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

METODOLOGI DAN PERANCANGAN

3.1 Metode Penelitian

Penelitian ini menggunakan metode-metode sebagai berikut.

1. Studi Pustaka

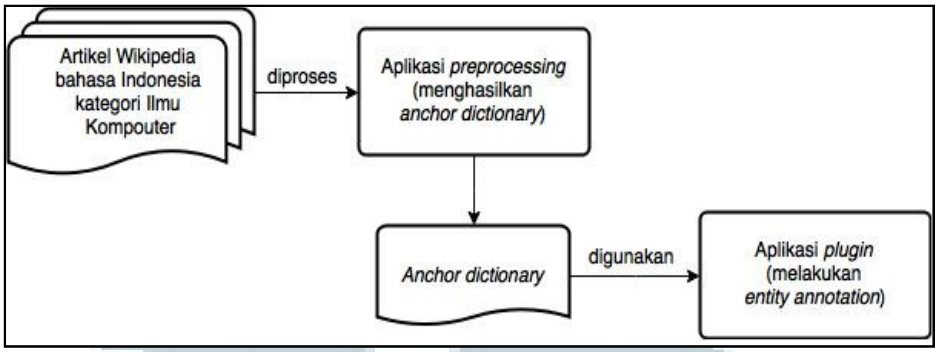
   Studi pustaka dilakukan dengan mencari, membaca, dan mempelajari buku, jurnal, artikel, dan referensi lain yang berkaitan dengan penelitian yang dilakukan.

2. Perancangan Aplikasi

   Pada penelitian ini dibangun dua aplikasi, yaitu aplikasi *preprocessing* dan aplikasi *plugin*. Aplikasi *preprocessing* digunakan untuk mengolah informasi yang ada pada Wikipedia, yaitu artikel Wikipedia bahasa Indonesia dengan kategori Ilmu Komputer untuk menghasilkan *anchor dictionary* yang diperlukan untuk mengimplementasikan teknologi TAGME. Aplikasi *plugin* merupakan *plugin* pada CMS WordPress yang digunakan untuk melakukan *entity annotation*. Proses *entity annotation* tersebut dilakukan dengan menggunakan *anchor dictionary* yang dihasilkan oleh aplikasi *preprocessing*. Dengan demikian, *anchor dictionary* yang dihasilkan oleh aplikasi *preprocessing* digunakan oleh aplikasi *plugin* yang mengimplementasikan teknologi TAGME untuk melakukan *entity annotation*.

   Hubungan antara aplikasi *preprocessing* dan aplikasi *plugin* ditunjukkan pada gambar 3.1. Adapun proses yang dilakukan untuk mendapatkan artikel Wikipedia bahasa Indonesia kategori Ilmu Komputer dijelaskan pada subbab 3.3.

Implementasi Algoritma ..., William Aprilius, FTI UMN, 2015



Gambar 3.1 Hubungan Antara Aplikasi *Preprocessing* dan Aplikasi *Plugin*

Pada tahap perancangan aplikasi ini, dilakukan perancangan secara fungsional, yaitu merancang tahapan yang dilakukan oleh kedua aplikasi tersebut. Rancangan tersebut didokumentasikan dalam bentuk *Data Flow Diagram* (DFD) dan diagram alir (*flowchart*). Perancangan struktur penyimpanan data pada aplikasi *plugin* ditampilkan dalam bentuk *Entity Relationship Diagram* (ERD). Selain itu, pada tahap ini juga dilakukan perancangan antarmuka pengguna dari aplikasi yang dibangun.

3. Penulisan Kode (*Coding*)

   Kode pada aplikasi *preprocessing* ditulis dengan menggunakan bahasa pemrograman Java. Selain itu, bahasa pemrograman (*scripting*) PHP digunakan untuk membangun aplikasi *plugin*.

4. *Testing* dan *Debugging* Aplikasi

   Proses ini bertujuan untuk memastikan aplikasi yang dibangun telah sesuai dengan spesifikasi yang telah ditentukan.

5. Evaluasi Aplikasi

   Pada tahap ini, aplikasi *plugin* yang telah dibangun diimplementasikan pada *Content Management System* (CMS) WordPress di sebuah *web server* lokal. Kemudian, aplikasi dievaluasi sesuai dengan variabel penelitian.

Implementasi Algoritma ..., William Aprilius, FTI UMN, 2015

6. Penulisan Laporan

   Sebuah laporan sebagai dokumentasi dari proses penelitian dan pengembangan aplikasi ditulis secara bertahap, mulai dari studi pustaka hingga penarikan simpulan.

3.2 Variabel Penelitian

Adapun yang menjadi variabel dalam penelitian ini adalah nilai *precision*, *recall*, dan *F-measure* untuk pembobotan seimbang antara *precision* dan *recall*. *Precision* dapat dihitung dengan membandingkan jumlah anotasi benar dan jumlah seluruh anotasi yang dihasilkan oleh aplikasi. *Recall* dapat dihitung dengan membandingkan jumlah anotasi benar dan jumlah seluruh anotasi yang terdapat pada dokumen teks pembanding (*ground truth*). *F-measure* merupakan sebuah pengukuran yang menyatakan *weighted harmonic mean* dari *precision* dan *recall*. Pembobotan seimbang antara *precision* dan *recall* umumnya ditulis sebagai F_1 dan dapat dihitung dengan menggunakan persamaan 3.1 sebagai berikut (Manning, Raghavan, & Schütze, 2009).

$$F_1 = \frac{2PR}{P+R} \qquad \text{... Persamaan 3.1}$$

Pada persamaan 3.1, P merupakan nilai *precision*, R merupakan nilai *recall*, dan F_1 merupakan nilai *F-measure* untuk pembobotan seimbang antara P dan R.

Implementasi Algoritma ..., William Aprilius, FTI UMN, 2015

3.3 Teknik Pengumpulan Data

Pada penelitian ini dilakukan pengunduhan terhadap artikel Wikipedia bahasa Indonesia dengan kategori Ilmu Komputer[2] yang dapat dilakukan dengan fasilitas ekspor halaman[3]. Pada penelitian ini hanya diambil artikel yang berada pada jangkauan dua subkategori dari kategori utama. Pengunduhan dilakukan pada 22 Februari 2015 pukul 15.43 WIB. Berdasarkan *tool* CatScan[4], terdapat 374 artikel Wikipedia yang akan diolah. Daftar judul artikel Wikipedia yang diperoleh tersebut terdapat pada bagian lampiran (Lampiran A).

3.4 Rancangan Aplikasi Preprocessing

Aplikasi *preprocessing* merupakan aplikasi yang digunakan untuk mengolah informasi yang ada pada Wikipedia. Informasi tersebut berbentuk berkas hasil ekspor halaman Wikipedia. Aplikasi ini menghasilkan sebuah *anchor dictionary* dalam bentuk berkas XML. *Anchor dictionary* tersebut kemudian akan digunakan oleh aplikasi *plugin* untuk melakukan *entity annotation*.

3.4.1 Data Flow Diagram

Berikut ini merupakan *Data Flow Diagram* (DFD) dari aplikasi *preprocessing*.

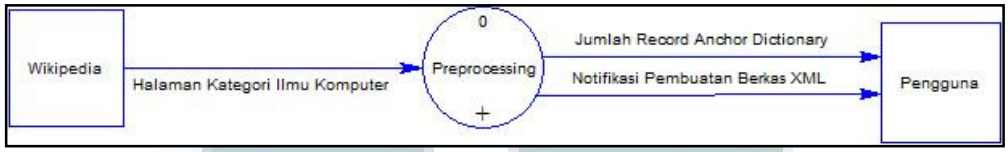
A. Diagram Konteks

Terdapat dua entitas eksternal, yaitu Wikipedia dan Pengguna. Entitas Wikipedia merepresentasikan asal data halaman-halaman Wikipedia yang berada dalam kategori Ilmu Komputer. Entitas Pengguna merepresentasikan pengguna

[2] http://id.wikipedia.org/wiki/Kategori:Ilmu_komputer
[3] http://id.wikipedia.org/wiki/Istimewa:Ekspor_halaman
[4] http://tools.wmflabs.org/catscan2/catscan2.php diakses pada 21 Februari 2015

Implementasi Algoritma ..., William Aprilius, FTI UMN, 2015

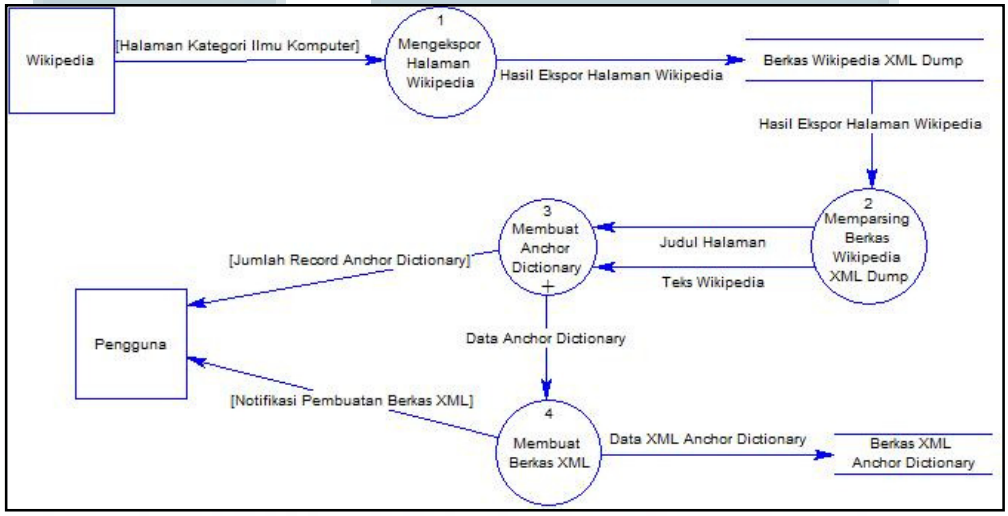
dari aplikasi yang menerima data jumlah *record anchor dictionary* dan notifikasi pembuatan berkas XML. Gambar 3.2 menunjukkan diagram konteks dari aplikasi *preprocessing*.



Gambar 3.2 Diagram Konteks Aplikasi *Preprocessing*

B. Diagram Level 0

Gambar 3.3 menunjukkan diagram *level* 0 yang merupakan detail dari diagram konteks pada gambar 3.2.



Gambar 3.3 Diagram *Level* 0 Aplikasi *Preprocessing*

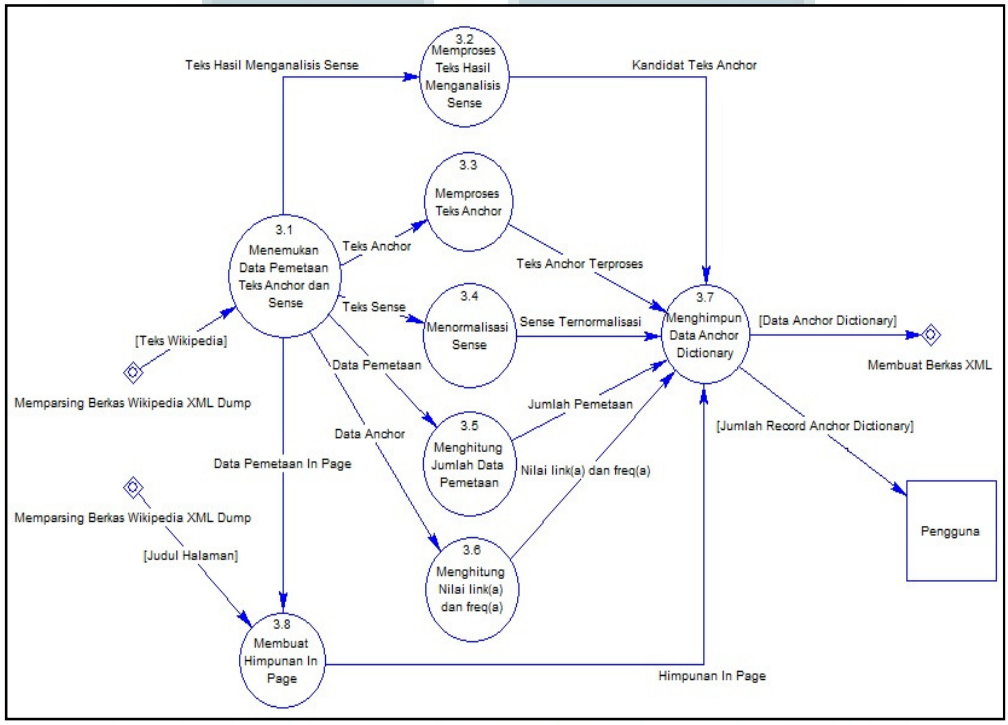
Diagram *level* 0 pada gambar 3.3 menunjukkan aliran data pada aplikasi *preprocessing*. Secara umum, data halaman-halaman Wikipedia yang berada dalam kategori Ilmu Komputer diekspor menggunakan fasilitas ekspor halaman. Hasil ekspor tersebut berbentuk berkas XML (untuk selanjutnya disebut Wikipedia XML *dump*). Kemudian, Wikipedia XML *dump* tersebut di-*parse* sehingga informasi yang terdapat di dalamnya dapat diolah untuk membuat

Implementasi Algoritma ..., William Aprilius, FTI UMN, 2015

anchor dictionary. *Anchor dictionary* yang dihasilkan kemudian disimpan dalam bentuk berkas XML.

C. Diagram Level 1 Proses Membuat Anchor Dictionary

Gambar 3.4 menunjukkan diagram *level* 1 yang merupakan detail dari proses Membuat *Anchor Dictionary* yang terdapat pada diagram *level* 0.



Gambar 3.4 Diagram *Level* 1 Proses Membuat *Anchor Dictionary*

Aliran data pada proses Membuat *Anchor Dictionary* ditunjukkan pada gambar 3.4. Secara umum, proses ini diawali dengan menemukan data pemetaan teks *anchor* dan *sense* yang terdapat dalam teks isi halaman Wikipedia. Kemudian, data pemetaan tersebut diproses untuk mendapatkan beberapa informasi yang merupakan bagian dari *anchor dictionary*, yaitu teks *anchor*, *sense* ternormalisasi, teks hasil menganalisis *sense* sebagai kandidat teks *anchor*, jumlah data pemetaan, nilai $link(a)$ dan $freq(a)$, dan himpunan judul halaman

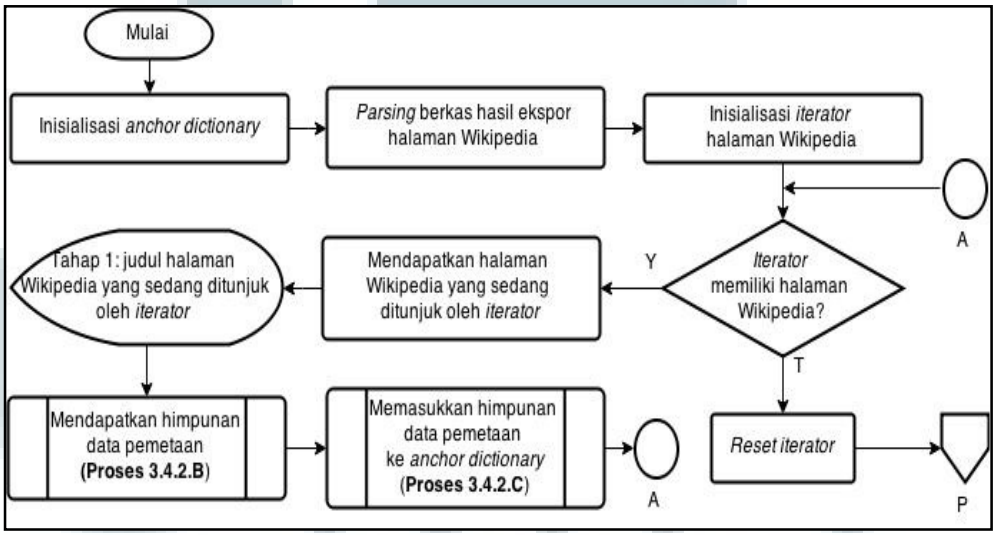
Wikipedia yang mengandung data pemetaan tersebut (untuk selanjutnya disebut himpunan *in page*). Seluruh informasi tersebut dihimpun untuk membuat *anchor dictionary*.

3.4.2 Diagram Alir

Diagram alir dari aplikasi *preprocessing* terdiri dari satu diagram alir utama dan empat diagram alir turunan. Diagram alir utama ditampilkan pada subbab 3.4.2.A dan keempat diagram turunan ditampilkan pada subbab 3.4.2.B hingga 3.4.2.E. Diagram alir turunan menggambarkan detail dari proses yang terdapat dalam diagram alir utama.

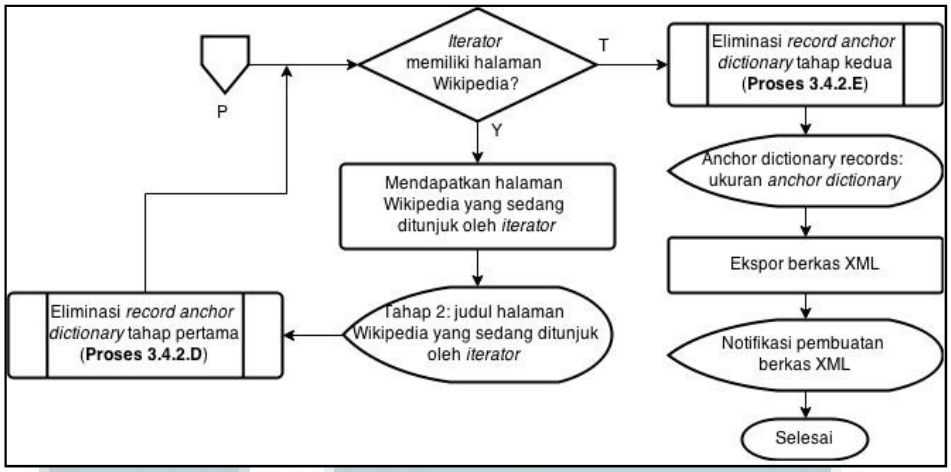
A. Proses Utama

Gambar 3.5 dan 3.6 menunjukkan diagram alir dari proses utama aplikasi *preprocessing*. Proses *parsing* dan iterasi dari Wikipedia XML *dump* dilakukan dengan menggunakan WikiXMLJ[5].



Gambar 3.5 Diagram Alir Proses Utama Aplikasi *Preprocessing* Bagian 1

[5] WikiXMLJ r43 di-*download* dari https://code.google.com/p/wikixmlj/ pada 22 Februari 2015



Gambar 3.6 Diagram Alir Proses Utama Aplikasi *Preprocessing* Bagian 2

B. Proses Mendapatkan Himpunan Data Pemetaan

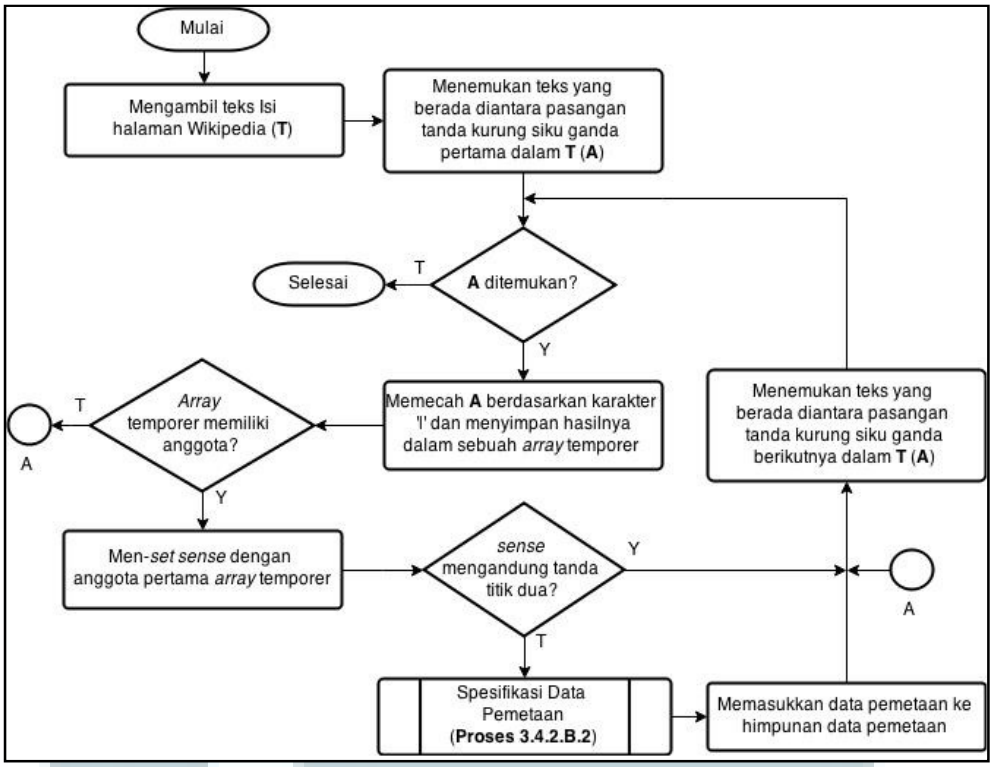
Diagram alir pada proses ini terdiri dari satu diagram alir utama yang ditampilkan pada subbab 3.4.2.B.1 dan satu diagram alir turunan yang ditampilkan pada subbab 3.4.2.B.2.

B.1 Proses Utama Mendapatkan Himpunan Data Pemetaan

Data pemetaan merupakan *wikilink* atau *internal link*, yaitu *link* yang menghubungkan satu halaman Wikipedia ke halaman Wikipedia yang lain. *Link* yang terdapat dalam suatu halaman Wikipedia merupakan teks yang berada diantara tanda kurung siku ganda. Dalam hal ini, teks yang muncul pada halaman asal sama dengan judul halaman Wikipedia yang dituju. Jika teks dalam pasangan tanda kurung siku ganda dipisahkan oleh simbol *pipe* ('|'), teks sebelum simbol *pipe* merupakan judul halaman Wikipedia yang dituju dan teks setelah simbol *pipe* merupakan teks yang muncul pada halaman asal (Wikipedia, 2015a).

Proses ini bertujuan untuk mendapatkan himpunan data pemetaan tersebut pada halaman Wikipedia yang sedang diproses dalam iterasi. Gambar 3.7 menunjukkan diagram alir dari proses ini.

Implementasi Algoritma ..., William Aprilius, FTI UMN, 2015

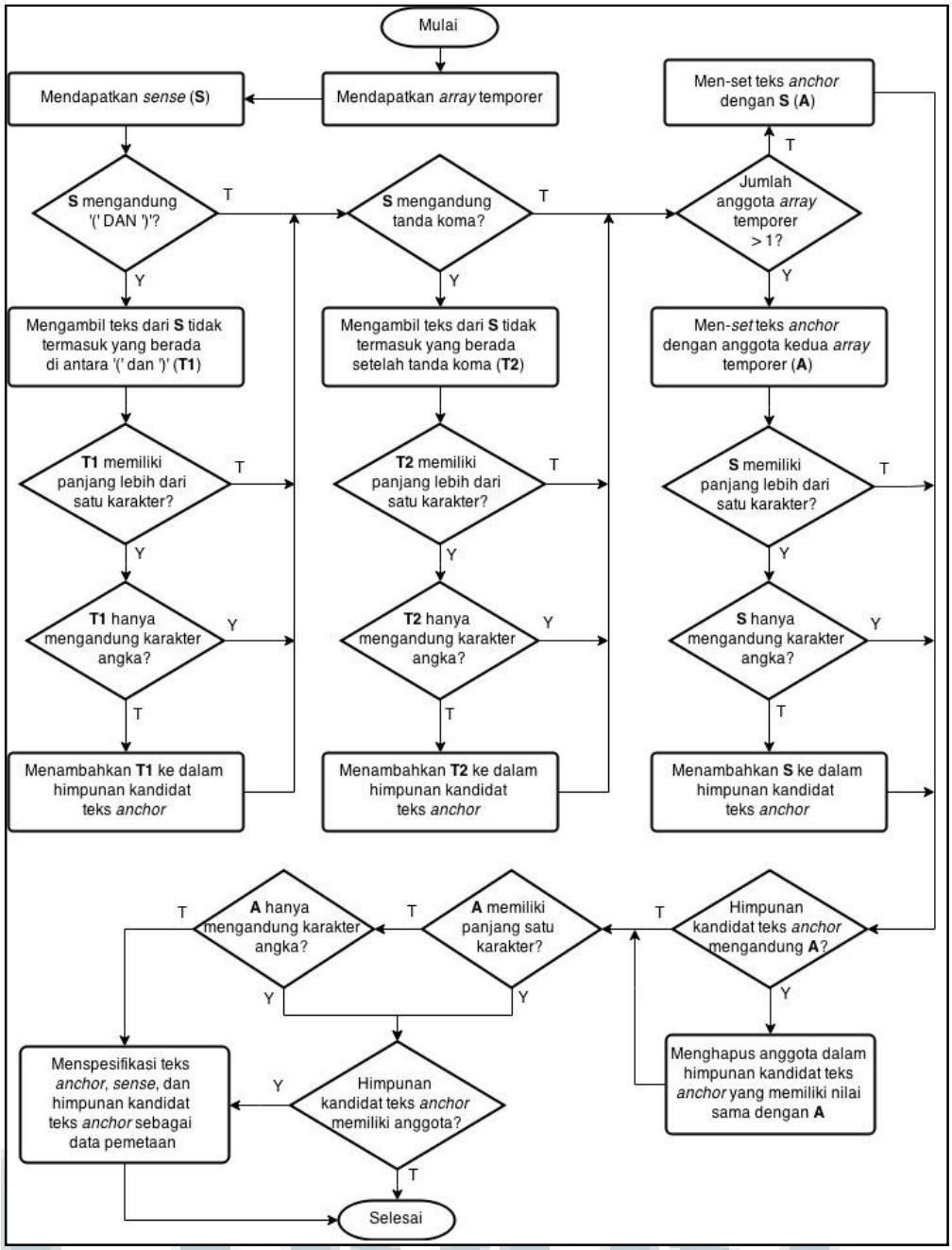


Gambar 3.7 Diagram Alir Proses Utama Mendapatkan Himpunan Data Pemetaan

B.2 Proses Spesifikasi Data Pemetaan

Tujuan dari proses ini adalah untuk menspesifikasi data pemetaan, yaitu teks *anchor* dan *sense*. Selain itu, dilakukan analisis terhadap *sense* untuk menghasilkan himpunan kandidat teks *anchor*. Kemudian, himpunan kandidat teks *anchor* tersebut disatukan dengan data pemetaan. Proses ini juga memastikan teks *anchor* dan kandidat teks *anchor* memiliki panjang lebih dari satu karakter dan tidak hanya mengandung karakter angka. Gambar 3.8 menunjukkan diagram alir dari proses ini.

Implementasi Algoritma ..., William Aprilius, FTI UMN, 2015



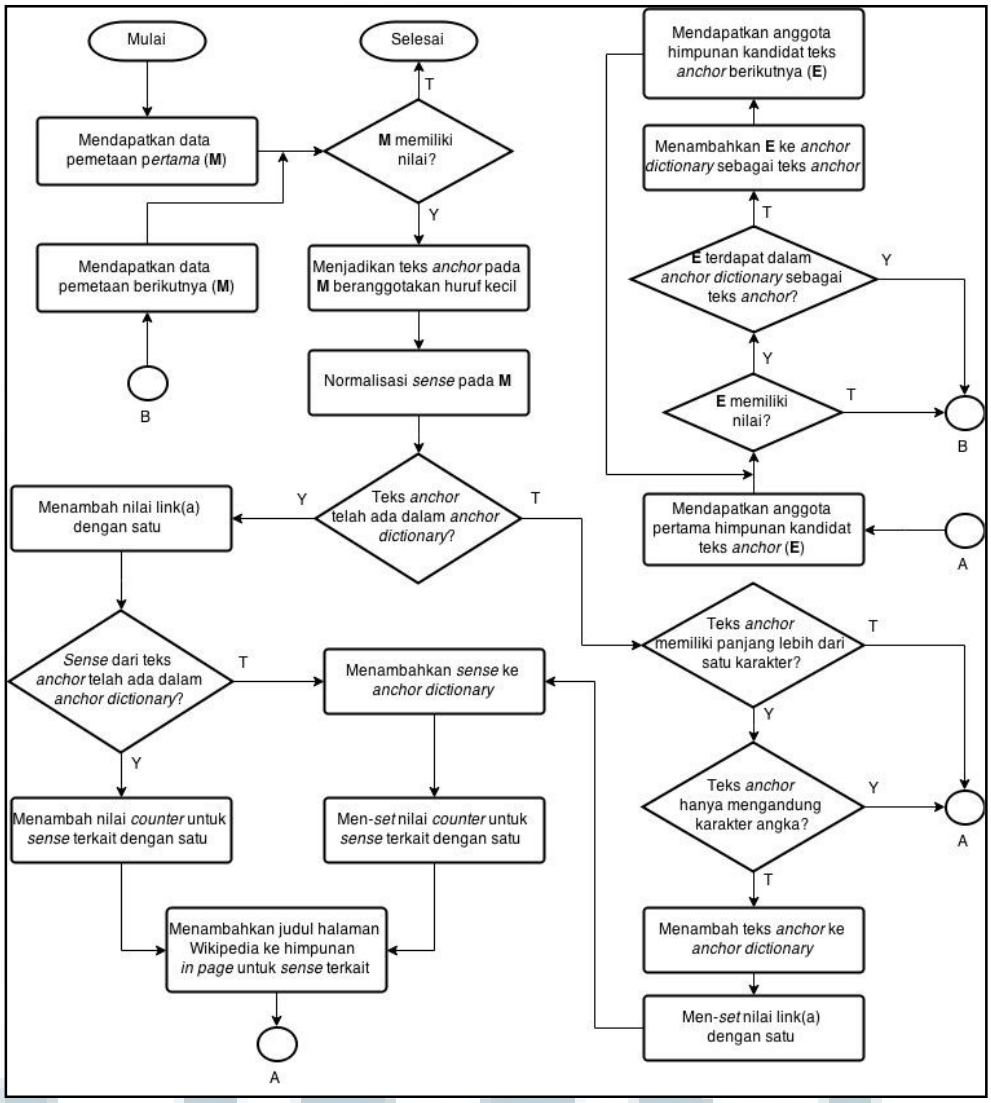
Gambar 3.8 Diagram Alir Proses Spesifikasi Data Pemetaan

C. Proses Memasukkan Himpunan Data Pemetaan ke Anchor Dictionary

Proses ini memasukkan himpunan data pemetaan yang telah didapatkan ke *anchor dictionary*. Selain itu, nilai $link(a)$, jumlah data pemetaan, dan himpunan

Implementasi Algoritma ..., William Aprilius, FTI UMN, 2015

in page yang berkaitan dengan suatu data pemetaan juga ditentukan. Berikut merupakan gambar diagram alir dari proses ini.



Gambar 3.9 Diagram Alir Proses Memasukkan Himpunan Data Pemetaan ke *Anchor Dictionary*

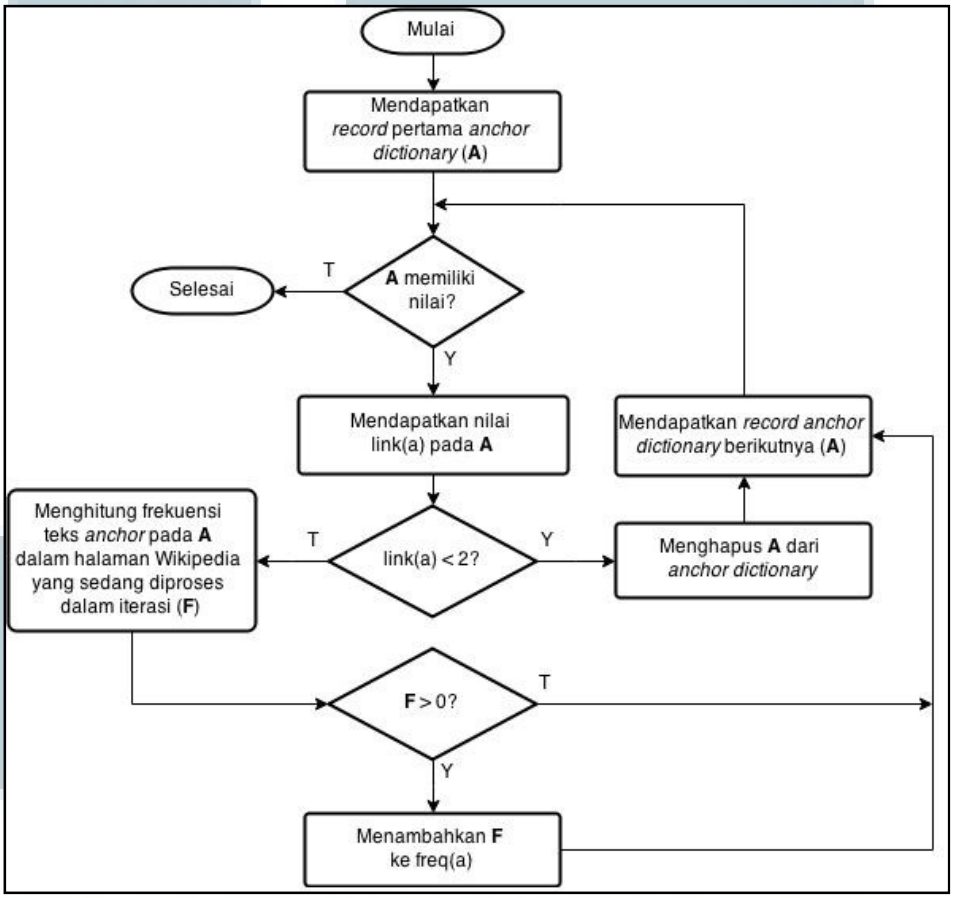
Proses normalisasi *sense*, yaitu judul halaman Wikipedia yang dituju oleh *link*, dilakukan dengan mengapitalisasi huruf pertama dari *sense* dan mengubah karakter spasi menjadi karakter garis bawah (*underscore*). Hal ini dilakukan karena pada Wikipedia, *link* yang diawali dengan sebuah huruf kecil diperlakukan

Implementasi Algoritma ..., William Aprilius, FTI UMN, 2015

sama dengan *link* yang diawali dengan sebuah huruf kapital. Selain itu, karakter spasi dan garis bawah juga diperlakukan sama (Wikipedia, 2015b).

D. Proses Eliminasi Record Anchor Dictionary Tahap Pertama

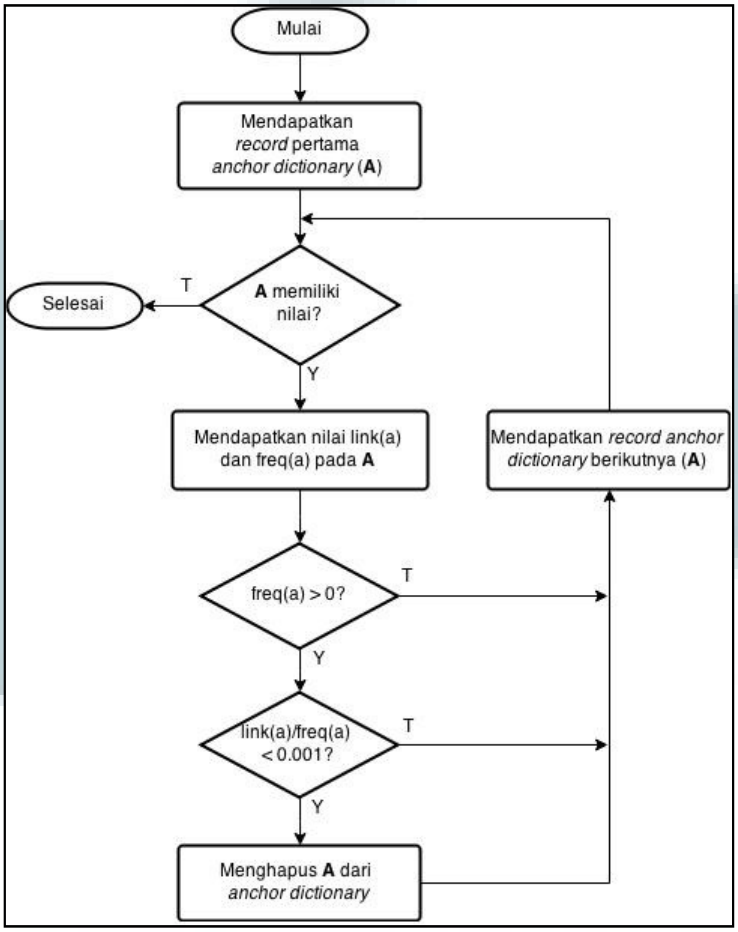
Tujuan dari proses ini adalah mengeliminasi *record anchor dictionary* yang memiliki nilai $link(a) < 2$. Selain itu, proses ini menghitung frekuensi kemunculan teks *anchor* pada halaman Wikipedia yang sedang diproses dalam iterasi, kemudian mengakumulasikannya dengan nilai $freq(a)$ pada *record anchor dictionary* yang bersesuaian dengan teks *anchor* tersebut. Gambar 3.10 menunjukkan diagram alir dari proses ini.



Gambar 3.10 Diagram Alir Proses Eliminasi *Record Anchor Dictionary* Tahap Pertama

E. Proses Eliminasi Record Anchor Dictionary Tahap Kedua

Proses ini mengeliminasi *record anchor dictionary* yang memiliki nilai $lp(a) < 0.1\%$. Berikut merupakan gambar diagram alir dari proses ini.



Gambar 3.11 Diagram Alir Proses Eliminasi *Record Anchor Dictionary* Tahap Kedua

3.5 Rancangan Aplikasi Plugin

Aplikasi *plugin* melakukan *entity annotation* dengan menggunakan *anchor dictionary* yang dihasilkan oleh aplikasi *preprocessing*. Proses *entity annotation* dilakukan dengan memberikan saran *tag* kepada pengguna berdasarkan teks masukan yang ditulis dan membuat *hyperlink* pada kata atau frasa dalam teks yang ditulis tersebut ke halaman Wikipedia berdasarkan saran *tag* yang dipilih

Implementasi Algoritma ..., William Aprilius, FTI UMN, 2015

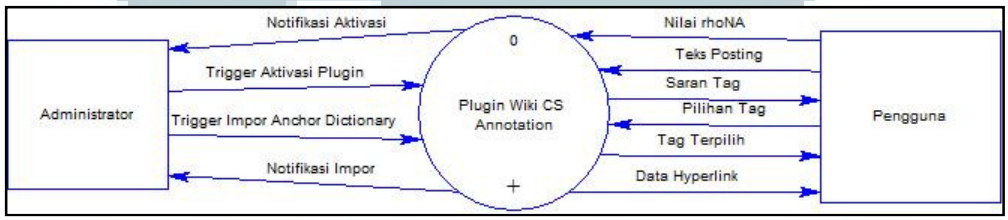
pengguna. Adapun nama dari aplikasi *plugin* yang dibangun adalah Wiki CS Annotation.

3.5.1 Data Flow Diagram

Berikut ini merupakan *Data Flow Diagram* (DFD) dari aplikasi *plugin*.

A. Diagram Konteks

Terdapat dua entitas eksternal, yaitu Administrator dan Pengguna. Entitas Administrator memberikan data berupa *trigger* proses aktivasi *plugin* dan impor *anchor dictionary*. Data yang diterima Administrator berupa notifikasi dari kedua proses tersebut. Entitas Pengguna memberikan beberapa data, yaitu nilai ρNA, teks yang ditulis, dan pilihan saran *tag*. Data yang diterima entitas Pengguna berupa saran *tag*, *tag* yang telah dipilih, dan data *hyperlink*. Gambar 3.12 menunjukkan diagram konteks dari aplikasi *plugin.*



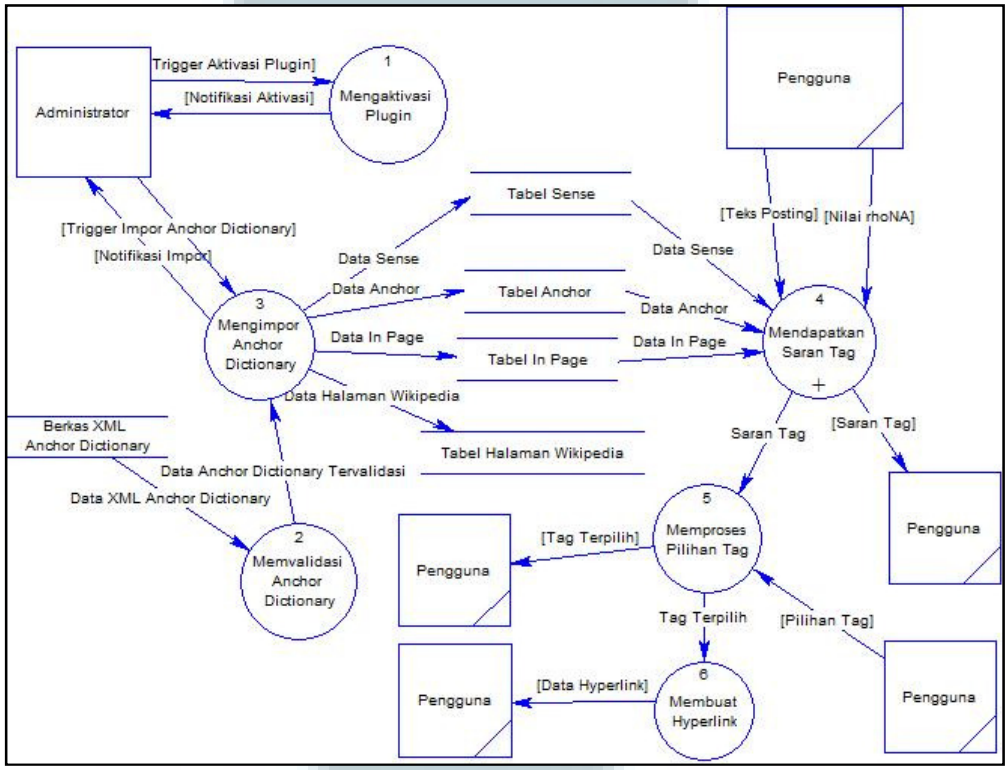
Gambar 3.12 Diagram Konteks Aplikasi *Plugin*

B. Diagram Level 0

Diagram *level* 0 pada gambar 3.13 menunjukkan aliran data pada aplikasi *plugin*. Secara umum, hal ini dapat dijelaskan sebagai berikut. Administrator mengaktivasi *plugin* dan mengimpor *anchor dictionary*. Data *anchor dictionary* berasal dari berkas XML yang dihasilkan oleh aplikasi *preprocessing*. Kemudian, *data anchor dictionary* tersebut disimpan dalam empat tabel. Data yang berada dalam keempat tabel dan relasinya dijelaskan pada subbab 3.5.4. Ketika Pengguna

Implementasi Algoritma ..., William Aprilius, FTI UMN, 2015

ingin mendapatkan saran *tag*, data *anchor dictionary* tersebut digunakan oleh *plugin* untuk menghasilkan saran *tag*. Kemudian, saran *tag* yang dipilih Pengguna akan diproses untuk membuat *hyperlink*. Berikut gambar diagram *level* 0 yang merupakan detail dari diagram konteks pada gambar 3.12.

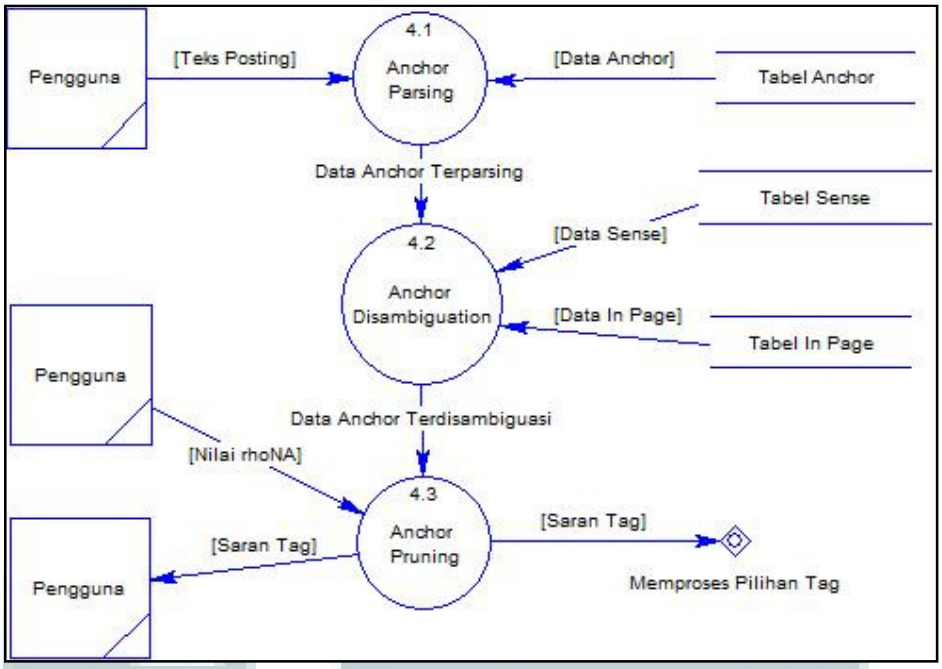


Gambar 3.13 Diagram Level 0 Aplikasi *Plugin*

C. Diagram Level 1 Proses Mendapatkan Saran Tag

Proses Mendapatkan Saran *Tag* memiliki tiga subproses, yaitu *Anchor Parsing*, *Anchor Disambiguation*, dan *Anchor Pruning*. Ketiga proses tersebut merupakan tahapan yang terdapat dalam teknologi TAGME. Gambar 3.14 menunjukkan diagram *level* 1 yang merupakan detail dari proses Mendapatkan Saran *Tag* yang terdapat pada diagram *level* 0.

Implementasi Algoritma ..., William Aprilius, FTI UMN, 2015



Gambar 3.14 Diagram Level 1 Proses Mendapatkan Saran *Tag*

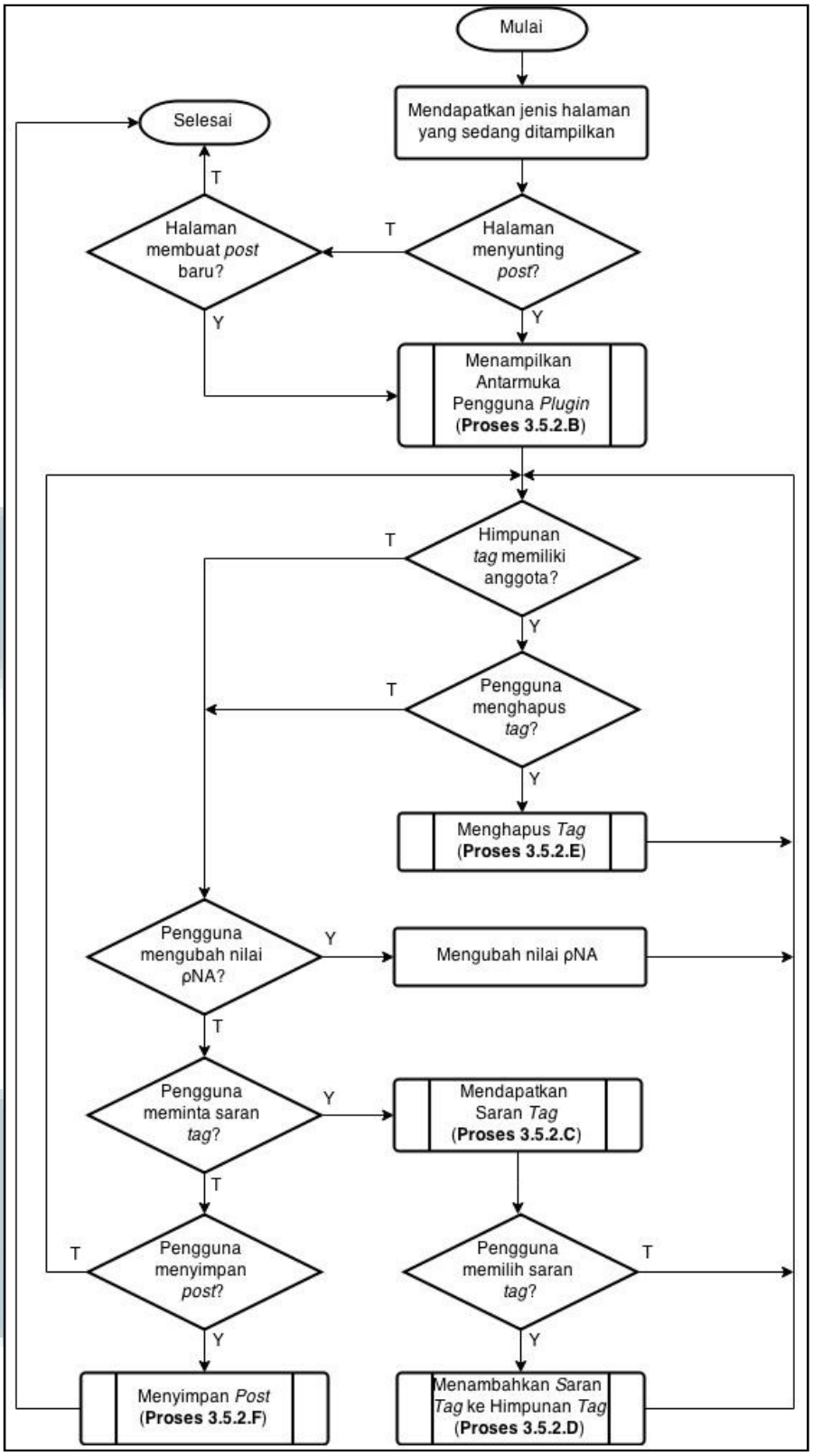
3.5.2 Diagram Alir Pengguna Plugin

Diagram alir untuk pengguna aplikasi *plugin* terdiri dari satu diagram alir utama dan lima diagram alir turunan. Diagram alir utama ditampilkan pada subbab 3.5.2.A dan kelima diagram turunan ditampilkan pada subbab 3.5.2.B hingga 3.5.2.F. Diagram alir turunan menggambarkan detail dari proses yang terdapat dalam diagram alir utama.

A. Proses Utama

Antarmuka pengguna *plugin* ditampilkan pada halaman menyunting *post* atau membuat *post* baru. Aplikasi *plugin* memungkinkan pengguna untuk meminta saran *tag*, memilih saran *tag* menjadi *tag* dari *post*, menghapus *tag* yang telah dipilih, dan mengubah nilai ρNA. Gambar 3.15 menunjukkan diagram alir proses utama pengguna aplikasi *plugin*.

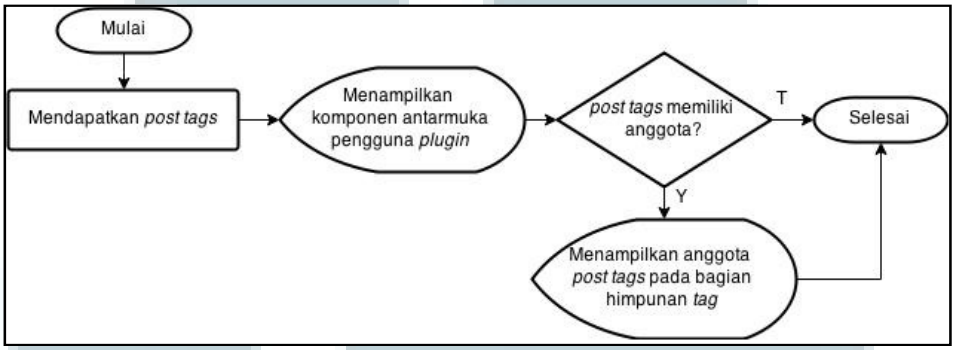
Implementasi Algoritma ..., William Aprilius, FTI UMN, 2015



Gambar 3.15 Diagram Alir Proses Utama Pengguna Aplikasi *Plugin*

B. Proses Menampilkan Antarmuka Pengguna Plugin

Proses ini bertujuan untuk menampilkan komponen antarmuka pengguna aplikasi *plugin*. Jika *post* telah memiliki *tag*, *tag* tersebut juga ditampilkan pada antarmuka pengguna aplikasi *plugin*. Gambar 3.16 menunjukkan diagram alir dari proses ini.



Gambar 3.16 Diagram Alir Proses Menampilkan Antarmuka Pengguna *Plugin*

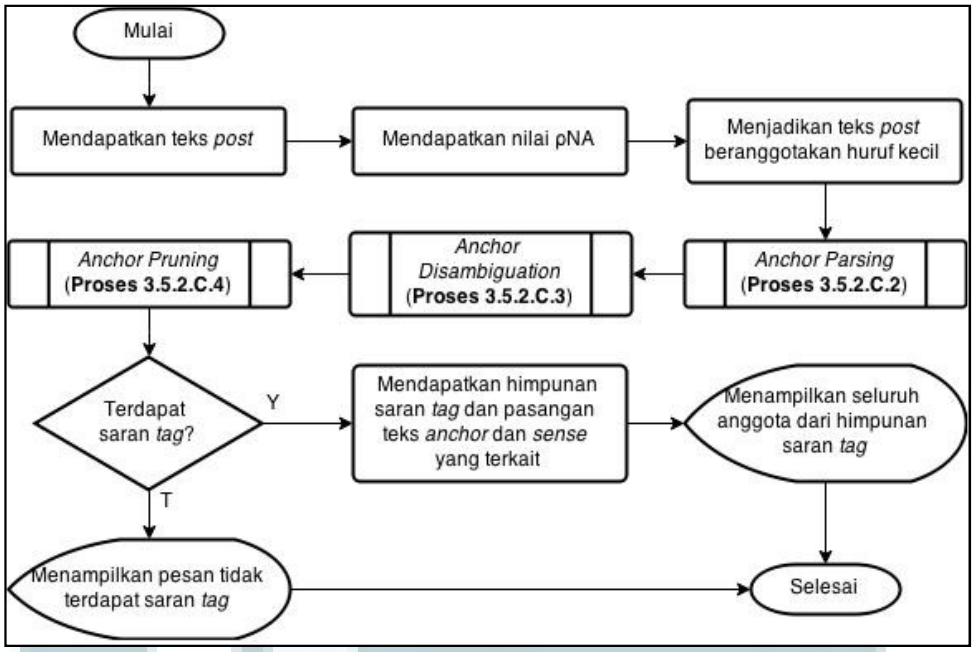
C. Proses Mendapatkan Saran Tag

Diagram alir pada proses ini terdiri dari satu diagram alir utama yang ditampilkan pada subbab 3.5.2.C.1 dan tiga diagram alir turunan yang dijelaskan pada subbab 3.5.2.C.2 hingga 3.5.2.C.4. Diagram alir turunan menggambarkan detail dari proses yang terdapat dalam diagram alir utama.

C.1 Proses Utama Mendapatkan Saran Tag

Proses ini dijalankan saat pengguna ingin mendapatkan saran *tag*. Gambar 3.17 menunjukkan diagram alir dari proses ini.

Implementasi Algoritma ..., William Aprilius, FTI UMN, 2015



Gambar 3.17 Diagram Alir Proses Utama Mendapatkan Saran *Tag*

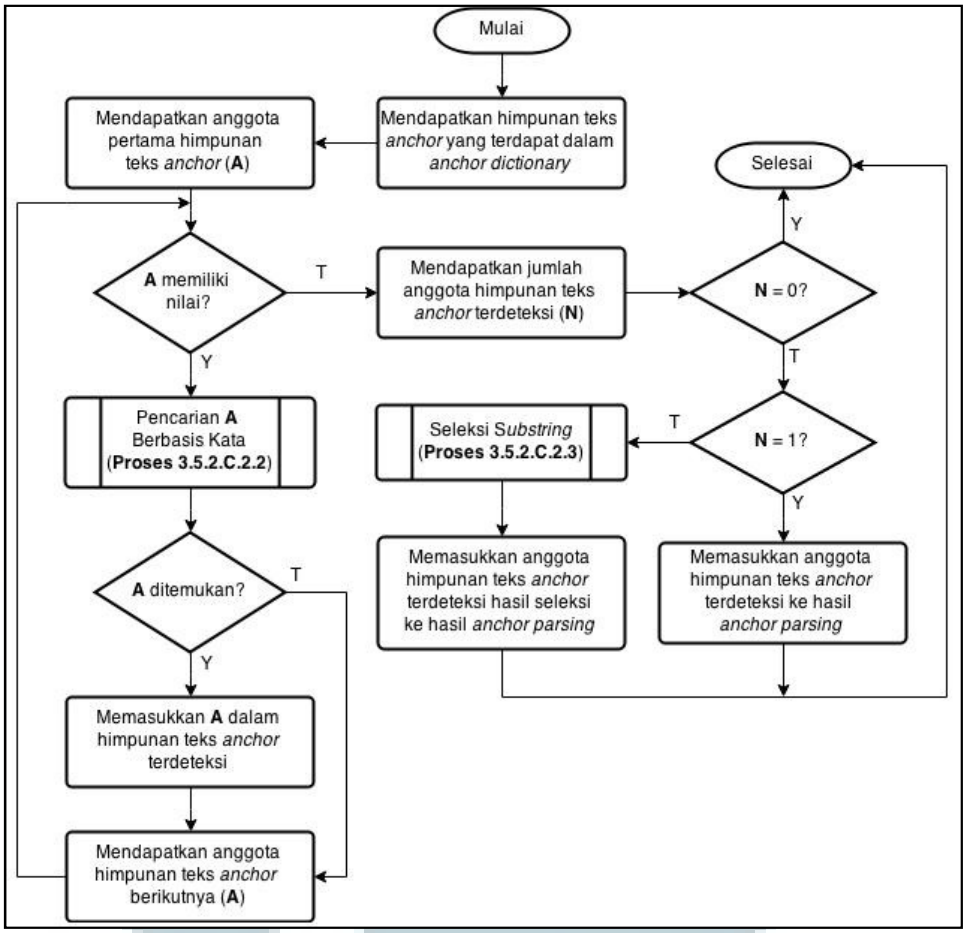
C.2 Proses Anchor Parsing

Diagram alir pada proses ini terdiri dari satu diagram alir utama yang ditampilkan pada subbab 3.5.2.C.2.1 dan empat diagram alir turunan yang dijelaskan pada subbab 3.5.2.C.2.2 hingga 3.5.2.C.2.5. Diagram alir turunan menggambarkan detail dari proses yang terdapat dalam diagram alir utama atau diagram alir turunan lain.

C.2.1 Proses Utama Anchor Parsing

Proses ini melakukan *anchor parsing* seperti yang dilakukan pada teknologi TAGME, yaitu mencari keberadaan teks *anchor* pada *anchor dictionary* dalam teks *post*. Selain itu, proses ini juga melakukan seleksi jika ditemukan lebih dari satu teks *anchor*, yang mana salah satu dari teks *anchor* tersebut merupakan *substring* berbasis kata dari teks *anchor* lainnya. Gambar 3.18 menunjukkan diagram alir dari proses utama *anchor parsing*.

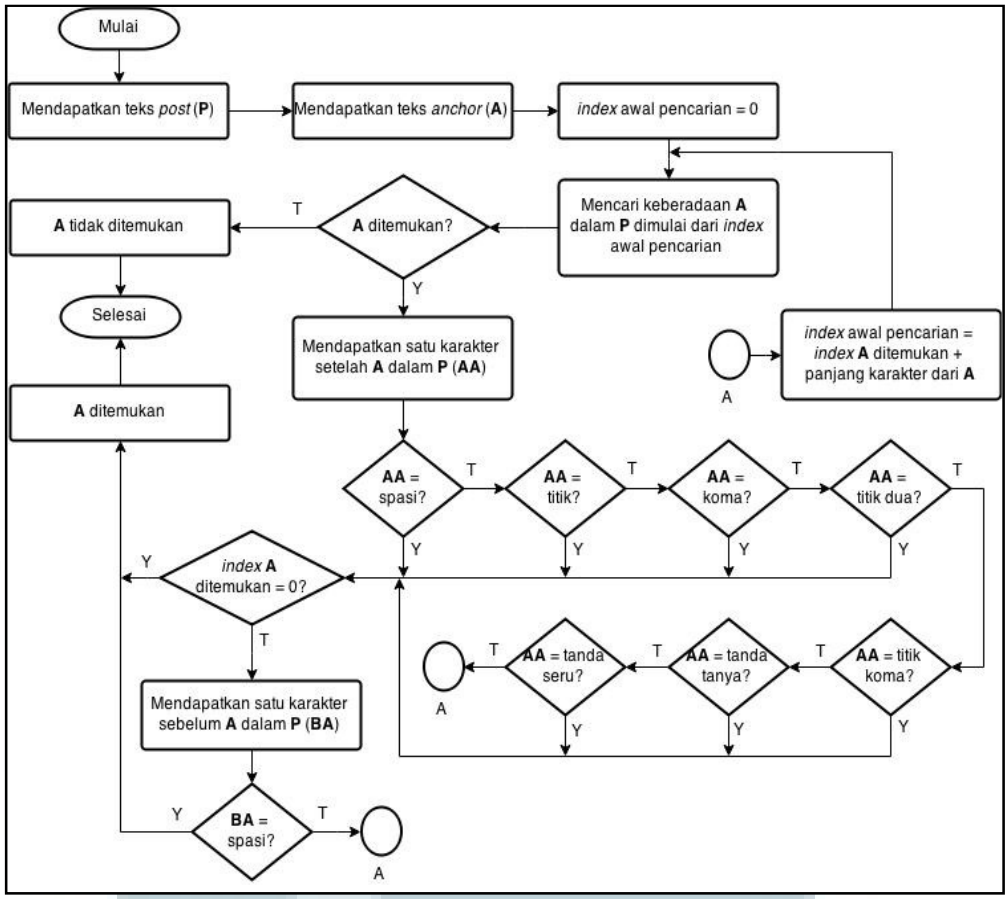
Implementasi Algoritma ..., William Aprilius, FTI UMN, 2015



Gambar 3.18 Diagram Alir Proses Utama *Anchor Parsing*

C.2.2 Proses Mencari Teks Anchor Berbasis Kata

Proses ini bertujuan untuk menemukan keberadaan teks *anchor* dalam teks *post* secara kata per kata. Hal ini dapat dijelaskan sebagai berikut. Jika suatu teks *anchor* ditemukan dalam teks *post*, satu karakter setelah teks *anchor* tersebut di dalam teks *post* harus karakter spasi, titik, koma, titik dua, titik koma, tanda tanya, atau tanda seru. Selain itu, jika teks *anchor* ditemukan bukan di awal teks *post*, satu karakter sebelum teks *anchor* tersebut di dalam teks *post* harus karakter spasi. Gambar 3.19 menunjukkan diagram alir dari proses ini.

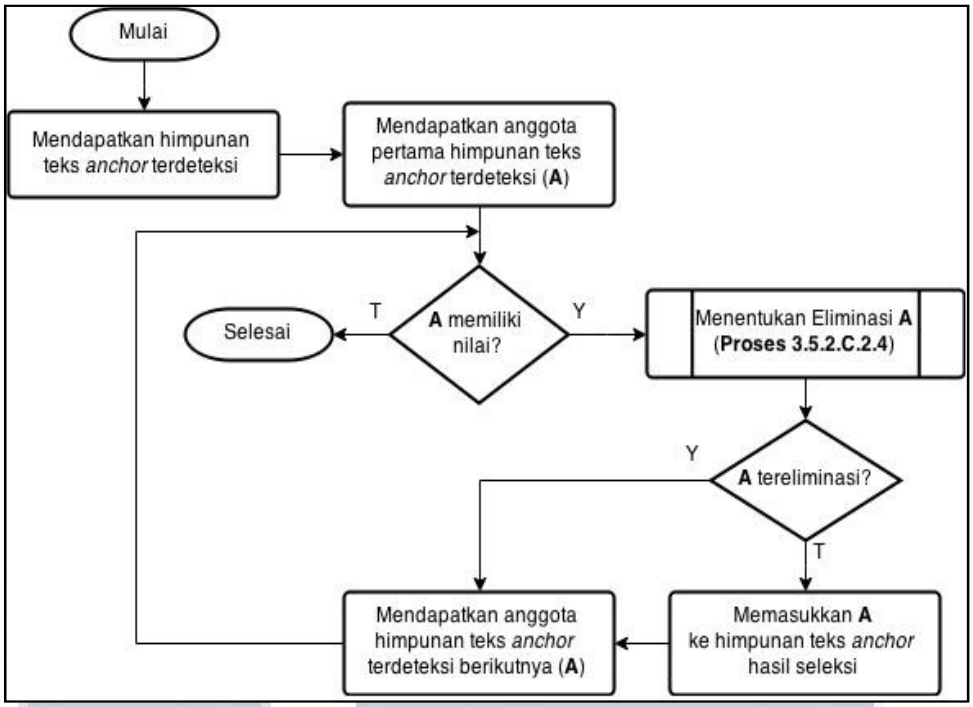


Gambar 3.19 Diagram Alir Proses Mencari Teks *Anchor* Berbasis Kata

C.2.3 Proses Seleksi Substring

Tujuan dari proses ini adalah untuk menyeleksi teks *anchor* yang ditemukan dalam teks *post* jika teks *anchor* tersebut merupakan *substring* berbasis kata dari teks *anchor* lain yang juga ditemukan dalam teks *post*. Gambar 3.20 merupakan diagram alir yang menunjukkan kerangka iterasi pada proses penyeleksian tersebut.

Implementasi Algoritma ..., William Aprilius, FTI UMN, 2015

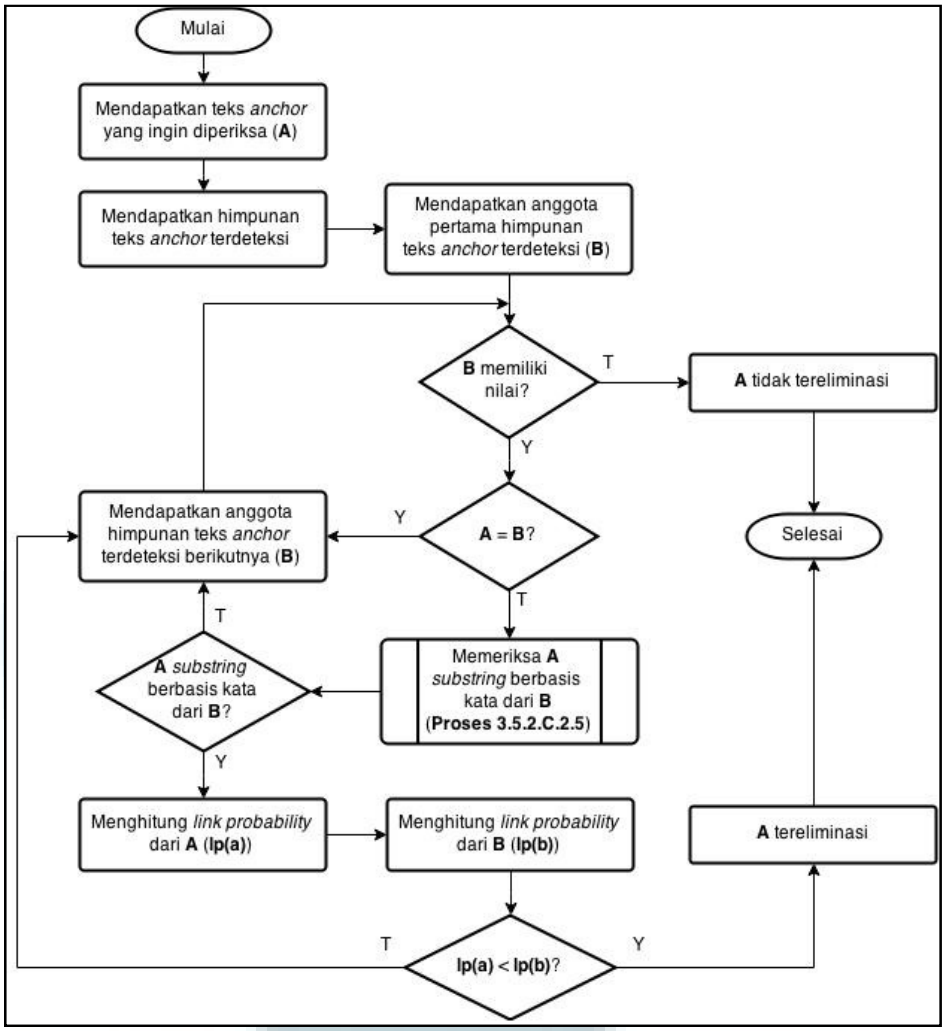


Gambar 3.20 Diagram Alir Proses Seleksi *Substring*

C.2.4 Proses Menentukan Eliminasi Teks Anchor

Proses ini memeriksa apakah suatu teks *anchor* yang ditemukan dalam teks *post* (disebut a_1) adalah *substring* berbasis kata dari teks *anchor* lain yang juga ditemukan dalam teks *post* (disebut a_2). Apabila a_1 merupakan *substring* berbasis kata dari a_2, jika $lp(a_1) < lp(a_2)$, a_1 akan tereliminasi. Kondisi lainnya tidak mengeliminasi a_1. Gambar 3.21 menunjukkan diagram alir dari proses ini.

Implementasi Algoritma ..., William Aprilius, FTI UMN, 2015

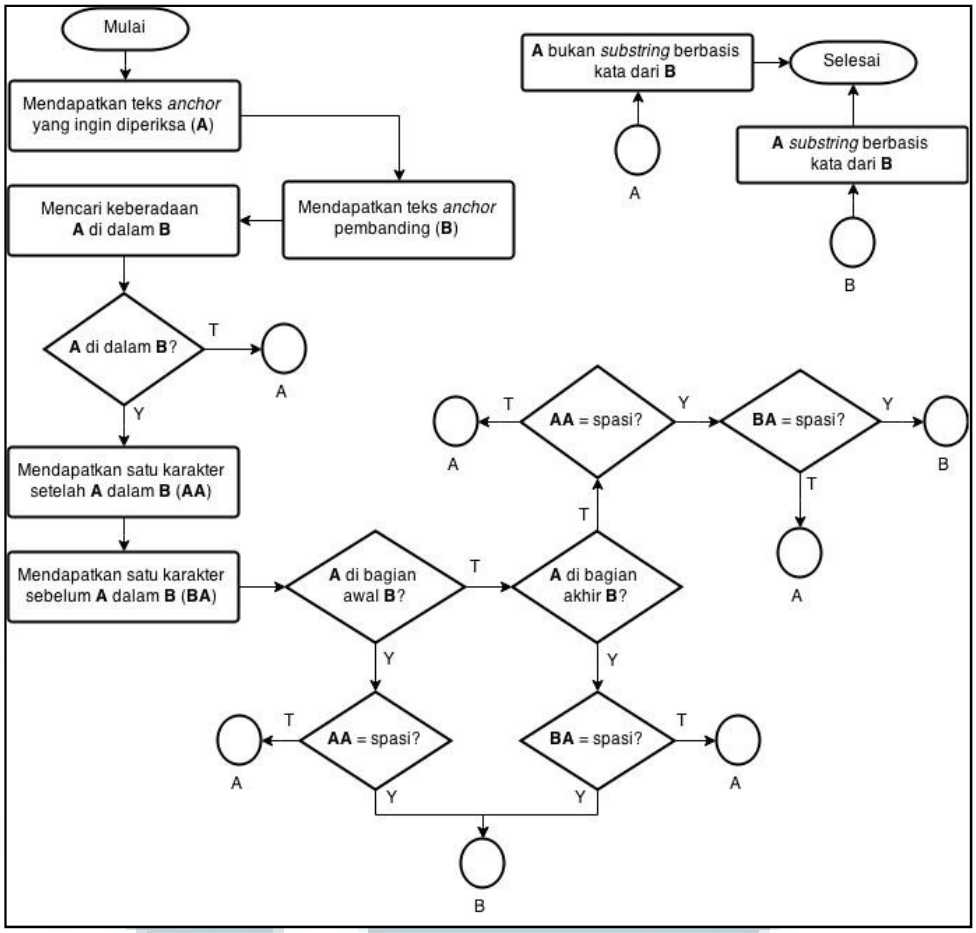


Gambar 3.21 Diagram Alir Proses Menentukan Eliminasi Teks *Anchor*

C.2.5 Proses Memeriksa Substring Berbasis Kata

Proses ini menerima dua teks masukan, kemudian memeriksa apakah teks pertama merupakan *substring* dari teks kedua secara kata per kata. Proses ini pada dasarnya sama dengan proses menemukan teks pertama dalam teks kedua secara kata per kata, seperti dijelaskan pada subbab 3.5.2.C.2.2. Namun, karakter yang digunakan hanya spasi sehingga satu karakter sebelum dan/atau sesudah teks pertama di dalam teks kedua harus berupa spasi. Gambar 3.22 menunjukkan diagram alir dari proses ini.

Implementasi Algoritma ..., William Aprilius, FTI UMN, 2015



Gambar 3.22 Diagram Alir Proses Memeriksa *Substring* Berbasis Kata

C.3 Proses Anchor Disambiguation

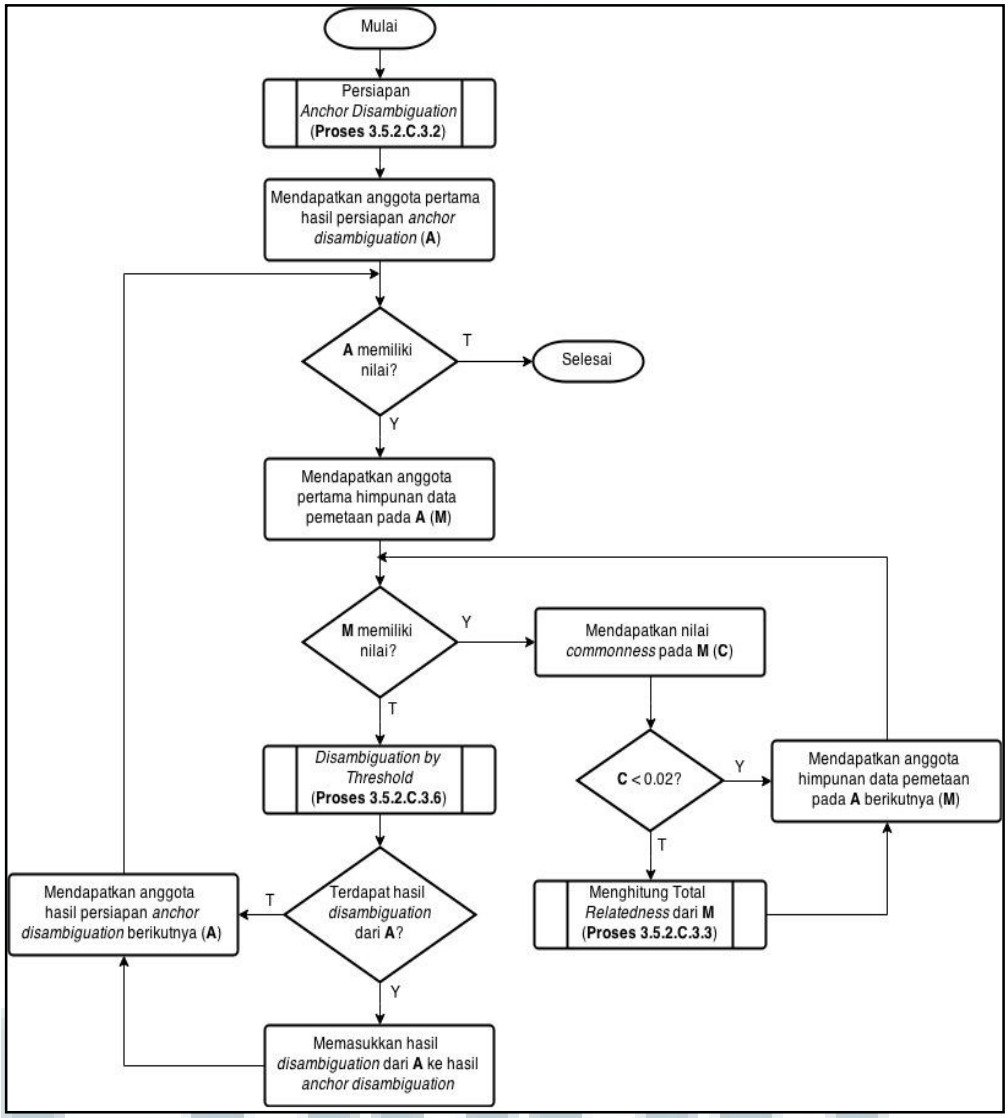
Diagram alir pada proses ini terdiri dari satu diagram alir utama yang ditampilkan pada subbab 3.5.2.C.3.1 dan lima diagram alir turunan yang dijelaskan pada subbab 3.5.2.C.3.2 hingga 3.5.2.C.3.6. Diagram alir turunan menggambarkan detail dari proses yang terdapat dalam diagram alir utama atau diagram alir turunan lain.

C.3.1 Proses Utama Anchor Disambiguation

Proses ini melakukan *anchor disambiguation* seperti yang dilakukan pada teknologi TAGME. Jika suatu *anchor* memiliki lebih dari satu *sense*,

Implementasi Algoritma ..., William Aprilius, FTI UMN, 2015

disambiguation merupakan proses memilih satu dari beberapa *sense* tersebut. Berikut merupakan gambar diagram alir dari proses utama *anchor disambiguation*.

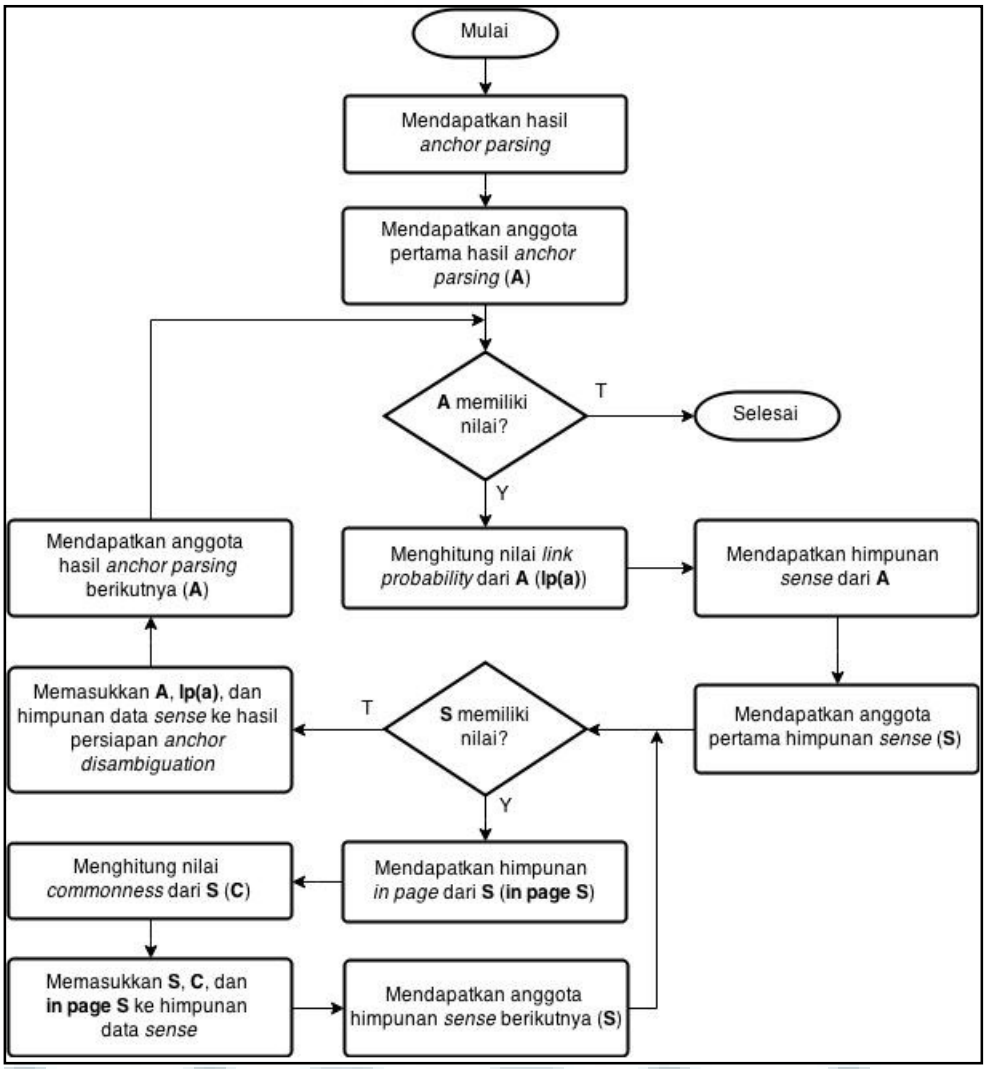


Gambar 3.23 Diagram Alir Proses Utama *Anchor Disambiguation*

C.3.2 Proses Persiapan Anchor Disambiguation

Proses ini menerima hasil dari proses *anchor parsing* dan mempersiapkan data yang dibutuhkan pada proses *anchor disambiguation*. Adapun data tersebut adalah teks *anchor*, *link probability*, dan himpunan data *sense*. Data *sense* terdiri

dari *sense*, nilai *commonness*, dan himpunan *in page*. Gambar 3.24 menunjukkan diagram alir dari proses ini.

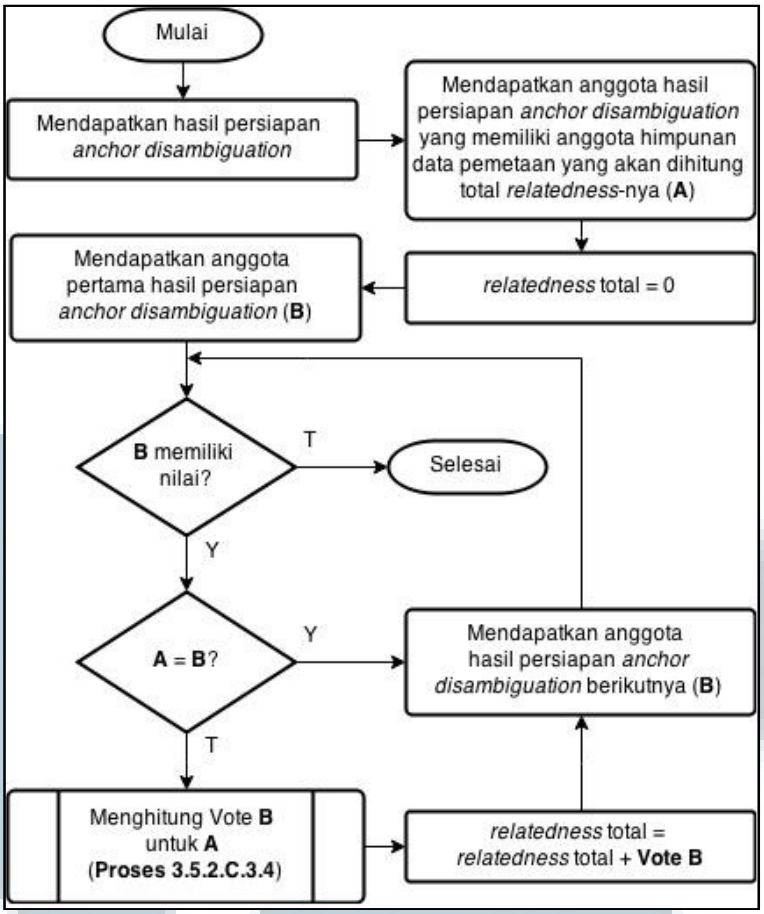


Gambar 3.24 Diagram Alir Proses Persiapan *Anchor Diambiguation*

C.3.3 Proses Menghitung Total Relatedness

Tujuan dari proses ini adalah mengimplementasi persamaan 2.3 yang digunakan untuk menghitung jumlah nilai *vote* yang diberikan ke suatu *anchor* oleh *anchor* lainnya yang juga ditemukan dalam teks masukan. Proses menghitung nilai *vote* yang berasal dari sebuah *anchor* dijelaskan pada subbab 3.5.2.C.3.4. Gambar 3.25 menunjukkan diagram alir dari proses ini.

Implementasi Algoritma ..., William Aprilius, FTI UMN, 2015

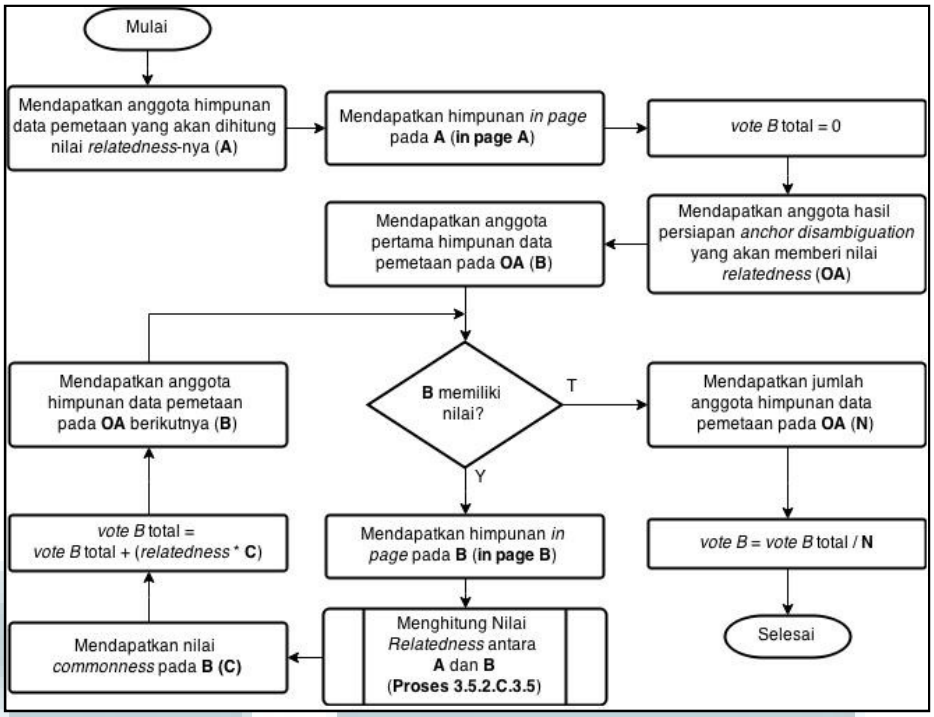


Gambar 3.25 Diagram Alir Proses Menghitung Total *Relatedness*

C.3.4 Proses Menghitung Nilai Vote

Proses ini mengimplementasi persamaan 2.2 yang digunakan untuk menghitung nilai *vote* yang diberikan oleh sebuah *anchor*. Gambar 3.26 menunjukkan diagram alir dari proses ini.

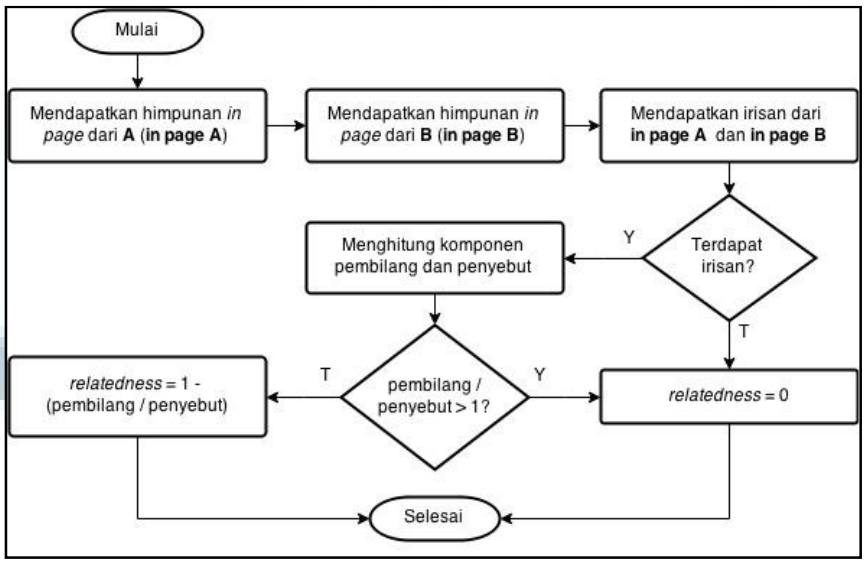
Implementasi Algoritma ..., William Aprilius, FTI UMN, 2015



Gambar 3.26 Diagram Alir Proses Menghitung Nilai *Vote*

C.3.5 Proses Menghitung Nilai Relatedness

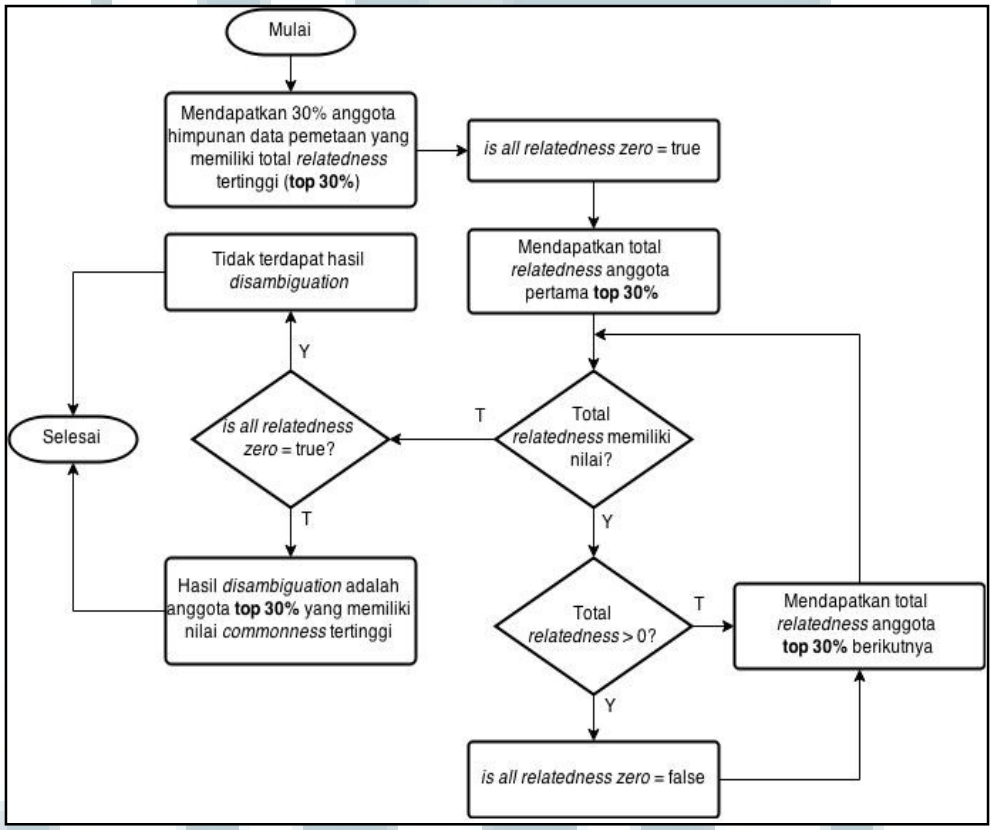
Tujuan dari proses ini adalah mengimplementasi persamaan 2.4 yang digunakan untuk menghitung nilai *relatedness* antara dua halaman Wikipedia. Berikut merupakan diagram alir dari proses ini.



Gambar 3.27 Diagram Alir Proses Menghitung Nilai *Relatedness*

C.3.6 Proses Disambiguation By Threshold

Proses ini mengimplementasi pendekatan yang dilakukan pada teknologi TAGME dalam melakukan *disambiguation*. Pendekatan ini dilakukan dengan menentukan 30% *sense* dengan total nilai *relatedness* tertinggi, kemudian memilih *sense* yang memiliki nilai *commonness* tertinggi. Gambar 3.28 menunjukkan diagram alir dari proses ini.



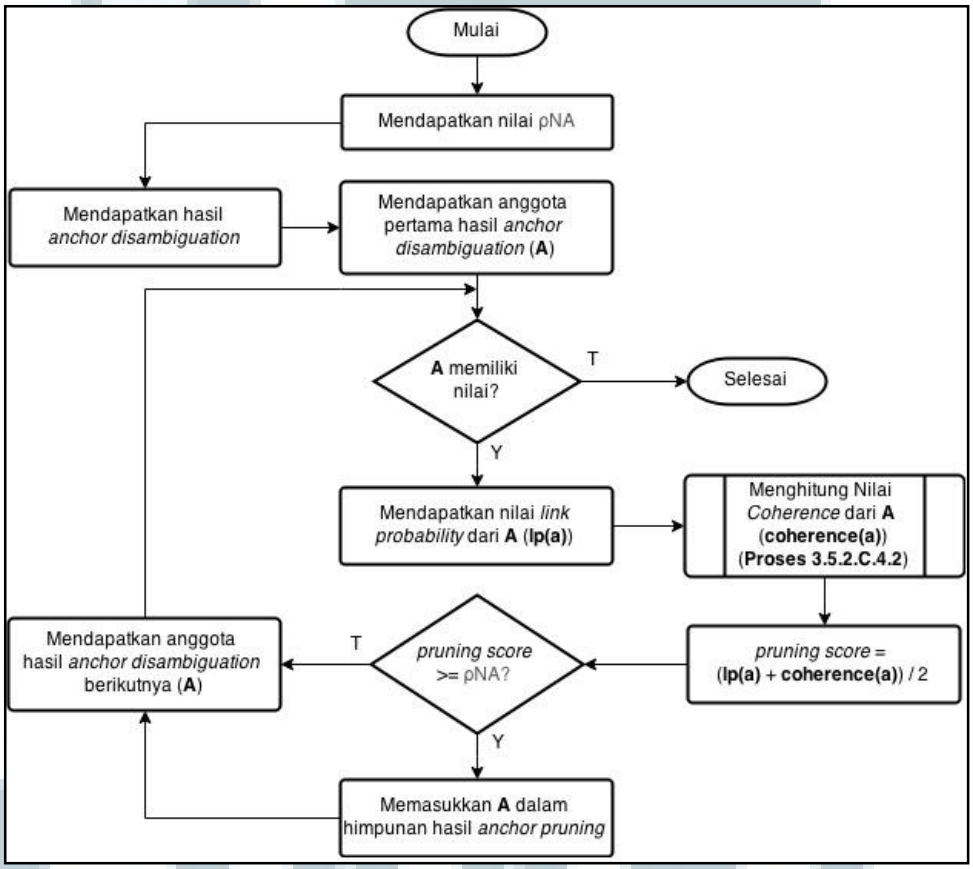
Gambar 3.28 Diagram Alir Proses *Disambiguation By Threshold*

C.4 Proses Anchor Pruning

Diagram alir pada proses ini terdiri dari satu diagram alir utama yang ditampilkan pada subbab 3.5.2.C.4.1 dan satu diagram alir turunan yang ditampilkan pada subbab 3.5.2.C.4.2.

C.4.1 Proses Utama Anchor Pruning

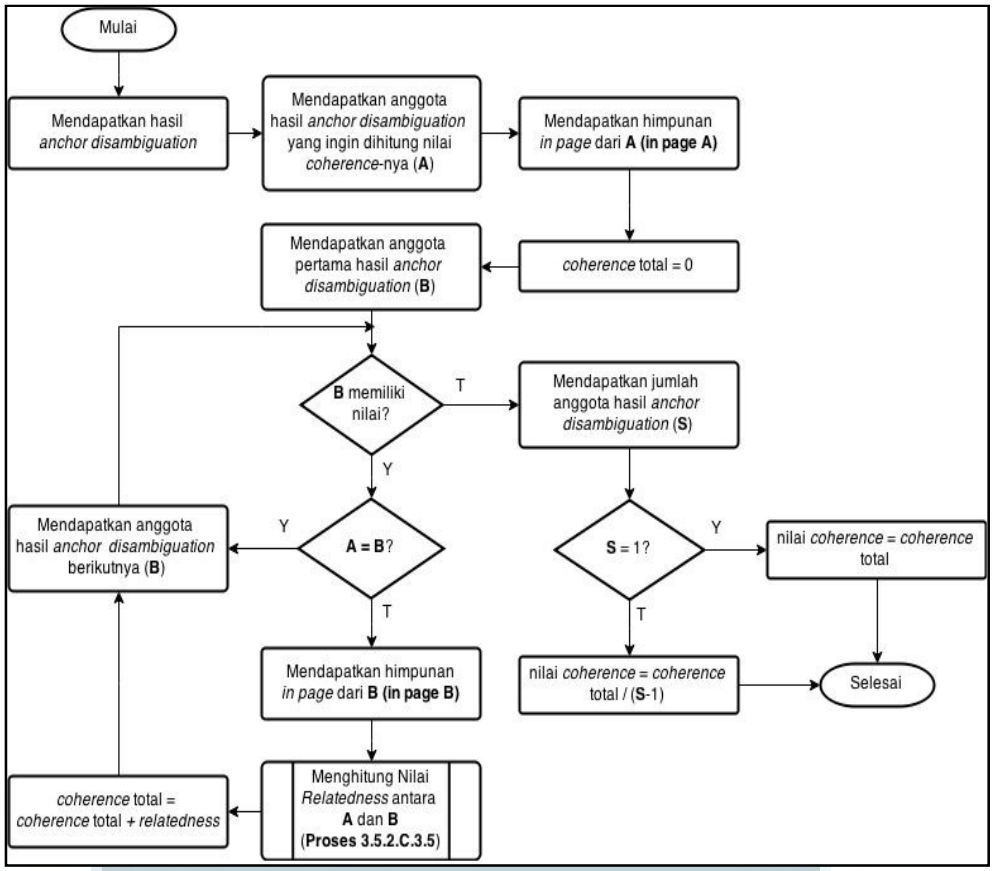
Proses ini melakukan *anchor pruning* seperti yang dilakukan pada teknologi TAGME yang bertujuan untuk menghapus anotasi yang tidak berarti. Hal ini dilakukan dengan menghitung *pruning score* dengan menggunakan persamaan 2.6, kemudian membandingkannya dengan suatu nilai *threshold* ρ_{NA}. Berikut merupakan diagram alir dari proses utama *anchor pruning*.



Gambar 3.29 Diagram Alir Proses Utama *Anchor Pruning*

C.4.2 Proses Menghitung Nilai Coherence

Proses ini mengimplementasi persamaan 2.5 yang digunakan untuk menghitung nilai *coherence*. Gambar 3.30 menunjukkan diagram alir dari proses Menghitung Nilai *Coherence*.

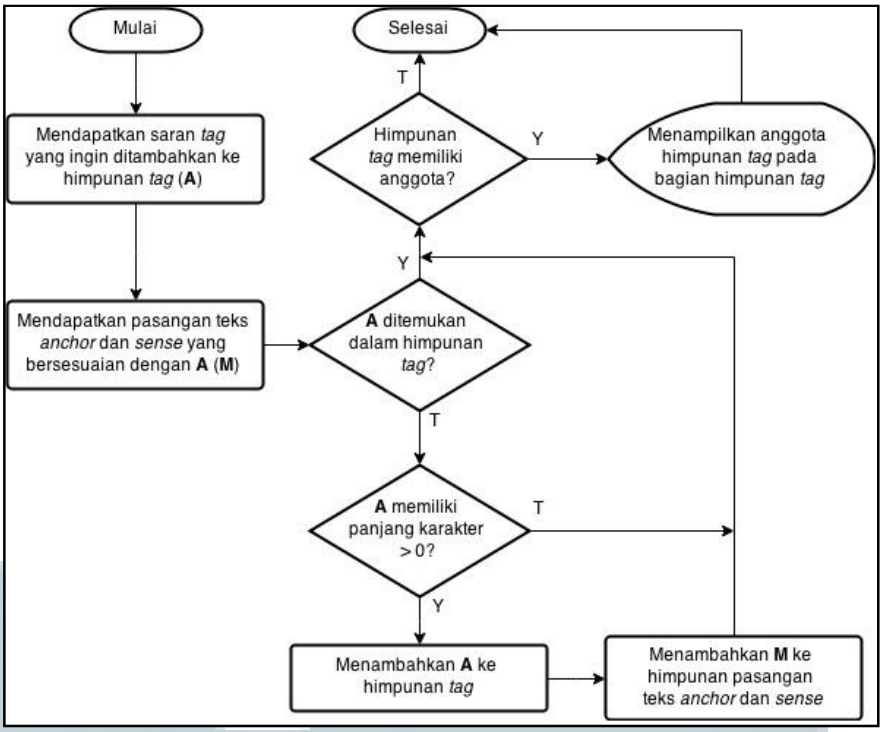


Gambar 3.30 Diagram Alir Proses Menghitung Nilai *Coherence*

D. Proses Menambahkan Saran Tag ke Himpunan Tag

Proses ini dijalankan saat pengguna memilih saran *tag* yang dihasilkan oleh *plugin* menjadi *tag* dari *post*. Gambar 3.31 menunjukkan diagram alir dari proses ini.

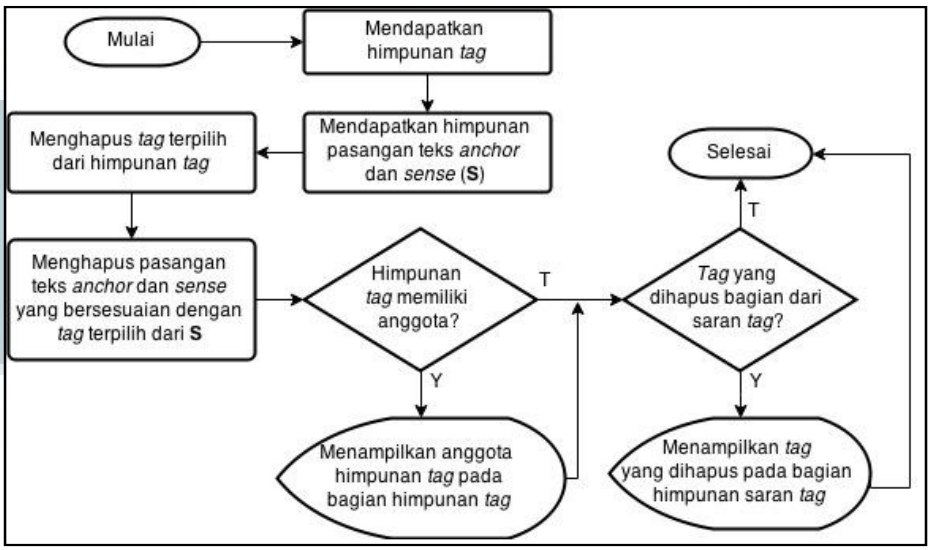
Implementasi Algoritma ..., William Aprilius, FTI UMN, 2015



Gambar 3.31 Diagram Alir Proses Menambahkan Saran *Tag* ke Himpunan *Tag*

E. Proses Menghapus Tag

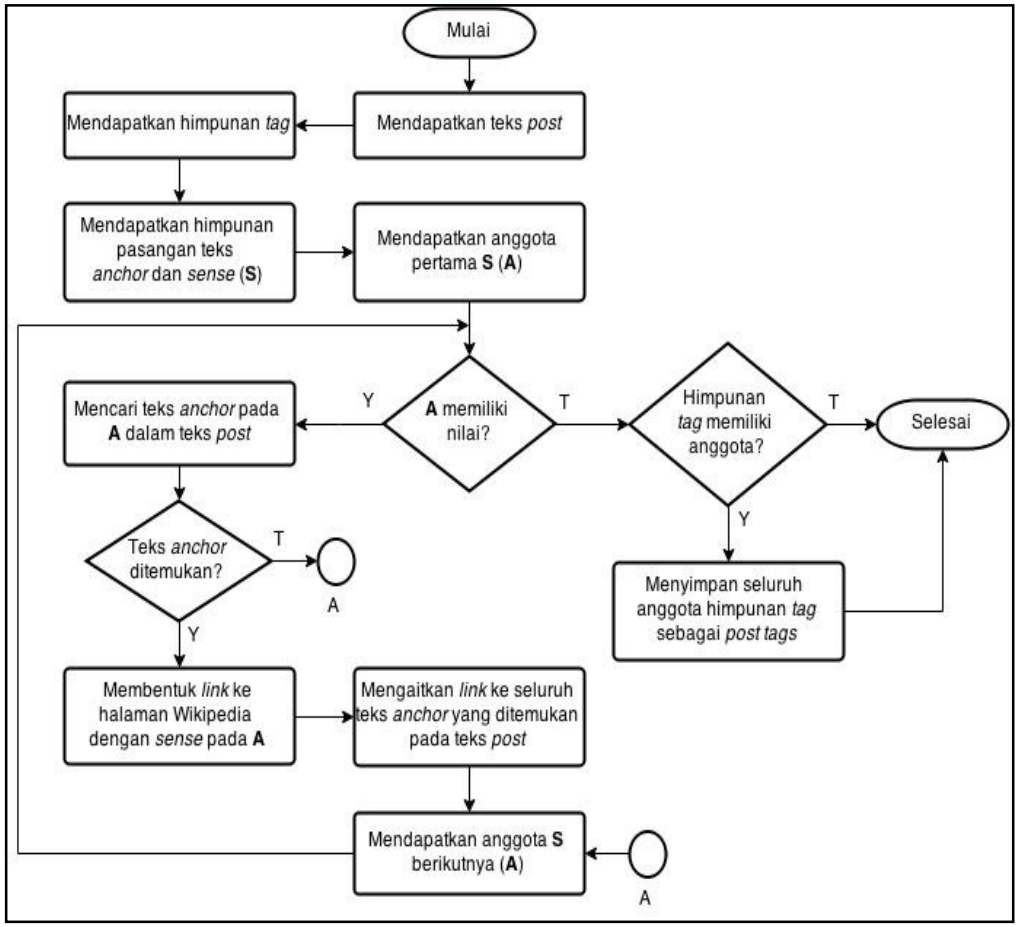
Proses ini dijalankan saat pengguna menghapus *tag* yang telah dimiliki oleh *post* atau membatalkan pilihan terhadap saran *tag* yang dipilih sebelumnya. Gambar 3.32 berikut menunjukkan diagram alir dari proses ini.



Gambar 3.32 Diagram Alir Proses Menghapus *Tag*

F. Proses Menyimpan Post

Proses ini dijalankan saat pengguna menyimpan atau mempublikasi *post*. Berikut merupakan gambar diagram alir dari proses ini.



Gambar 3.33 Diagram Alir Proses Menyimpan *Post*

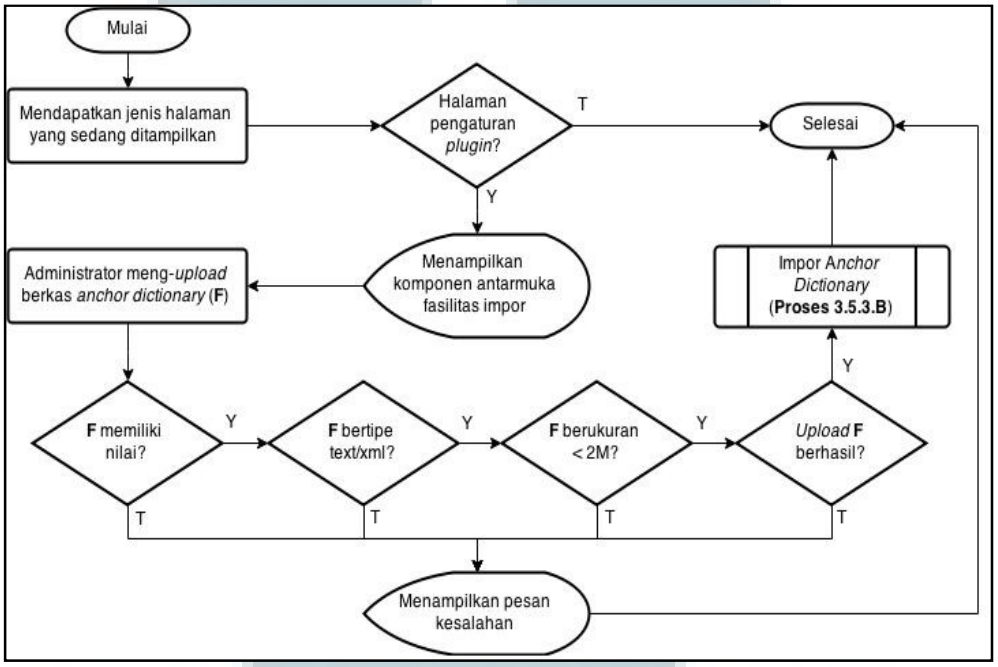
3.5.3 Diagram Alir Administrator

Diagram alir untuk administrator aplikasi *plugin* terdiri dari satu diagram alir utama dan tiga diagram alir turunan. Diagram alir utama ditampilkan pada subbab 3.5.3.A dan ketiga diagram turunan ditampilkan pada subbab 3.5.3.B hingga 3.5.2.D. Diagram alir turunan menggambarkan detail dari proses yang terdapat dalam diagram alir utama atau diagram alir turunan lain.

Implementasi Algoritma ..., William Aprilius, FTI UMN, 2015

A. Proses Utama

Administrator aplikasi *plugin* bertugas untuk mengimpor *anchor dictionary* yang berasal dari berkas XML yang dihasilkan oleh aplikasi *preprocessing*. Berikut merupakan gambar diagram alir dari proses utama administrator aplikasi *plugin*.

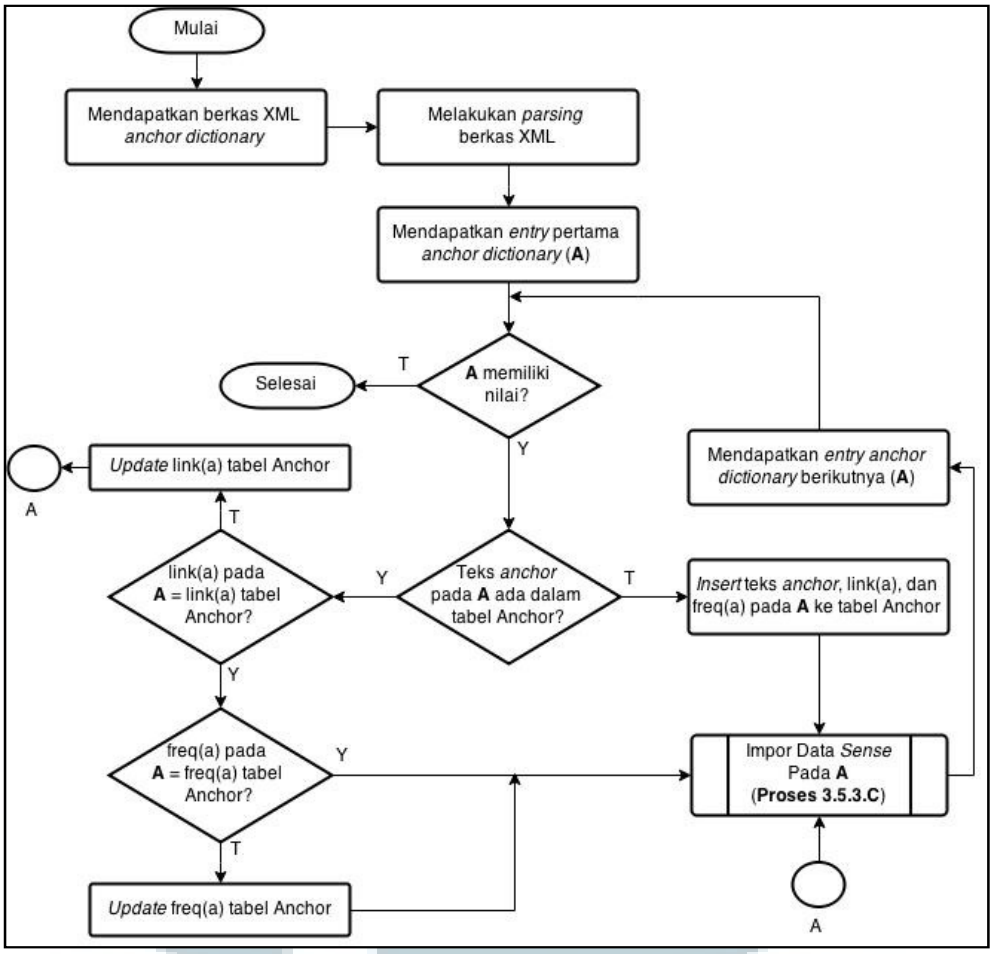


Gambar 3.34 Diagram Alir Proses Utama Administrator Aplikasi *Plugin*

B. Proses Impor Anchor Dictionary

Proses ini bertujuan untuk mem-*parsing* berkas XML dari *anchor dictionary* dan mengimpor data yang terdapat di dalamnya ke empat tabel dalam *database*. Gambar 3.35 menunjukkan diagram alir dari proses ini.

Implementasi Algoritma ..., William Aprilius, FTI UMN, 2015

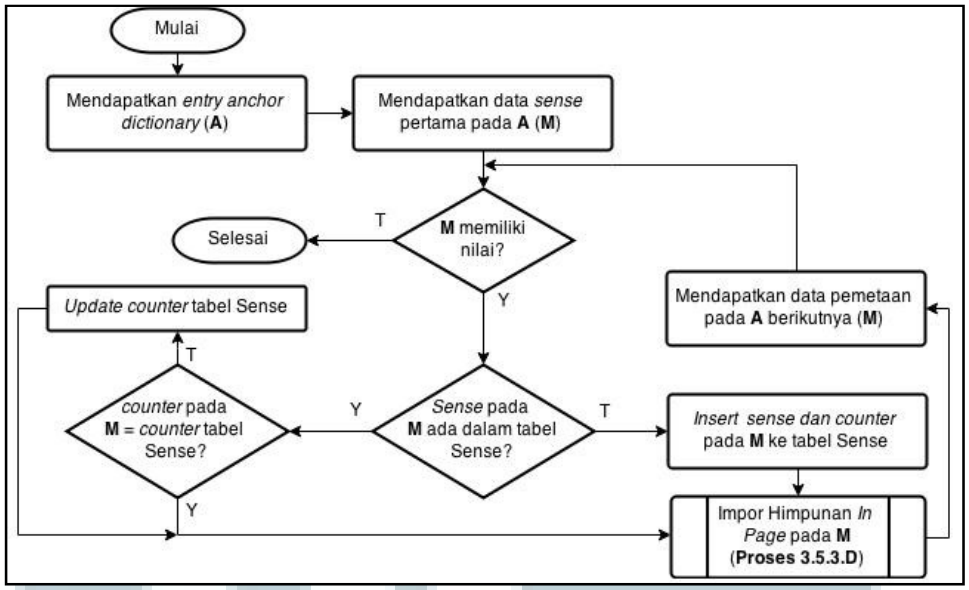


Gambar 3.35 Diagram Alir Proses Impor *Anchor Dictionary*

C. Proses Impor Data Sense

Tujuan dari proses ini adalah mengimpor data *sense* yang terdiri dari *sense*, jumlah pemetaan teks *anchor* ke *sense* tersebut (*counter*), dan himpunan *in page*. Gambar 3.36 menunjukkan diagram alir dari proses ini.

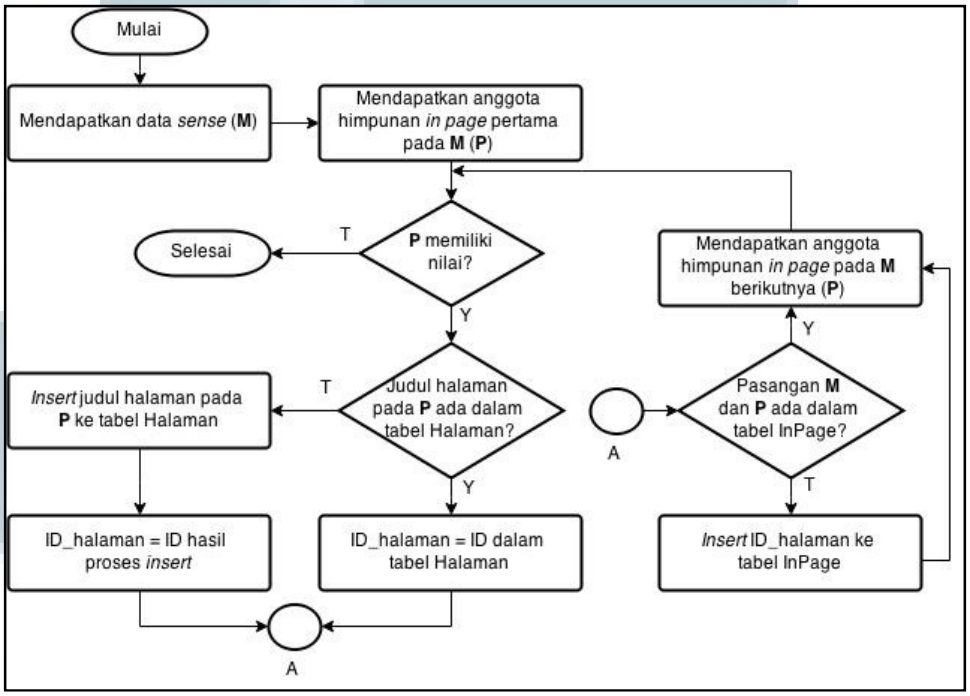
Implementasi Algoritma ..., William Aprilius, FTI UMN, 2015



Gambar 3.36 Diagram Alir Proses Impor Data *Sense*

D. Proses Impor Himpunan In Page

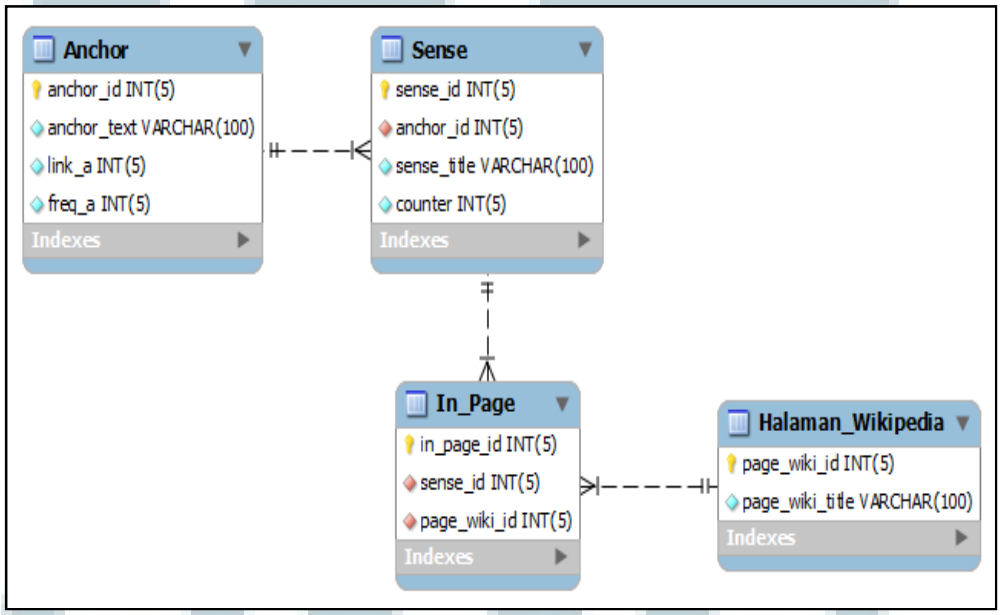
Proses ini bertujuan mengimpor himpunan *in page*. Berikut merupakan diagram alir dari proses ini.



Gambar 3.37 Diagram Alir Proses Impor Himpunan *In Page*

3.5.4 Entity Relationship Diagram

Gambar 3.38 merupakan *Entity Relationship Diagram* (ERD) yang menunjukkan struktur dan relasi dari tabel yang digunakan untuk menyimpan *anchor dictionary* dalam *database*. Tabel Anchor menyimpan data *anchor*. Sebuah data *anchor* dapat memiliki relasi dengan satu atau lebih data *sense* yang disimpan pada tabel *Sense*. Sebuah data *sense* dapat memiliki relasi dengan satu atau lebih data himpunan *in page* yang disimpan pada tabel In_Page. Tabel In_Page berisi *id* dari data *sense* dan *id* dari judul halaman Wikipedia yang disimpan pada tabel Halaman_Wikipedia.



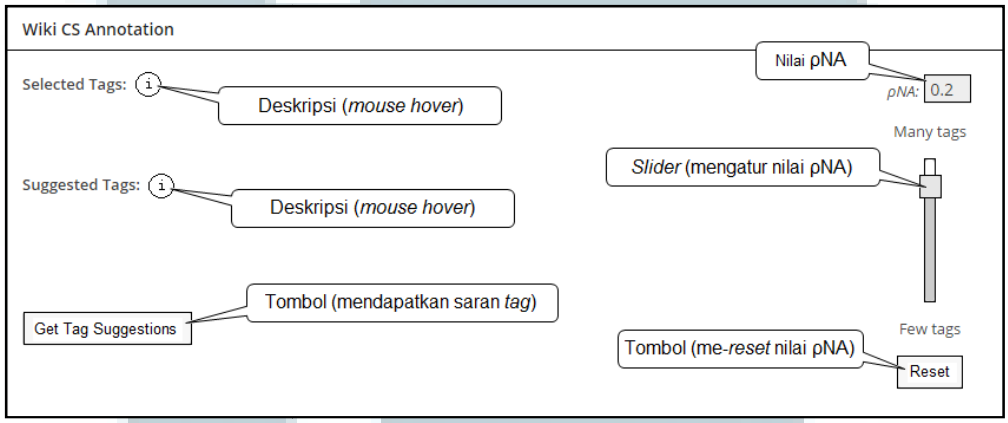
Gambar 3.38 *Entity Relationship Diagram* dari *Anchor Dictionary*

3.5.5 Rancangan Antarmuka Pengguna

Aplikasi *plugin* digunakan oleh dua pihak, yaitu pengguna yang melakukan *entity annotation* dan administrator yang mengimpor *anchor dictionary*. Berikut rancangan antarmuka pengguna untuk kedua pihak tersebut.

A. Pengguna Plugin

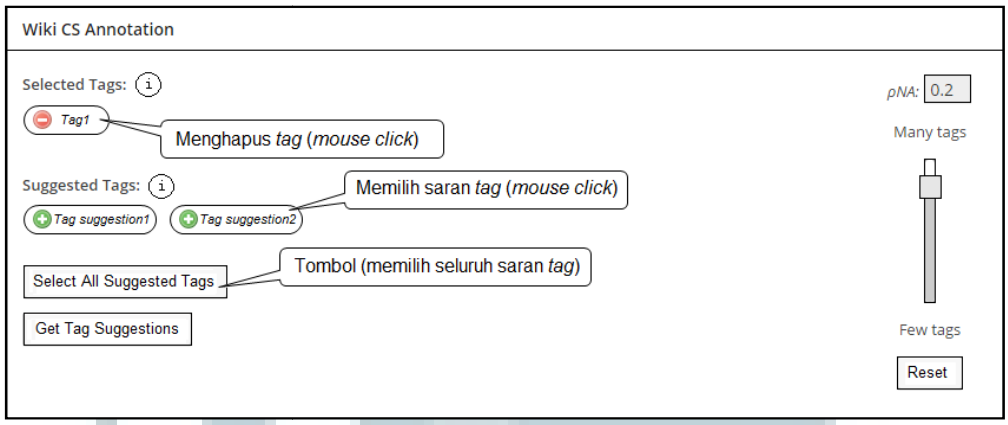
Pada bagian ini dijelaskan rancangan antarmuka untuk pihak pengguna yang melakukan *entity annotation*. Gambar 3.39 menunjukkan rancangan antarmuka jendela awal aplikasi *plugin* yang diletakkan di bawah editor teks WordPress. Pada gambar tersebut ditunjukkan bahwa terdapat dua tombol yang masing-masing digunakan untuk mendapatkan saran *tag* dan me-*reset* nilai ρNA. Selain itu, terdapat sebuah *slider* yang digunakan untuk mengatur nilai ρNA.



Gambar 3.39 Rancangan Antarmuka Jendela Awal Aplikasi *Plugin*

Pada gambar 3.39 juga ditunjukkan bahwa terdapat dua area yang masing-masing digunakan untuk menampilkan saran *tag* dan *tag* pilihan pengguna. Saran *tag* akan ditampilkan pada area di bawah teks "Suggested Tags" dan *tag* pilihan pengguna akan ditampilkan pada area di bawah teks "Selected Tags". Di sebelah kanan kedua teks tersebut terdapat sebuah lingkaran (dengan huruf i di dalamnya) yang jika di-*hover* oleh *mouse*, menunjukkan deskripsi dari area yang berada di bawah masing-masing teks tersebut.

Implementasi Algoritma ..., William Aprilius, FTI UMN, 2015



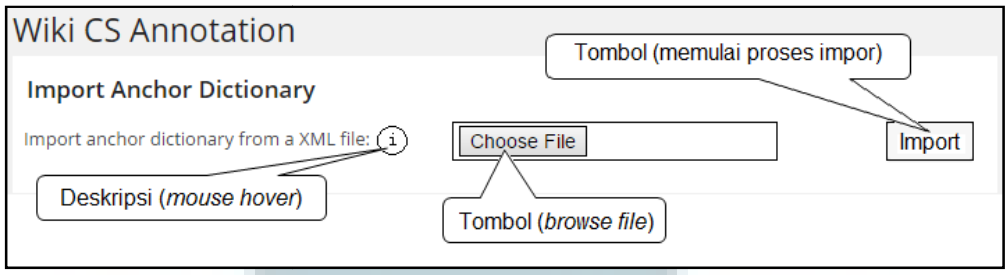
Gambar 3.40 Rancangan Antarmuka yang Menampilkan Saran *Tag* dan *Tag* Pilihan Pengguna

Rancangan antarmuka yang menampilkan saran *tag* dan *tag* pilihan pengguna ditunjukkan pada gambar 3.40. Pada gambar tersebut, terdapat dua saran *tag* dan satu *tag* pilihan pengguna. Saran *tag* dapat dipilih menjadi *tag* dari *post* oleh pengguna dengan mengeklik area saran *tag* tersebut, kemudian saran *tag* akan berpindah ke area di bawah teks "Selected Tags" sebagai *tag* pilihan pengguna. Selain itu, *tag* pilihan pengguna dapat dihapus dengan mengeklik area *tag* tersebut. Jika *tag* tersebut sebelumnya merupakan bagian dari saran *tag*, *tag* tersebut akan ditampilkan kembali pada area di bawah teks "Suggested Tags" sebagai saran *tag*. Pada gambar tersebut juga ditunjukkan sebuah tombol yang digunakan untuk memilih seluruh saran *tag* menjadi *tag* dari *post*.

B. Administrator Plugin

Pada bagian ini dijelaskan rancangan antarmuka untuk pihak administrator yang melakukan impor *anchor dictionary*. Gambar 3.41 merupakan rancangan antarmuka halaman awal fasilitas impor *anchor dictionary*.

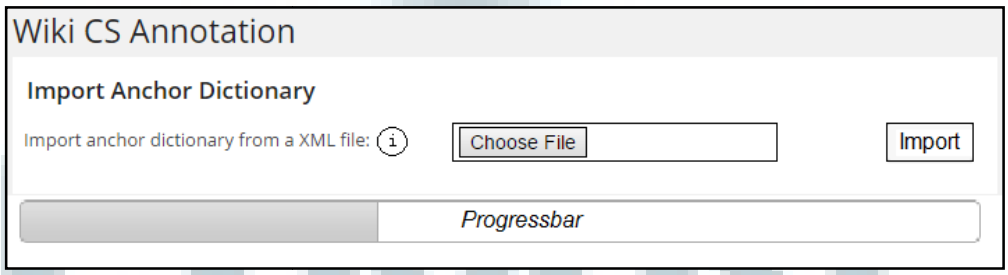
Implementasi Algoritma ..., William Aprilius, FTI UMN, 2015



Gambar 3.41 Rancangan Antarmuka Halaman Awal Fasilitas Impor *Anchor Dictionary*

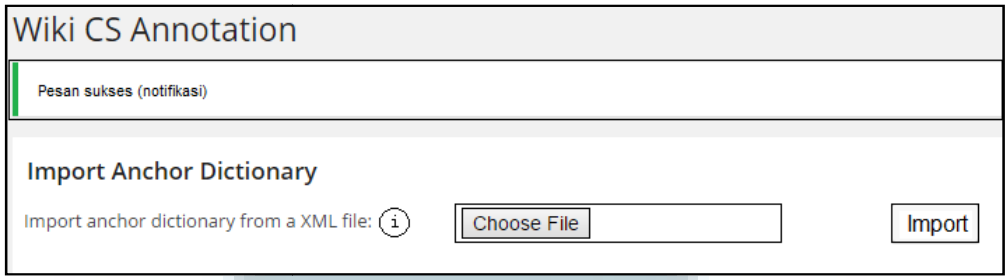
Pada gambar 3.41 ditunjukkan bahwa terdapat sebuah tombol yang digunakan untuk mencari (*browse*) berkas XML dari *anchor dictionary* pada direktori lokal komputer dan sebuah tombol yang digunakan untuk memulai proses impor. Selain itu, di sebelah kanan teks "Import anchor dictionary from a XML file" terdapat sebuah lingkaran (dengan huruf i di dalamnya) yang jika di-*hover* oleh *mouse*, menunjukkan deskripsi dari fasilitas impor *anchor dictionary*.

Selama proses impor, pada halaman fasilitas impor ditampilkan sebuah *progressbar* yang menunjukkan perkembangan proses impor. *Progressbar* tersebut diletakkan di bagian bawah halaman fasilitas impor. Hal ini ditunjukkan pada gambar 3.42.

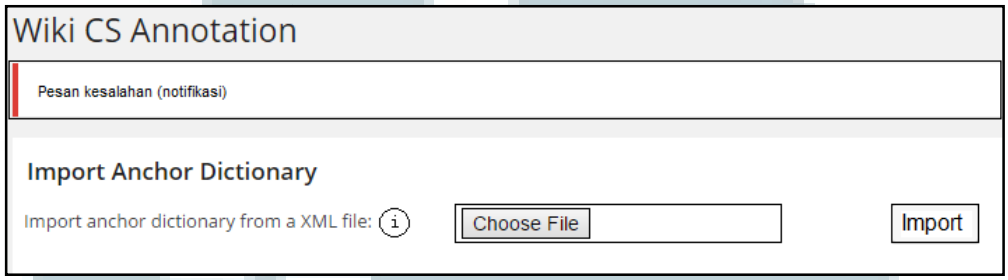


Gambar 3.42 Rancangan Antarmuka Halaman Fasilitas Impor yang Menampilkan Sebuah *Progressbar* Saat Proses Impor Berlangsung

Apabila proses impor berhasil, pesan sukses akan ditampilkan pada area yang ditandai dengan warna hijau seperti ditunjukkan pada gambar 3.43. Apabila proses impor gagal, pesan kesalahan akan ditampilkan pada area yang ditandai dengan warna merah seperti ditunjukkan pada gambar 3.44.



Gambar 3.43 Rancangan Antarmuka Halaman Fasilitas Impor yang Menampilkan Pesan Sukses Proses Impor



Gambar 3.44 Rancangan Antarmuka Halaman Fasilitas Impor yang Menampilkan Pesan Kesalahan Proses Impor