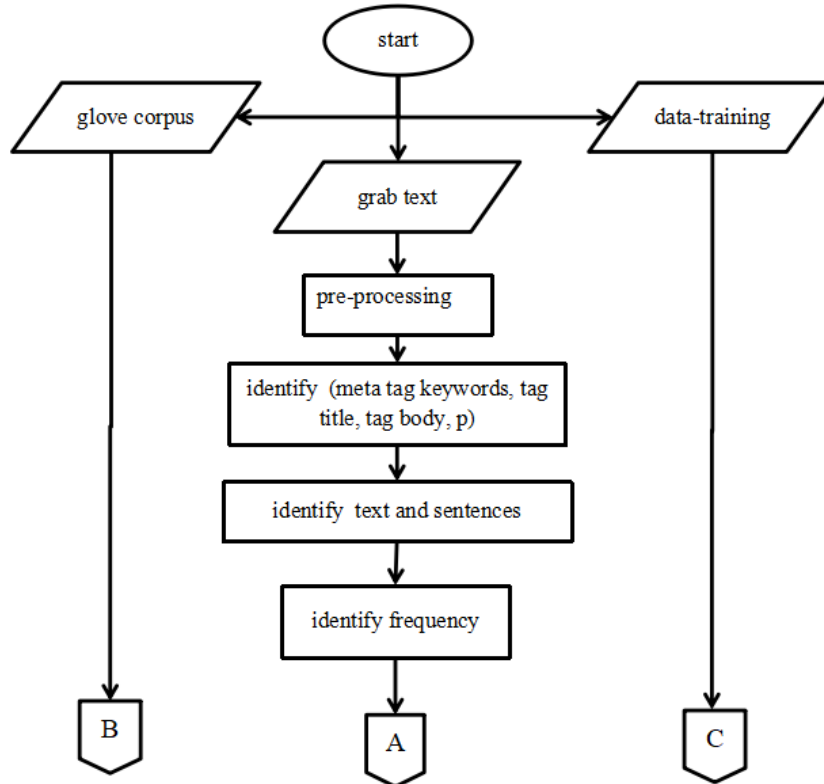
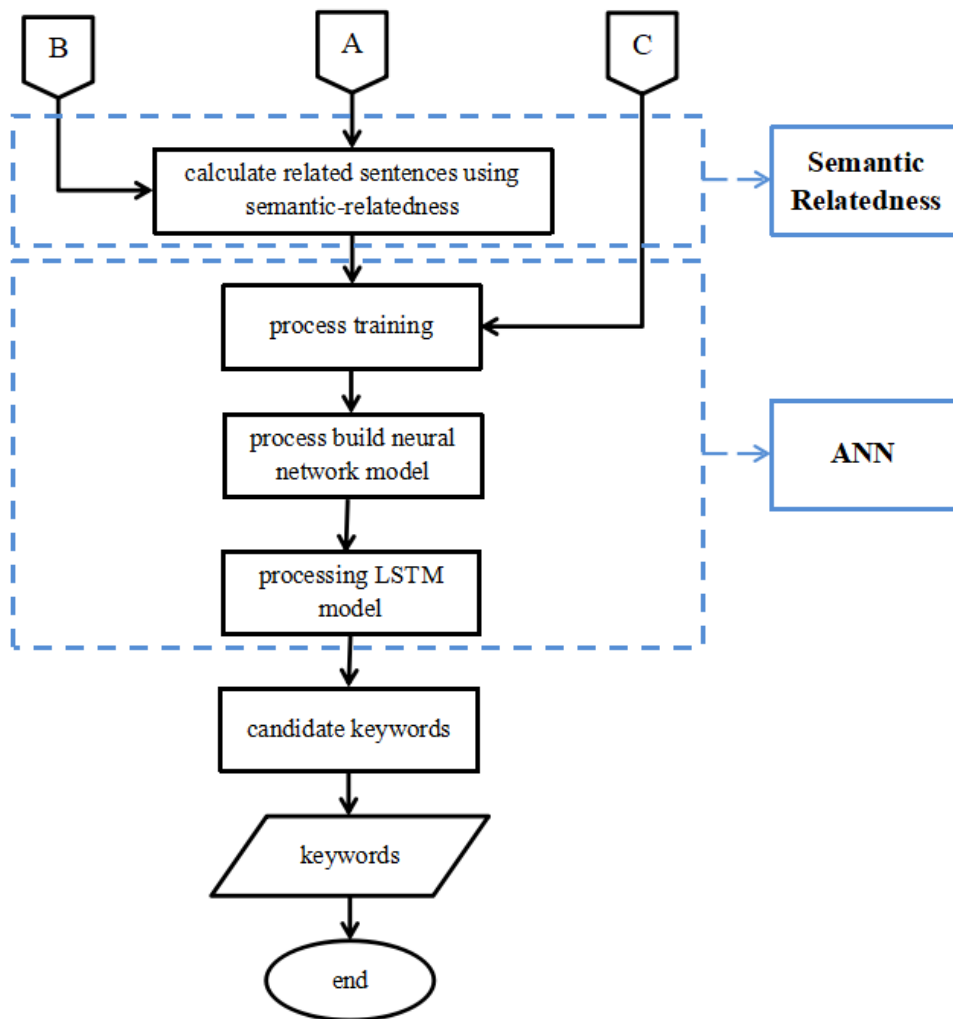


BAB III METODA PENELITIAN

3.1. Tahapan Teknik Diusulkan

Dalam ekstraksi kata kunci otomatis banyak menggunakan beberapa metode, yaitu yang paling banyak digunakan adalah perhitungan dari jumlah dari frekuensi kata atau biasa disebut *statistical approach*, artinya semakin banyak kata tersebut muncul, maka akan dianggap sebagai suatu kata penting, sehingga dari kata - kata tersebut, munculah patokan yang akan digunakan sebagai kata kunci. Akan tetapi metode seperti *statistical approach* masih ada kelemahan, karena pada dasarnya tidak hanya jumlah frekuensi kata saja yang dijadikan patokan tetapi hubungan kata tersebut dengan kata yang lainnya dan banyaknya penggunaan kata kunci yang biasa digunakan pada dokumen – dokumen lainnya yang berhubungan dengan topik.





Gambar 3.1 Tahapan Proses Menghasilkan Kandidat Kata Kunci

Berdasarkan pada gambar 3.1 ada beberapa tahapan proses untuk menghasilkan kandidat kata kunci, berikut penjelasan tahapan prosesnya :

3.1.1 Data Training

Data training adalah data yang digunakan untuk pelatihan pada *machine learning* untuk *LSTM*, yaitu data yang diambil dari Hulth 2003, berisi 1000 *files* data *training*.

3.1.2 *Glove Corpus*

Glove Corpus berisi kata – kata dan frase yang memrepresentasikan nilai vektor dari setiap kata. Pada kasus ini untuk *glove*, data *training* yang digunakan adalah gabungan dari *bag of words*, *n-gram* dan *np-chunks* yaitu sebuah konsep yang diambil dari analisis teks, dengan mempresentasikan dokumen sebagai sebuah kantung informasi penting, tanpa mengurutkan setiap kata ataupun kalimatnya.

3.1.3 *Grabbing Teks Dari Website*

Grabbing adalah teknik mengambil konten atau objek dari situs/*website* lain, lalu dimasukkan ke dalam *website* kita.

3.1.4 *Identify Teks Dengan Meta Tag*

Setelah dapat mengambil teks yang diperlukan, maka selanjutnya adalah melakukan identifikasi pada setiap *meta tag* seperti *meta tag keywords*, *tag title*, *tag body*, *p* dan lain - lain.

3.1.5 *Identifikasi teks dan kalimat dengan menggunakan N-Gram*

Identifikasi teks atau kalimat menggunakan *N-Gram*, sangat diperlukan karena terkadang pada keyword tidak hanya satu kata saja, tetapi terkadang gabungan dari beberapa kata (*phrase*).

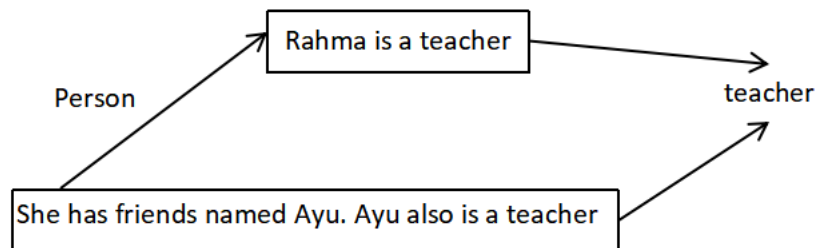
3.1.6 *Frekuensi Teks*

Setelah mendapatkan text atau kalimat yang dari *N-Gram*, maka selanjutnya mengkalkulasikan banyaknya teks yang digunakan pada dari *website* tersebut, karena salah satu ciri *keyword* yaitu banyaknya kata yang muncul.

3.1.7 *Calculate Related Sentences Using Semantic Relatedness*

Keterkaitan semantik dapat dilihat dari banyaknya hubungan seperti *Hypernym*, *Hiponim*, *Holonim*, *Meronim*, *Troponim*, dan Antonim. Hal itu

menunjukkan bahwa hubungan antara kata-kata atau makna kata-kata pada tingkat kata serta makna kalimat dan frasa kunci di tingkat kalimat. Hubungan semantik antara kata-kata akan menjadi fitur utama dalam sistem karena dapat memberikan makna pada kata yang akan membantu mengidentifikasi kata kunci dari dokumen.



Gambar 3.2 Hubungan Semantik

Dalam gambar diatas ada *[relation teacher]* hubungan semantik “*teacher*” disebutkan bahwa Rahma dan Ayu adalah seorang guru. Dalam analisis semantik hubungan antar kata ke kata atau kalimat ke kalimat dapat ditemukan hubungannya. Dengan menggunakan hubungan semantik, kata kunci penting dapat dideteksi dari dokumen atau *website* tertentu.

3.1.8 *Process Training*

Training yang dilakukan pada proses ini yaitu mempelajari dari *dataset* yang disediakan dan nantinya akan membuat sebuah model.

3.1.9 *Process Build Neural Network Model*

Setelah melakukan proses pembelajaran menggunakan dataset yang disediakan, maka selanjutnya adalah proses pembuatan model *neural network* yang dimana menggunakan model LSTM. Model inilah yang nantinya akan menghasilkan informasi.

3.1.10 Processing LSTM Model

Setelah proses pembuatan model selesai, maka selanjutnya adalah proses pemodelan yang dilakukan oleh LSTM. Proses pemodelan LSTM ini akan mengeluarkan *output* untuk memecahkan masalah yang diberikan oleh sistem pada saat testing kedepannya dan akan melakukan prediksi berdasarkan pembelajaran yang telah dilakukan menggunakan *data training*.

3.1.11 Candidate Keyword

Candidate keyword adalah hasil dari ekstraksi kata kunci yang didapat dari gabungan dari *dataset* yaitu *dataset* dari *word vector* yang telah diberi nilai dan *dataset* pelatihan.

3.2. Implementasi Teknik Penggabungan

ANN digambarkan bagaimana membuat *memory* untuk *Neural Network* bersifat *associative*, sehingga didapat algoritma yang biasa digunakan yaitu LSTM. Pada LSTM, untuk memorinya sendiri telah diatur sendiri pada *Hidden State*-nya. Berikut rumus yang digunakan pada implementasi LSTM :

Tabel 3.1 Perhitungan untuk LSTM

<i>Forget Gate</i>	$f = \sigma (x_t U^f + S_{t-1} W^f)$
<i>Input Gate</i>	$i = \sigma (x_t U^i + S_{t-1} W^i)$
<i>Output Gate</i>	$o = \sigma (x_t U^o + S_{t-1} W^o)$
<i>Candidate Layer</i>	$g = \tanh (x_t U^g + S_{t-1} W^g)$
<i>Memory State</i>	$C_t = C_{t-1} \circ f + g \circ i$
<i>Hidden State</i>	$S_t = \tanh (c_t) \circ O$

$S_t =$ *Hidden State*

$C_{t-1} =$ *Memory state* sebelumnya

$C_t =$ *Memory state* saat ini