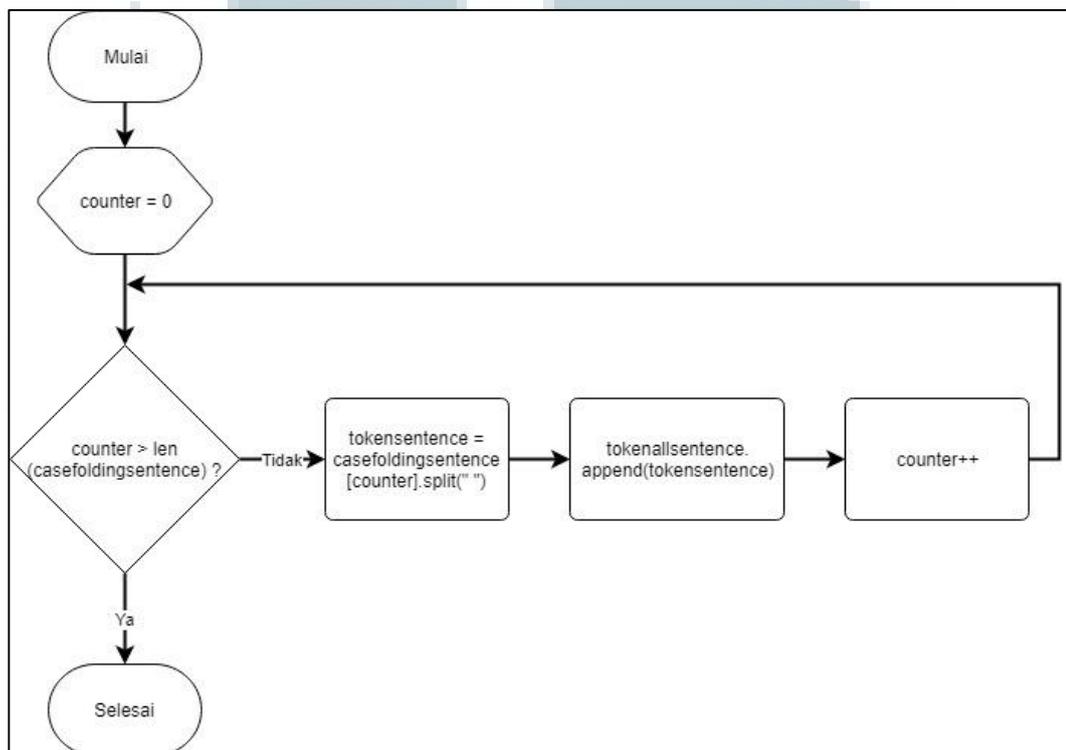
Pada proses *case folding*, berita yang telah dipecah-pecah kalimatnya akan dirubah semua huruf pada masing-masing kalimat menjadi huruf kecil dengan menggunakan iterasi for dan `lower()` untuk membuat karakter menjadi huruf kecil. Lalu dilanjutkan dengan proses menghilangkan karakter selain huruf terhadap semua kalimat yang sudah dilakukan `lower`. Setelah proses, hasil dari proses *case folding* dikirim ke proses selanjutnya. Gambar 3.2 merupakan *flowchart case folding*.



Gambar 3.2 Flowchart Case Folding

C. Flowchart Tokenizing

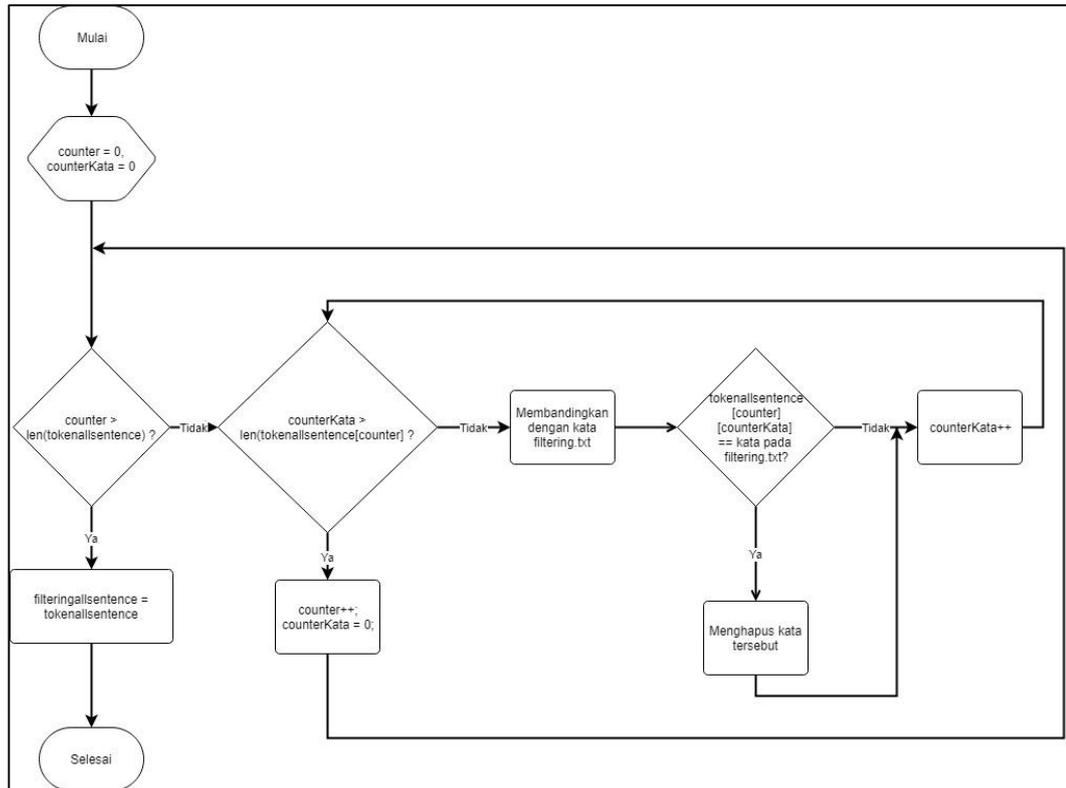
Pada proses *tokenizing*, kalimat pada berita yang sudah dilakukan *case folding* akan dipecah per kata pada kalimat tersebut dengan *delimiter* spasi. Setelah itu, dimasukkan ke sebuah *array* baru yang berisi pemecahan kata pada setiap kalimat. Gambar 3.3 merupakan *flowchart tokenizing*.



Gambar 3.3 Flowchart Tokenizing

D. Flowchart Filtering

Pada proses *filtering*, kata-kata pada berita yang sudah di *tokenizing* diiterasi sesuai kalimat yang ada pada *array* dan akan dibandingkan dengan kata pada *stopwords* yang ada pada *filtering.txt*. Jika kata pada berita tersebut sesuai dengan kata pada *filtering.txt* maka kata tersebut akan dihapus dari *array* tersebut, tetapi jika kata pada berita tersebut tidak sesuai dengan kata pada *filtering.txt* yang ada maka kata tersebut tidak akan dihapus. Gambar 3.4 merupakan *flowchart filtering*.

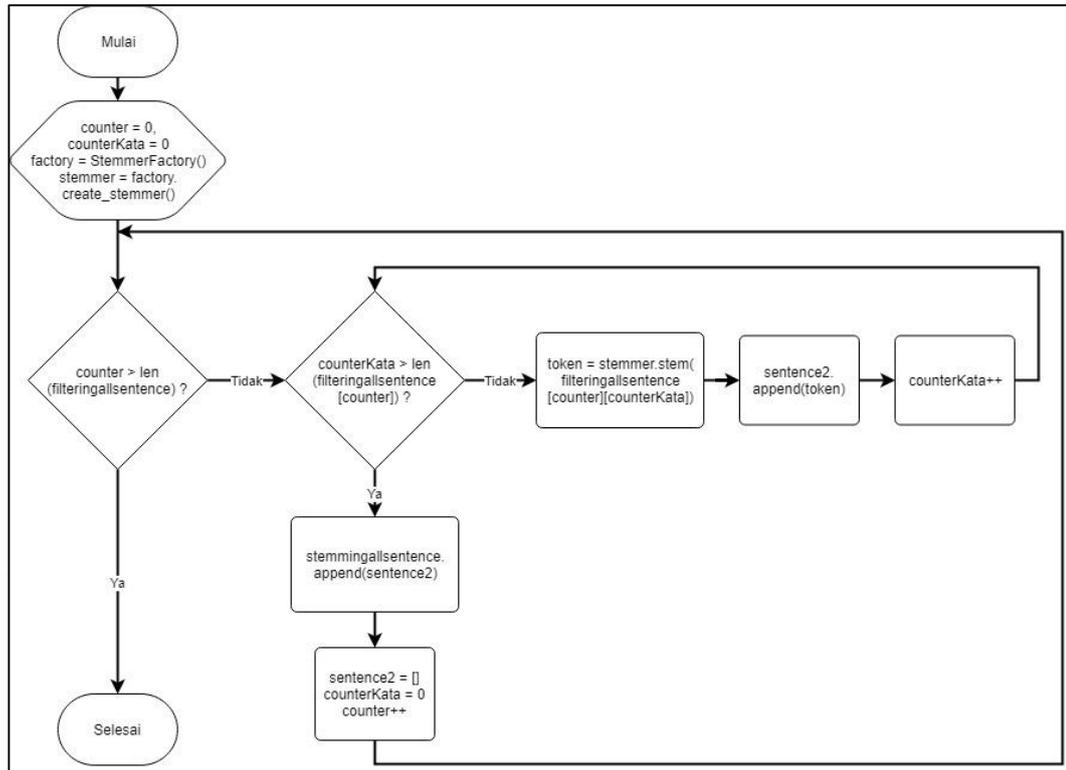


Gambar 3.4 Flowchart Filtering

E. Flowchart Stemming Nazief & Adriani

Pada proses *stemming* Nazief & Adriani, tujuan akhir adalah mencari kata dasar pada setiap kata di berita dengan mengiterasi *array* yang sudah diproses sebelumnya dan ingin diringkas. API Sastrawi adalah salah satu API yang menggunakan proses *stemming* Nazief & Adriani.

Cara penggunaannya adalah menginisialisasi variabel pada API Sastrawi lalu melakukan *stemming* pada setiap kata pada kalimat yang sudah di *tokenizing* dan *filtering*. setelah melakukan hal tersebut di masukkan ke dalam sebuah *array* baru yang sudah berisi pemecahan kata tiap kalimat yang sudah dijadikan kata dasarnya. Gambar 3.5 merupakan *flowchart stemming* Nazief & Adriani.



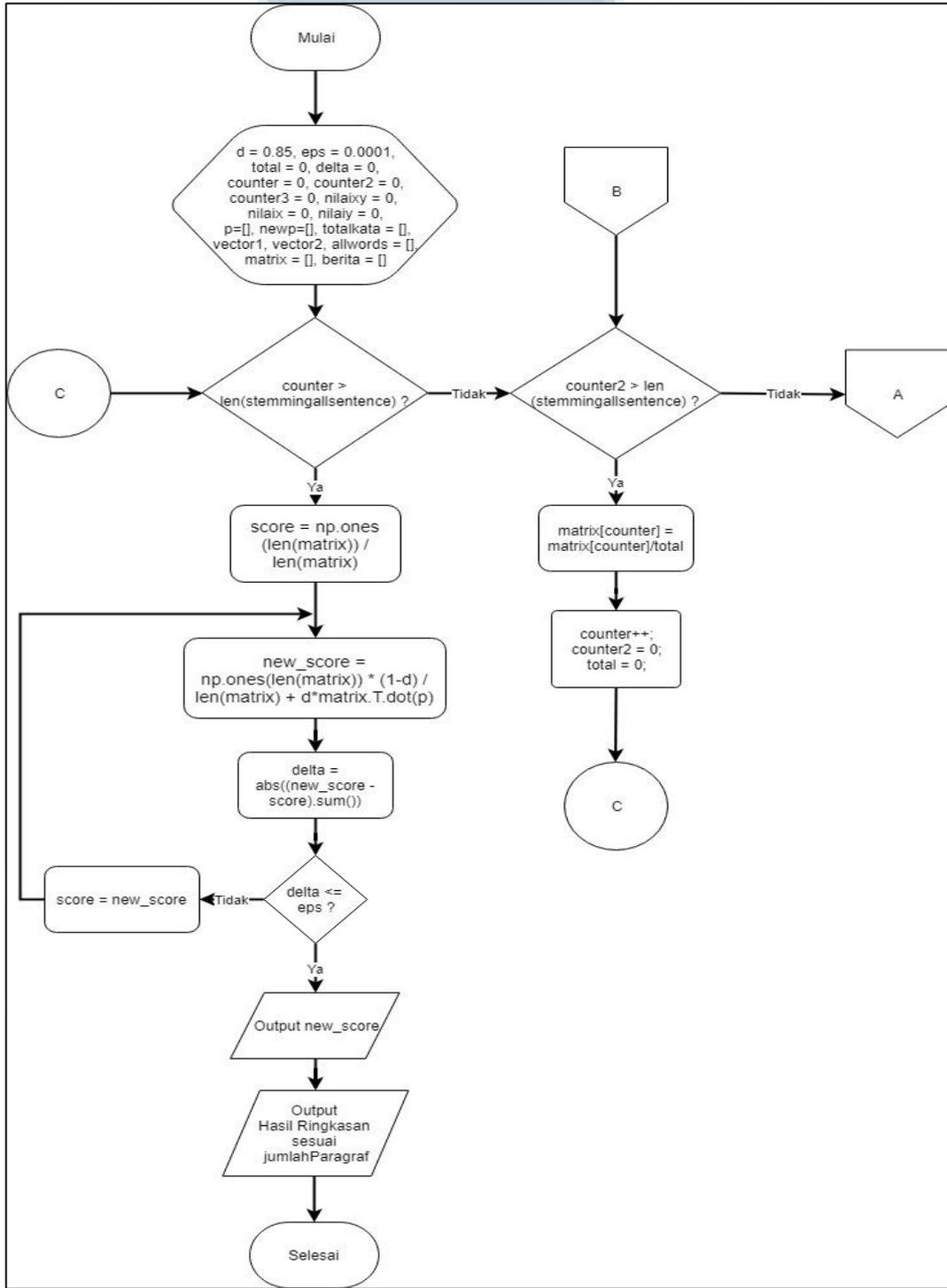
Gambar 3.5 Flowchart Stemming Nazief & Adriani

F. Flowchart TextRank

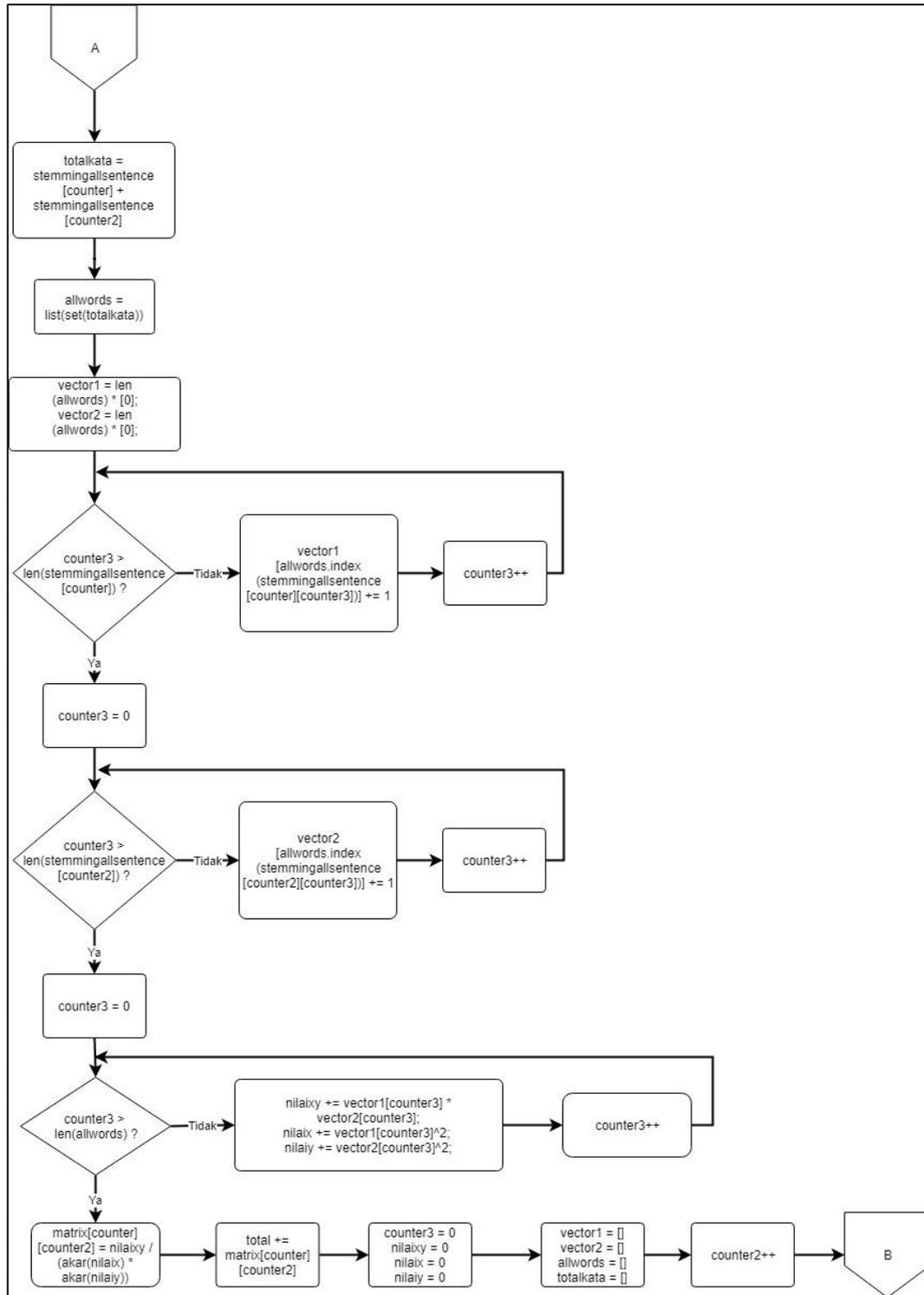
Pada proses algoritma TextRank, diawali dengan menghitung jumlah kata pada tiap dua kalimat berita setelah itu dilakukan pembuatan *vector* dari kedua kalimat berdasarkan jumlah kata pada dua kalimat tersebut, lalu menambah nilai bobot kata ketika kata tersebut berada pada kalimat. Setelah itu, dilakukan rumus *cosine similiarity* untuk menentukan nilai persamaan di antara kedua kalimat tersebut.

Kemudian, setelah melakukan perbandingan ke semua kalimat maka dibuatlah sebuah *adjacency matrix* antar kalimat pada berita tersebut dan juga dilakukan normalisasi *matrix* pada setiap *row*. Lalu, menggunakan rumus TextRank untuk melaksanakan algoritmanya, dibandingkan hasil dengan nilai yang lama mengitung delta nilai tersebut, jika delta sudah lebih kecil dari *threshold* (*eps*) maka

sudah tercapai ringkasan tersebut. Namun jika delta belum lebih kecil dari *threshold* maka akan melakukan *looping* ke rumus TextRank lagi dan nilai yang lama menjadi nilai perhitungan rumus yang baru. Gambar 3.6 dan Gambar 3.7 merupakan adalah *flowchart* TextRank.



Gambar 3.6 Flowchart TextRank



Gambar 3.6 Flowchart TextRank (Lanjutan)