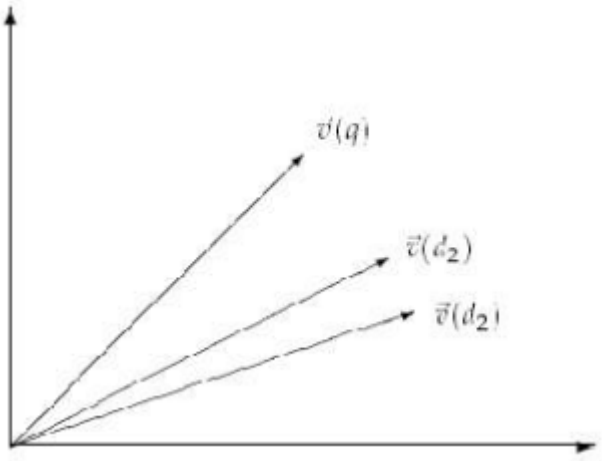
VEKTOR SPACE MODEL

Model ini berkaitan dengan vektor pada aljabar linear. Kumpulan dokumen dalam koleksi kemudian berubah menjadi bentuk *vector space*, dengan sebuah *axis* untuk setiap *terms*. *Term* merupakan kata yang *term*asuk dalam himpunan semua kata dalam dokumen kecuali *stopwords*. *Stopwords* sendiri adalah kata-kata yang sangat sering muncul sehingga dapat diabaikan dalam perhitungan. *Vectos space model* menggambarkan sebuah dokumen sebagai vektor dengan nilai dari *term* untuk setiap *axis*nya(*Vector space retrieval*,2007). Nilai *term* untuk *axis* tersebut adalah nilai *Term Frequency – Inverse Document Frequency* (TF-IDF) . Secara singkat TF-IDF menggambarkan nilai kemunculan *term* pada dokumen.



Gambar. 1. VEKTOR SPACE MODEL

Kedua dokumen yang memiliki kemiripan vektor memiliki kesamaan topik. Kemiripan antar dokumen dilihat melalui relasi antar vektor dokumen satu dengan vektor dokumen lain. Nilai kemiripan antar dokumen ditentukan oleh perbedaan sudut antara kedua vektor, tanpa mempertimbangkan panjang vektor. Nilai kemiripan dokumen dikenal dengan *cosine similarity* yaitu perkalian kedua vektor dibagi dengan perkalian panjang dua *vektor* tersebut, dengan sim sebagai *similarity*, d_1 dokumen pertama, d_2 dokumen kedua, $\vec{V}$ *vector*, dan $|\vec{V}|$ panjang dari *vector*[4]. Persamaan *cosine similarity* ditunjukkan oleh Persamaan 2.1.

$$sim(d1, d2) = \frac{\vec{V}(d_1).\vec{V}(d_2)}{|\vec{V}(d_1)||\vec{V}(d_2)|}$$

Persamaan 2.1 *Cosine similarity* antar d_1 dan d_2

Perhitungan kemiripan dari pencarian dokumen ditentukan oleh dokumen yang paling *relevan* dengan *query* yang ada ditunjukkan oleh Persamaan 2.2 dengan q *query* dan d dokumen.

$$\text{score}(q, d) = \frac{\vec{V}(q).\vec{V}(d)}{|\vec{V}(q)||\vec{V}(d)|}$$

Persamaan 2.2 *Cosine Similarity* antara *query* dan dokumen

Persamaan tersebut akan diaplikasikan untuk semua dokumen dalam koleksi, yang berarti akan didapat sejumlah n *score* dari n dokumen dalam *collection*.

2.1.4. Term Frequency – Inverse Document Frequency

Dalam *vector space model*, sebuah dokumen digambarkan sebagai sebuah vektor dengan satu *terms* setiap sumbunya. Sumbu tersebut merupakan relasi antara *terms* dan dokumen. Bentuk paling sederhana dari relasi tersebut adalah *term frequency*, yaitu jumlah kemunculan sebuah *term* pada tiap dokumen[5]. Ini berarti bahwa semakin banyak kemunculan *term* pada sebuah dokumen, semakin besar pula nilai *term frequency* yang dimiliki.

Perhitungan *term frequency*, berkembang lebih kompleks dengan adanya *terms* yang sering muncul tidak hanya pada sebuah dokumen, melainkan hampir semua dokumen. Kata- kata tersebut bukan s*topwords* tetapi memiliki frekuensi kemunculan yang tinggi, misalkan dalam dokumen teknologi informasi, istilah komputer pasti sering ditemui. Kemudian muncul ide untuk menurunkan skala berat dari *term* yang memiliki frekuensi kemunculan tinggi di tiap dokumen (*Scoring and term weighting*,2007). Untuk memperkecil nilai dari *terms* tersebut digunakanlah persamaan sebagai berikut dengan N jumlah dokumen dalam koleksi, Df jumlah dokumen yang memiliki *term* t. Persamaan *inverse document frequency* ditunjukkan oleh Persamaan 2.3.

$$\mathrm{idf}_t = \log \frac{N}{\mathrm{df}_t} + 1$$

Persamaan 2.3 *Inverse Document Frequency*

Kemunculan *term*s dari setiap dokumen kemudian dikalikan dengan *inverse document frequency* supaya didapat nilai yang disebut *weight*. Persamaan ini ditunjukkan oleh Persamaan 2.4.

$$\mathrm{tf} - \mathrm{idf}_{t,d} = \mathrm{tf}_{t,d} \; \mathrm{X} \; \mathrm{idf}_{t}$$

Persamaan 2.4 *Term Frequency- Inverse Document Frequency*

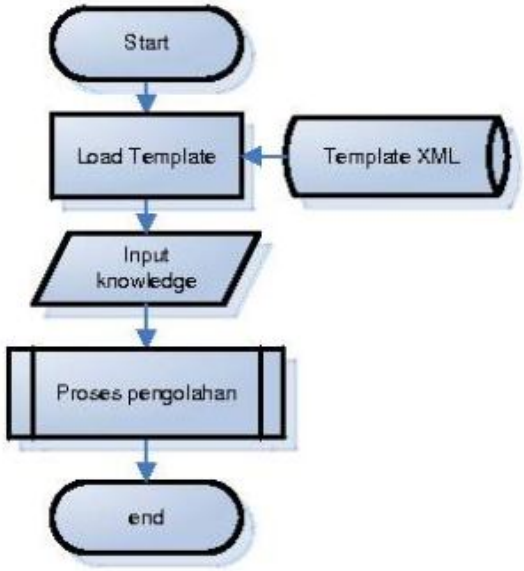
Apakah 20 kali kemunculan dari sebuah *term* sungguh- sungguh bernilai dua puluh kali lebih akurat dari sebuah kemunculan(*Scoring and term weighting*,2007). Hal ini menimbulkan adanya variasi dalam Persamaan 2.3. Usaha untuk memperkecil perbedaan nilai dari *term frequency*, dilakukan dengan mengganti nilai tf dari tf-idf dengan Persamaan 2.5.

$$\left\{ wf_{t,d} = \begin{Bmatrix} 1 + \log tf_{t,d} \\ 0 \end{Bmatrix} \begin{matrix} \text{if } \mathrm{tf}_{t,d} > 0 \\ \text{otherwise} \end{matrix} \right.$$

Persamaan 2.5 Persamaan $\mathrm{wf}_{t,d}$

3.1. Input Knowledge

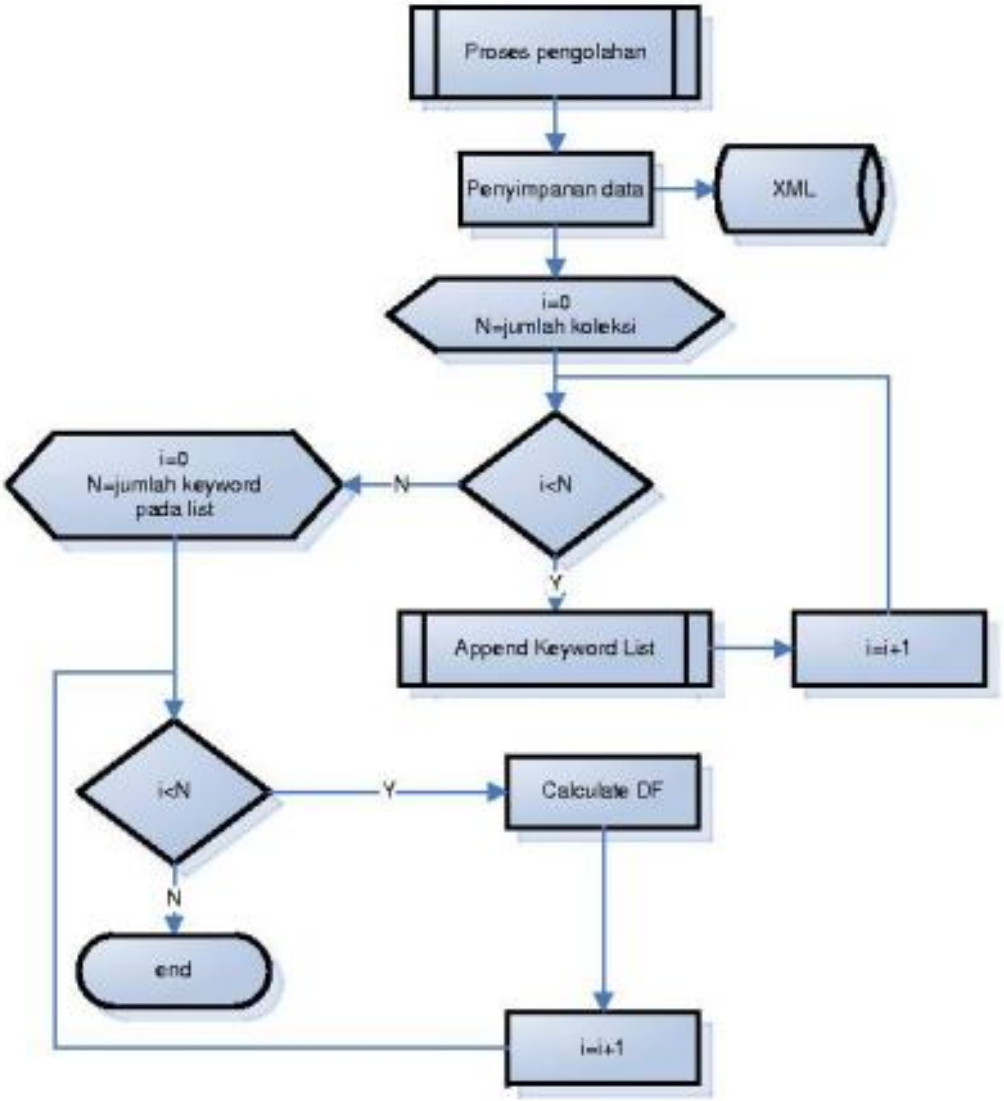
Proses input knowledge merupakan proses yang membantu kita dalam melakukan input data baru sebagai knowledge. Knowledge ini kemudian disimpan dalam komputer. Sistem customer support akan melakukan pencarian pada knowledge ini sewaktu melakukan pencarian jawaban. Proses input knowledge pada sistem ditunjukkan oleh Gambar 3.1.



Gambar 3.1 Alur diagram Input Knowledege

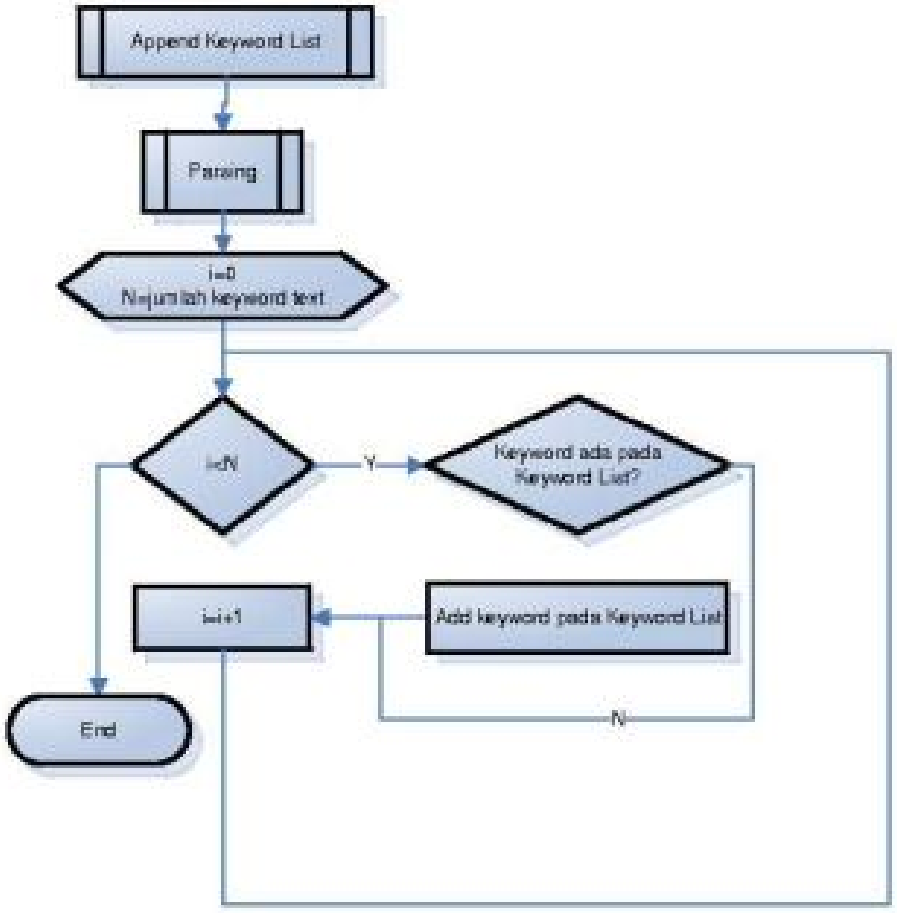
Proses dimulai dengan melakukan loading template yang sudah ada. Template ini akan memberikan aturan bagaimana struktur dari knowledge yang akan dibuat. Kemudian user membuat sebuah knowledge data ke dalam aplikasi berdasarkan template tersebut.

Pengolahan knowledge melakukan penyimpanan data, penambahan list keyword, dan penghitungan nilai document frequency. Pengolahan knowledge ini ditunjukkan oleh Gambar 3.2.



Gambar 3.2 Alur diagram Pengolahan Knowledge

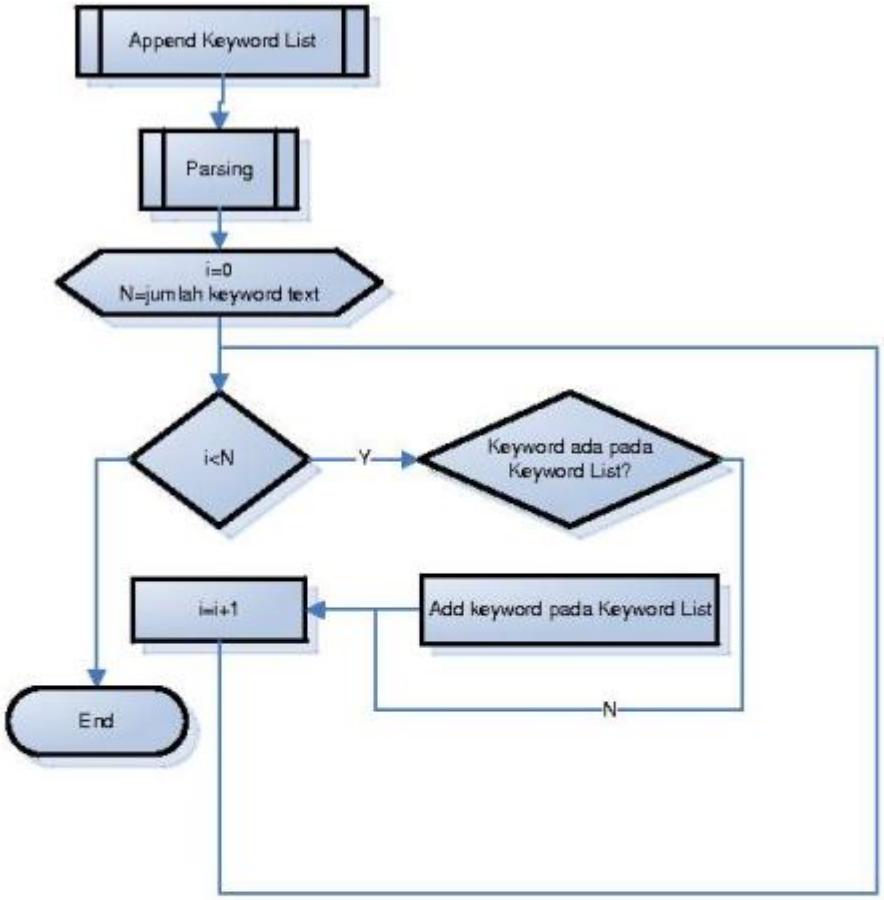
Penambahan keyword list merupakan proses menambahkan keyword baru ke dalam sistem. Penamabahan keyword list ditunjukkan oleh Gambar 3.3



Gambar 3.3 Alur diagram Append Keyword List

3.2. Pencarian knowledge

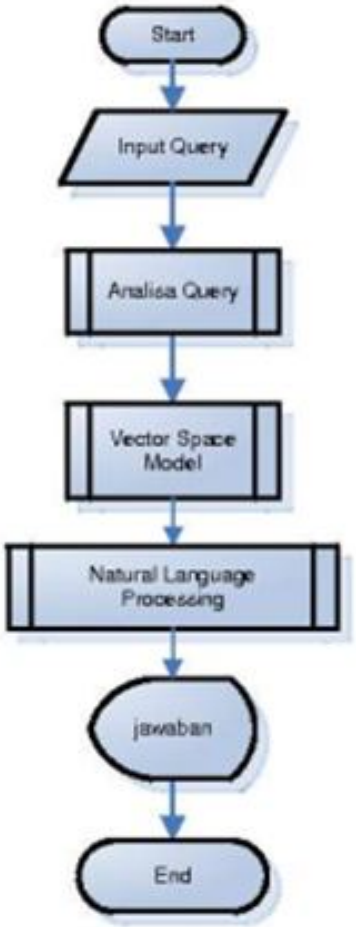
Proses pencarian knowledge merupakan proses dimana sistem melakukan pencarian jawaban dari koleksi. Proses pencarian dilakukan menggunakan penggabungan antara XML retrieval, vector space model, dan natural language processing. Alur proses pencarian dalam sistem secara garis besar dapat dilihat pada Gambar 3.2.



Gambar 3.3 Alur diagram Append Keyword List

3.2. Pencarian knowledge

Proses pencarian knowledge merupakan proses dimana sistem melakukan pencarian jawaban dari koleksi. Proses pencarian dilakukan menggunakan penggabungan antara XML retrieval, vector space model, dan natural language processing. Alur proses pencarian dalam sistem secara garis besar dapat dilihat pada Gambar 3.2.

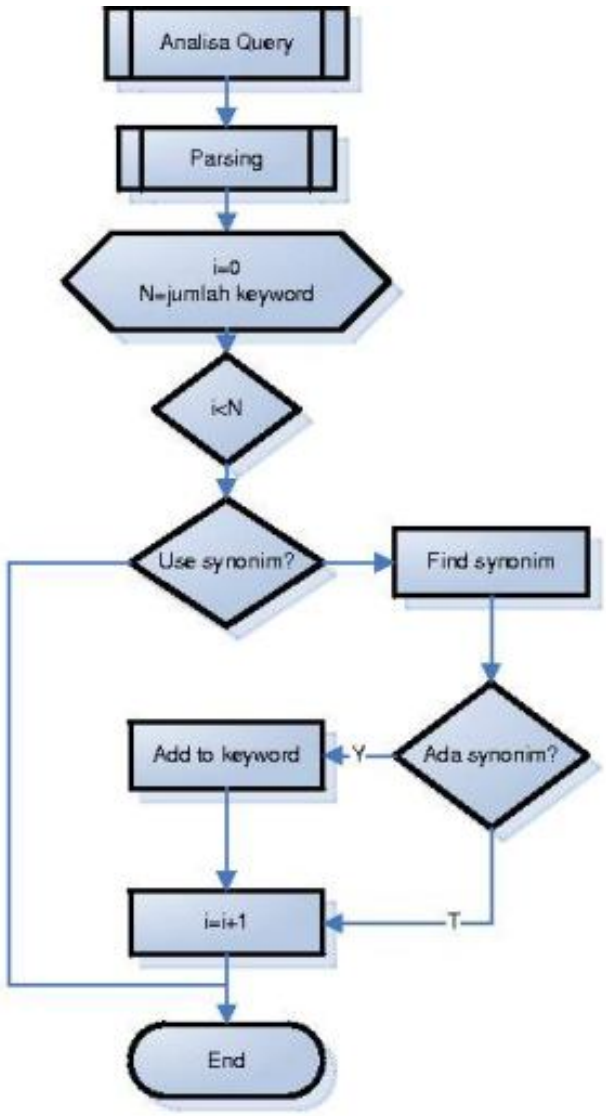


Gambar 3.4 Alur diagram Pencarian Knowledge

Pada alur diagram pencarian knowledge, proses dimulai dengan input query berupa text. Query tersebut akan melalui proses-proses seperti analisa query, analisa vector space model, dan analisa natural language processing. Setelah melalui proses tersebut pengguna memiliki kesempatan untuk memperluas pencarian. Penyimpanan knowledge dalam XML memungkinkan kita menggunakan model hierarki yang ada.

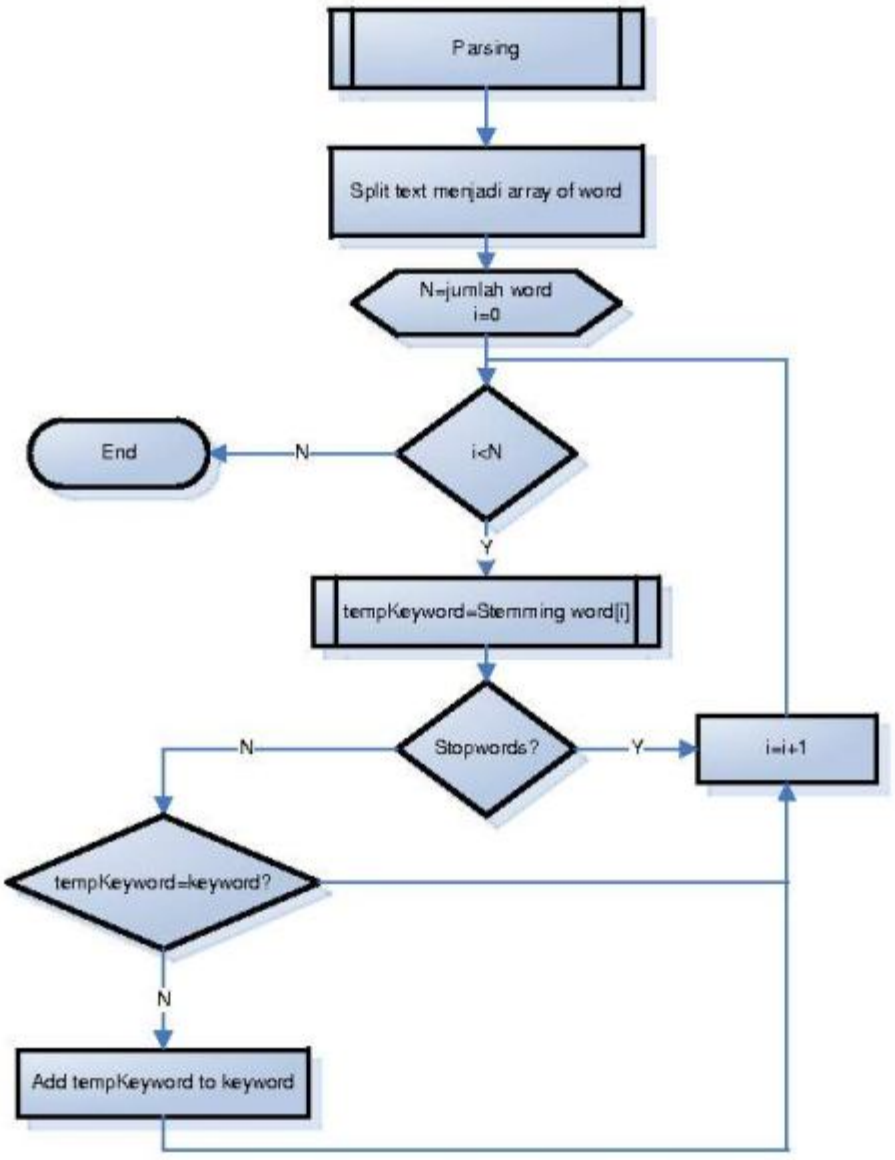
3.2.1. Proses analisa query dalam sistem

Proses analisa query merupakan pengolahan text query menjadi kumpulan keywords melalui proses parsing dan pencarian sinonim. Proses analisa query pada sistem ditunjukkan pada Gambar 3.5.

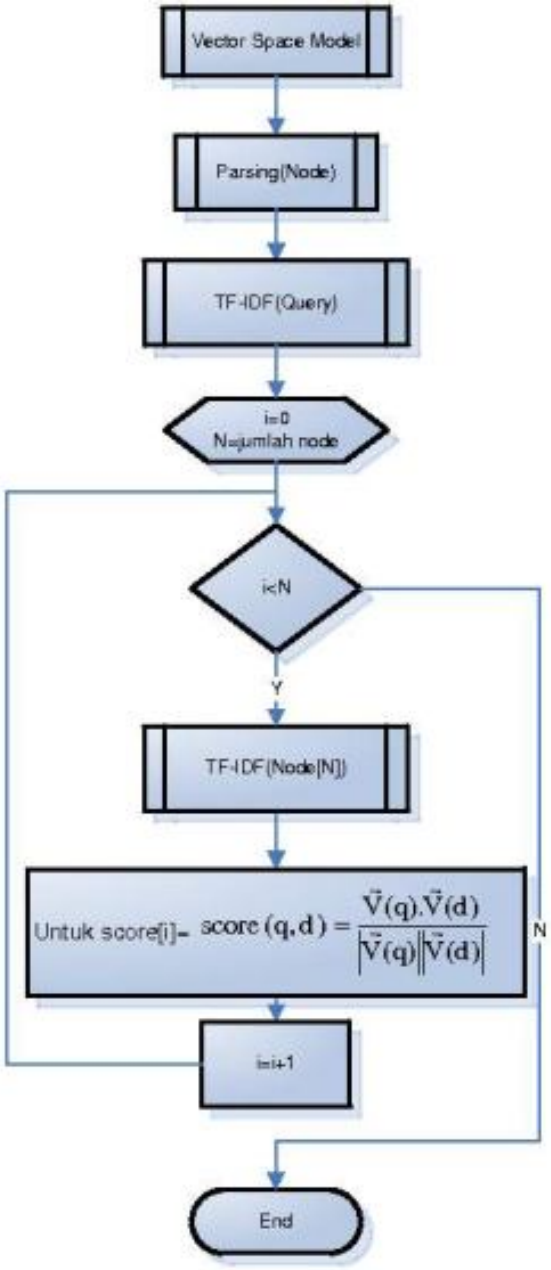


Gambar 3.5 Alur diagram Analisa Query

Proses parsing query melakukan pengolahan text menjadi keywords. Proses parsing query ditunjukkan oleh Gambar 3.6.

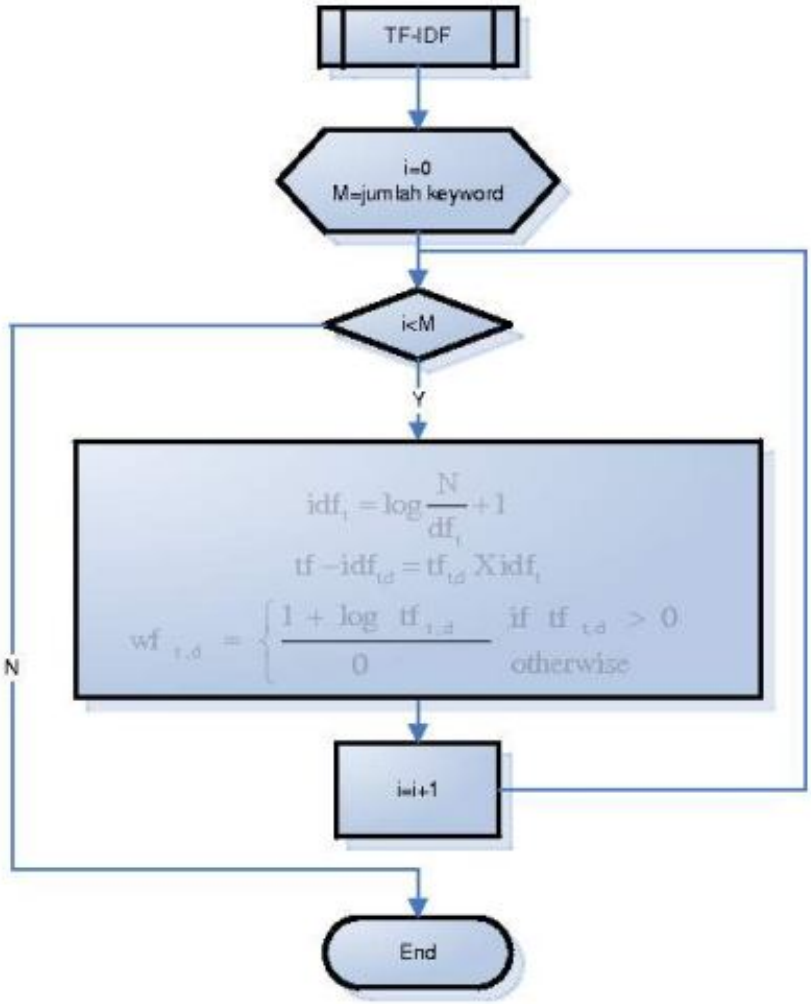


Proses parsing query dimulai dengan melakukan splitting pada text query untuk mendapatkan array of word. Proses splitting ini dilakukan dengan menggunakan karakter-karakter di dalam kalimat selain huruf sebagai pemisah antar word. Hasil dari proses splitting ini adalah kumpulan word, yang disebut array of word. Array of word ini akan melalui proses pengecekan stopwords. Pengecekan ini berfungsi untuk menghilangkan word yang kurang penting dalam query. Proses dilanjutkan dengan proses stemming untuk mendapatkan kata dasar dari word pada query. Kata dasar ini yang menjadi keyword dari query.

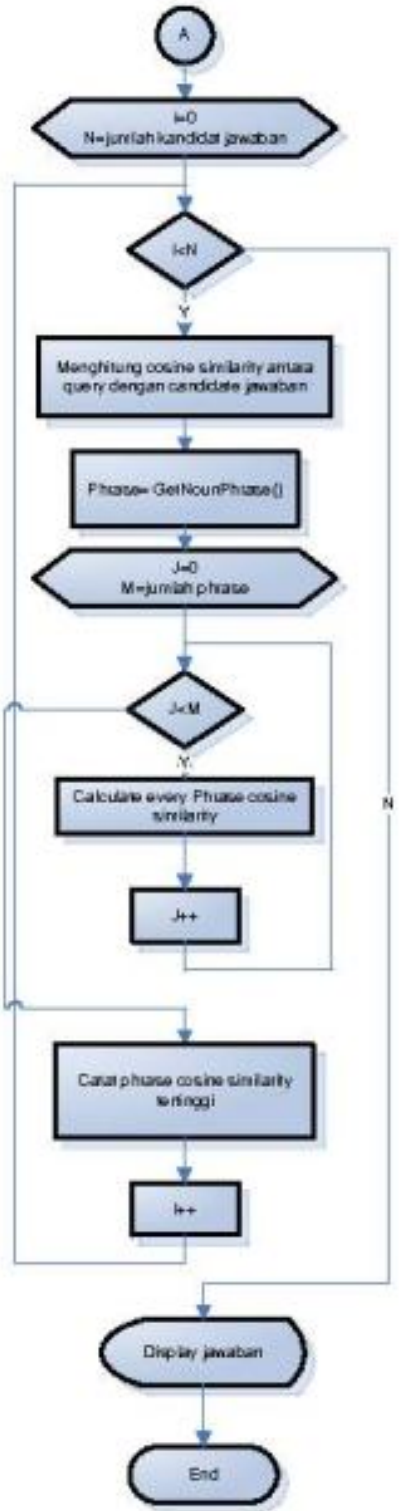


Gambar 3.7 Alur diagram Vector Space Model

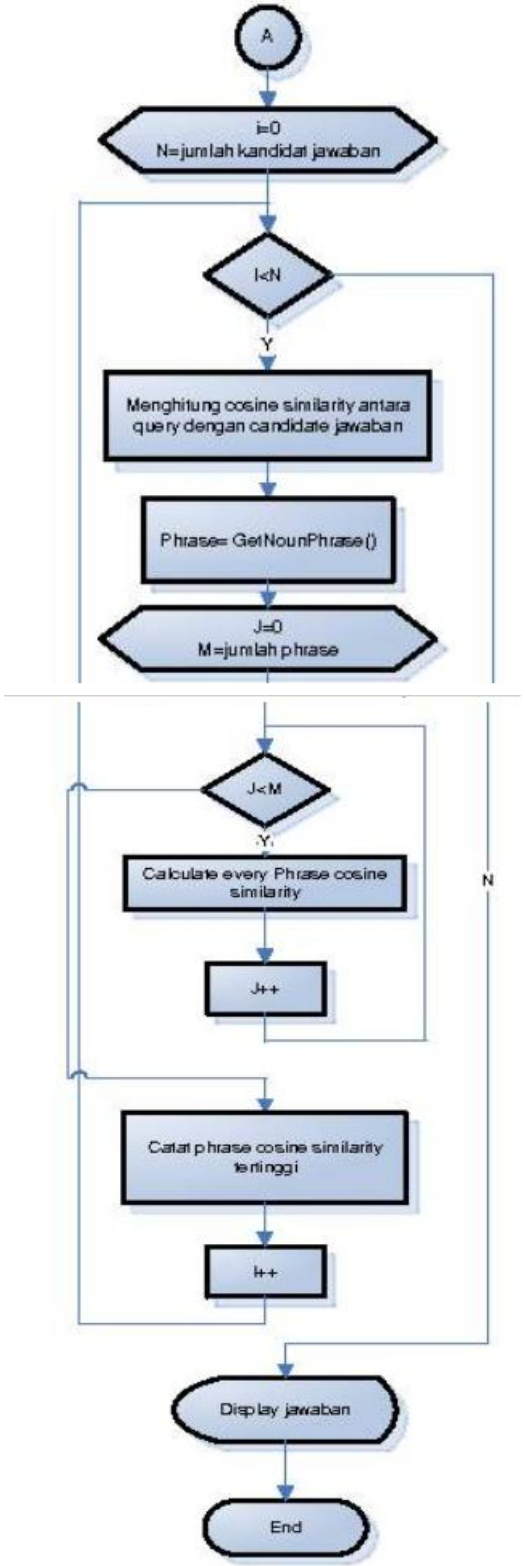
Proses perhitungan TF-IDF pada alur diagram vector space model melakukan perhitungan nilai weight keyword menggunakan hasil modifikasi term frequency- inverse document frequency. Proses perhitungan ini ditunjukan oleh alur diagram pada Gambar 3.8.



Gambar 3.8 Alur diagram Tf-Idf

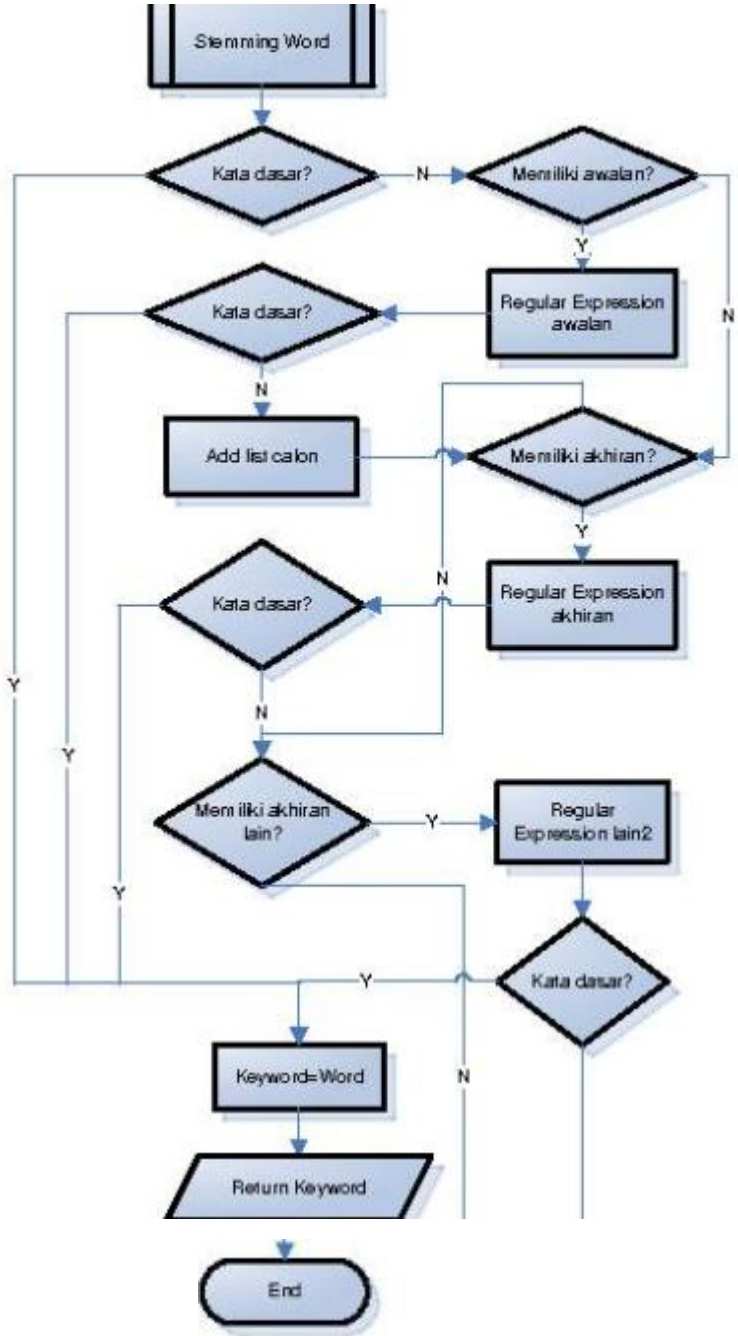


ambar 3.10 Alur diagram Natural Language Processing(b)


A
i=0
N=jumlah kandidat jawaban
I<N
Y
Menghitung cosine similarity antara query dengan candidate jawaban
Phrase= GetNounPhrase()
J=0
M=jumlah phrase
J<M
Y
Calculate every Phrase cosine similarity
J++
Catat phrase cosine similarity tertinggi
I++
N
Display jawaban
End

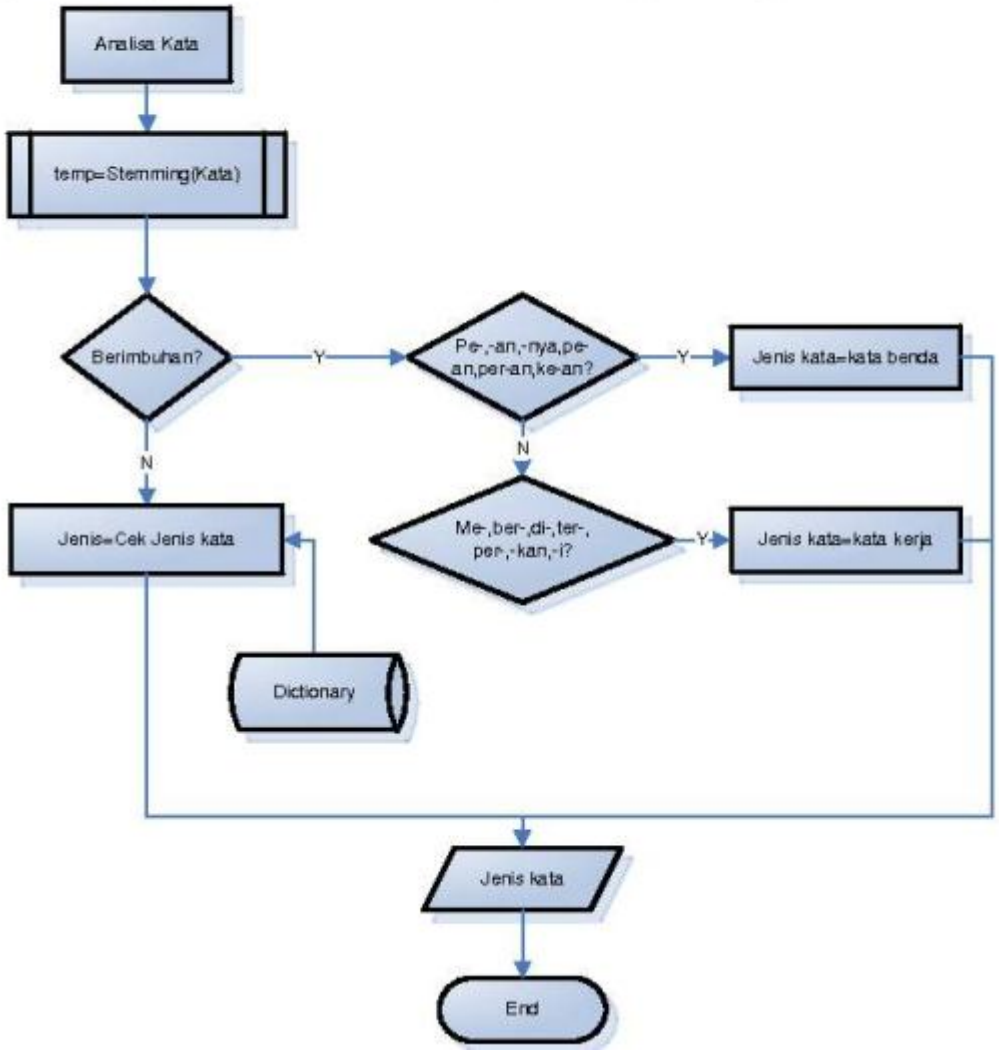
3.2.5. Proses Stemming

Proses stemming merupakan proses dimana sebuah kata berimbuhan diubah menjadi bentuk kata dasarnya. Hal ini berguna untuk meminimalkan, memberikan efisiensi dalam pemakaian terms, dan dapat pula membantu menentukan jenis kata. Alur proses stemming ditunjukkan oleh Gambar 3.11.



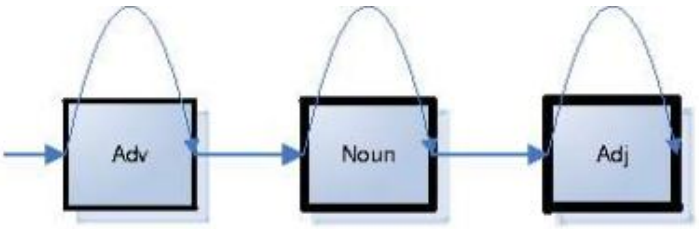
3.2.6. Proses pencarian jenis kata

Untuk menentukan sebuah frase kita perlu mengetahui jenis- jenis kata pembentuknya. Alur proses menentukan jenis kata dapat dilihat pada Gambar 3.12.



3.2.7. Frase Kata Benda

Frase kata benda dalam bahasa indonesia ditentukan melalui aturan-aturan yang ditunjukkan oleh pada Gambar 3.13. Kotak yang memiliki garis tebal merupakan final state.



Gambar 3.13 Aturan pembentukan Noun phrase

Misalkan sebuah frase "modem cepat" memenuhi syarat dari aturan pembentukan noun phrase tapi tidak dengan "cepat modem".