BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Analisis sistem dilakukan untuk mempelajari dan menganalisa kebutuhan sistem yang akan dibuat sehingga dapat dilakukan perancangan sistem dengan kriteria dan perangkat-perangkat yang ditentukan. Analisis system bertujuan untuk mengidentifikasi permasalahan permasalahan yang ada pada system dimana aplikasi dibangun yang meliputi perangkat lunak (software), pengguna (user) serta hasil analisis terhadap system dan elemen-elemen yang terkait. Analisis ini diperlukan sebagai dasar bagi tahapan perancangan system.

3.1.1 Gambaran Umum

Aplikasi pencarian ayat al-qur'an ini akan mencari ayat yang diinginkan pengguna dan dianggap penting maupun ayat yang belum dimengerti baik makna ataupun artinya. Proses pencarian dilakukan denagn mengambil seluruh data ayatdan terhjemahan dalam al-qur'an yang akan diolah. Proses kerja system ini adalah user memasukkan data berupa query (topik yang akan dicari), kemudian query tersebut akan diproses, mulai dari preprocessing hingga proses pembobotan dan perangkingan kata kemudian akan dicari ayat sesuai .

System akan menampilkan daftar ayat yang sesuai dengan query. Contohnya adalah pengguna memasukkan query berbentuk kata berbahasa Indonesia yang akan dicari kedalam sistem pencarian ayat al-qur'an kemudian aplikasi akan memproses dan menghasilkan daftar list ayat dan terjemahan al-qur'an yang akan ditampilkan sesuai nilai perhitungan dan yang memliki nilai kemunculan lebih dari satu dan perangkingan.

3.1.2 Kebutuhan Pembangunan Sistem

1. Analisis Perangkat Keras

   Perangkat keras adalah device yang digunakan untuk menunjang dalam pembuatan sistem. Dalam per 24 an sistem ini perangkat keras yang digunakan yaitu laptop dengan spesifikasi :

   a. Processor Dual Core
   b. RAM 2 GB
   c. HDD 320 GB
   d. Monitor 14”
   e. Mouse

1. Analisis Perangkat Lunak

   Perangkat lunak adalah program atau aplikasi yang digunakan untuk membangun sistem. Perangkat lunak yang dibutuhkan dalam pembuatan sistem ini adalah:

   a. Windows XP
   b. Aplikasi server, dalam hal ini digunakan XAMPP
   c. Editor PHP, dalam hal ini digunakan EditPlus3
   d. Browser, dapat menggunakan aplikasi internet seperti Internet Explorer, Mozilla Firefox, Opera dan lain-lain.
   e. SQLyog Enterprise

3.1.3 Deskripsi system

Aplikasi ini menerapkan teknik sistem temu kembali informasi (information retrieval), yang tahapan awalnya adalah preprocessing terhadap dokumen. Dilanjutkan proses analysing yaitu menghitung kedekatan relevansi antara terjemahan ayat dengan query user. Hingga mendapatkan nilai similiarity terjemahan ayat dengan query. Dan dilakukan proses pencarian ayat yang mirip menggunakan algoritma vector space model.

Sistem mengolah data berupa terjemahan ayat al-qur’an yang kemudian menghasilkan term berupa kata dasar yang disimpan dalam database untuk proses

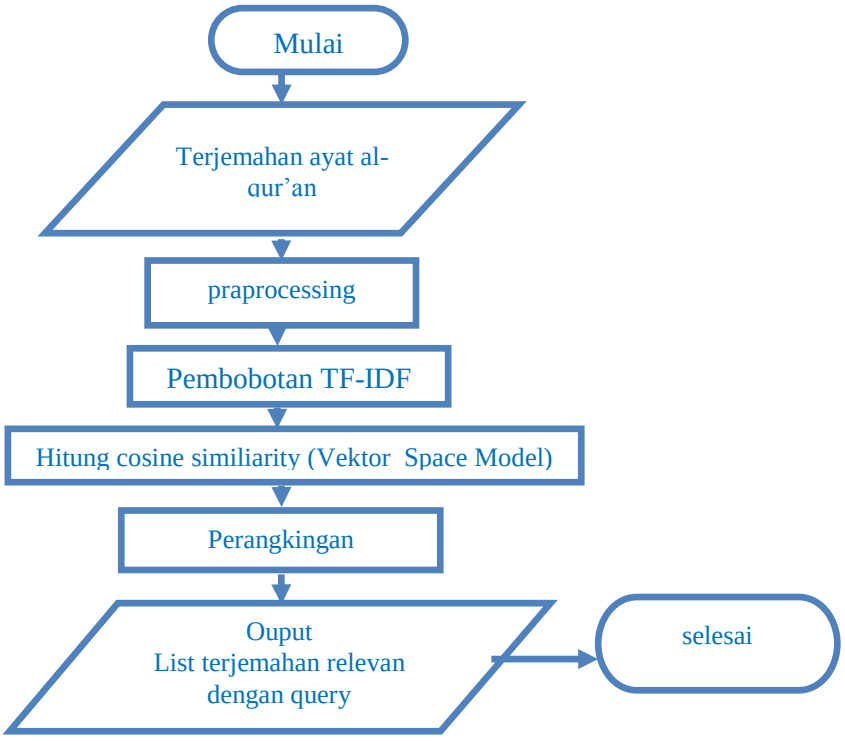
perhitungan frekuensi dan bobot tiap term dengan dokumen. Selanjutnya sistem akan melakukan perhitungan similarity antara dokumen dengan query inputan user. Hasilnya sistem akan meranking nilai simmiliarity dari terbesar. Ditemukan sebuah ayat yang sekiranya relevan dengan query. Ayat ini nantinya akan dijadikan output dari sistem.

Kerja sistem dilanjutkan dengan pencarian dokumen lainnya yang mirip dengan dokumen paling relevan tersebut. Dokumen yang memiliki nilai similiarity paling besar akan diasumsikan sebagai rangking 1 menggunakan algoritma vector space model.Hasil dari sistem nantinya adalah menampilkan beberapa ayat al-qur'an dan terjemahan ayat yang relevan dan memiliki nilai kemiripan dengan query user.

3.2 Perancangan Sistem

Perancangan sistem ini berisi tetang gambaran mengenai diagram alir sistem, data flow diagram, perancangan database dan interface aplikasi.

3.2.1Flowchart Utama Sistem

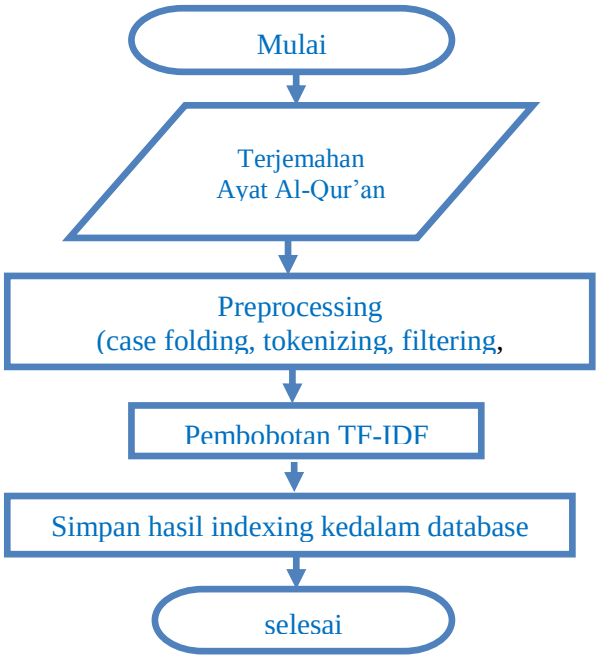


Gambar 3.1 Flowchart utama sistem pencarian ayat al-qur'an.

Berdasarkan gambar.3.1 sistem pencarian ayat al-qu'an :

1. Pertama sistem mengambil dan membaca terjemahan al-qur’an yang tersimpan di dalam tabel_qur’an.
2. Sistem melakukan preprocessing terhadap terjemahan, yang meliputi case-folding, tokenizing, filtering dan stemming, sehingga didapatkan kata *(terms)* dalam bentuk akar yang nantinya dilakukan pengindeksan.
3. Sistem melakukan pembobotan dari tiap terms dalam tiap dokumen berdasarkan rumus (2.4). term dokumen disimpan dalam database.
4. User menginputkan query, kemudian sistem melakukan proses preprocessing hingga didapatkan *term query*, dan dilakukan pembobotan terhadap *terms query*. Hasilnya disimpan dalam database.
5. Sistem malakukan perhitungan kemiripan *(simmiliarity)* antar *terms query* dengan *term* dalam dokumen menggunakan persamaan rumus (2.5) yang kemudian di ranking berdasarkan nilai simmiliarity tertinggi.
6. Sistem mengambil abstrak yang ranking 1(nilai simmiliairty tertinggi) untuk direpresentasikan dalam proses *vector space model*.

Sistem mengeluarkan output berupa list ayat al-qur’an dan terjemahannya.

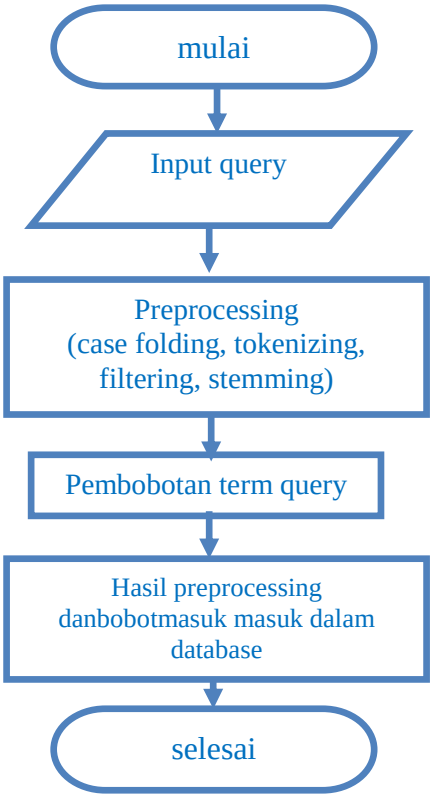


Gambar 3.2 Proses Indexing

Berdasarkan **Gambar 3.2** proses indexing, dilakukan beberapa tahapan :

1. Proses petama dalam sistem dimulai dengan membaca isi terjemahan ayat Al-qur'an yang disimpan dalam database sebagai *corpus*
2. system melakukan preprocessing terhadap *corpus* tersebut, yaitu case folding, tokenizing, filtering dan stemming. Hasil proses ini berupa termHasil proses tersebut kemudian dilakukan proses pembobotan tiap term. Bobot tiap term terjemahan ayat Al-qur'an disimpan dalam tabel database.

Berikutnya Tahapan formulasi query. **Gambar 3.3** merupakan gambaran flowchart tahapan formulasi query. Dalam tahapan ini dilakukan tahapan proses:



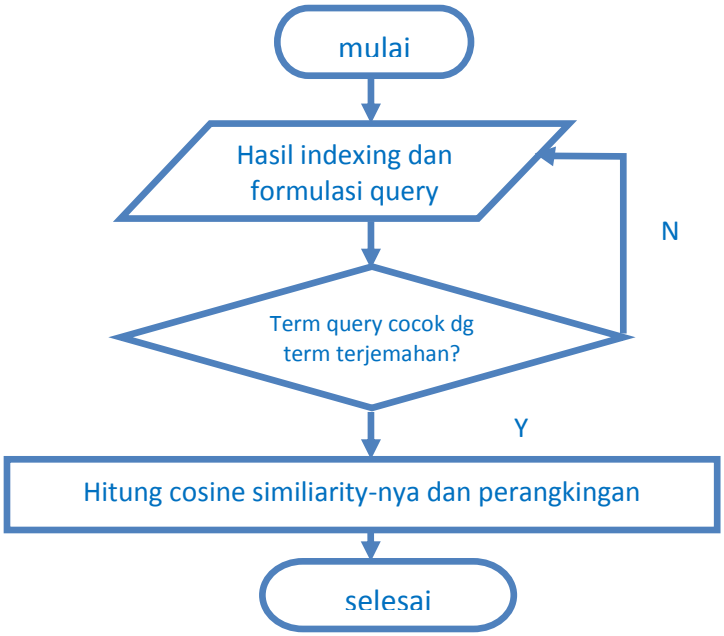
Gambar 3.3 Flowchart formulasi query

Berdasarkan Gambar 3.3 flowchart formulasi query yaitu :

1. User menginputkan query berupa kata berbahasa Indonesia.
2. Aplikasi melakukan preprocessing terhadap query tersebut, yaitu case folding, tokenizing, filtering dan stemming.

3. Hasil proses ini berupa daftar term query tersebut kemudian dilakukan proses pembobotan tiap term query dan bobotnya disimpan dalam tabel database.

Selanjutnya tahapan pencarian, merupakan tahap terakir. Dalam tahapan ini terdapat proses: Sistem mengambil database hasil indexing dan formulasi query, kemudian melakukan pencocokan term query dengan term terjemahan ayat Al-qur'an. Jika term query cocok dengan term terjemahan maka sistem melakukan perhitungan simmiliarity antar query dengan terjemahan ayat al-qur'an. Sistem mendapatkan hasil nilai simmiliarity yang kemudian merankingnya berdasarkan nilai terbesar. Sistem mengambil terjemahan al-qur'an yang bernilai cosine tertinggi, nantinya direpresentasikan sebagai sebuah cluster 1 dalam proses *vector space mode*. Sistem melakukan pencarian kembali terjemahan ayat al-qur'an lainnya yang mirip dengan terjemahan ayat al-qur'an dalam cluster tersebut menggunakan *vector space model*. Sistem mengambil hasil *vector space model* yaitu anggota rangking 1 untuk dijadikan output. Sistem mengeluarkan output berupa list ayat al-qur'an dan terjemahan.



Gambar 3.4 Flowchart Pencarian

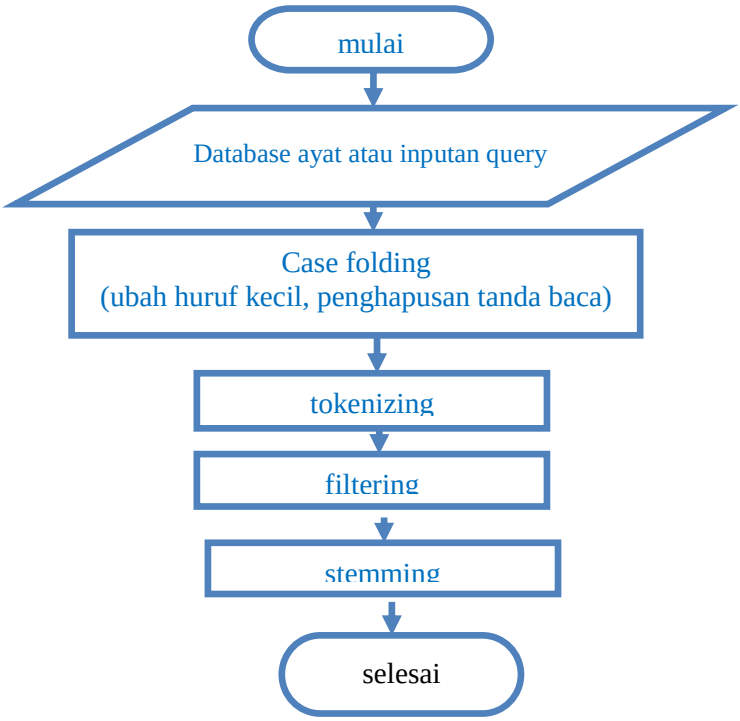
Tahapan yang terakhir, tahapan pencarian. **Gambar 3.4** merupakan gambaran flowchart tahapan formulasi query. Dalam tahapan ini terdapat proses:

1. Sistem mengambil database hasil indexing dan formulasi query, kemudian melakukan pencocokan term query dengan term terjemahan ayat al-Qur'an.
2. Jika term query cocok dengan term terjemahan ayat maka sistem melakukan perhitungan simmiliarity antar query dengan terjemahan ayat Al-Qur'an.
3. Sistem mendapatkan hasil nilai simmiliarity yang kemudian merankingnya berdasarkan nilai terbesar.

Dari diagram alir sistem diatas terdapat beberapa sub proses yang dilakukan:

3.2.2 Teks Prepocesing

Karakteristik teks dalam proses *text mining* pada umumnya adalah memiliki dimensi yang tinggi, terdapat nois pada data, dan terdapat struktur teks yang tidak baik. Agar proses analisis teks dan pencarian padanan dengan *cosine similarity* dapat lebih optimal maka perlu dilakukan proses dari tahapan awal *text mining* ini. Tahapan awal yang dilakukan adalah *preprocessing*, langka-langka yang akan dilakukan adalah *case folding* kemudian dilakukan proses *tokenizing.* Diagram alir tahapan *text prepocesing* di tunjukkan pada **gambar 3.5.**



Gambar 3.5 Dagram Alir Proses Preprocessing

Berdasarkan **Gambar 3.5** dalam tahapan perprosesing yaitu:

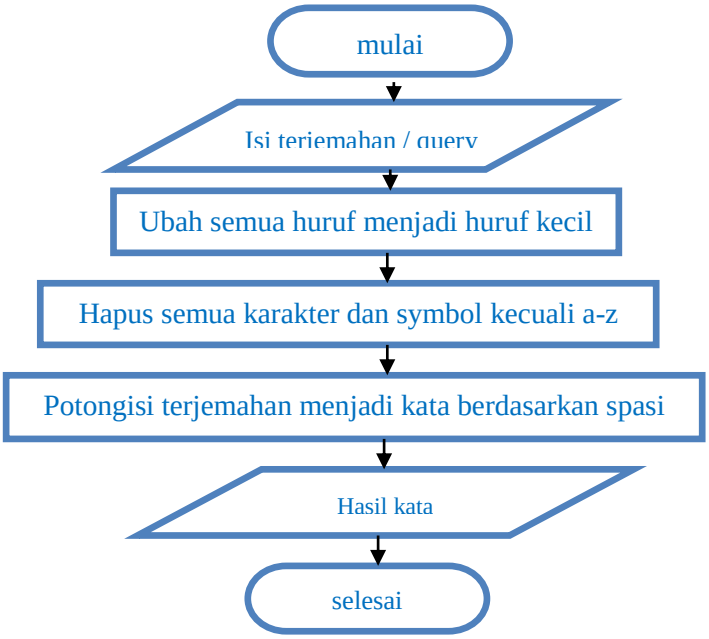
1. Pengolahan terhadap terjemahan ayat dan inputan query didatabase.
2. Proses case folding, yaitu proses penghilangan karakter atau tanda baca dan juga pengubahan huruf menjadi huruf kecil dalam terjemahan ayat Al-Qur'an maupun query
3. Proses *tokenizing* yaitu memecah kalimat menjadi *terms*.
4. *Filtering* adalah pemilihan kata dan pencocokan kata terhadap stopword kemudian menghilangkan kata yang ada dalam stopword.
5. *Stemming* proses penghilangan awalan dan akhiran atau bisa disebut juga kata-kata yang muncul di dalam terjemahan sering mempunyai banyak varian morfologik.

3.2.2.1 Case Folding

Proses case folding, yaitu proses penghilangan karakter atau tanda baca dan juga pengubahan huruf menjadi huruf kecil dalam terjemahan ayat maupun query.

3.2.2.2 Tokenizing

Proses tokenizing, merupakan proses pemecahan kalimat dokumen abstrak menjadi kata. Begitu juga dengan query inputan user. Hasil dari proses ini akan dilakukan filtering. Proses ini digambarkan dalam flowchart **Gambar 3.6.**



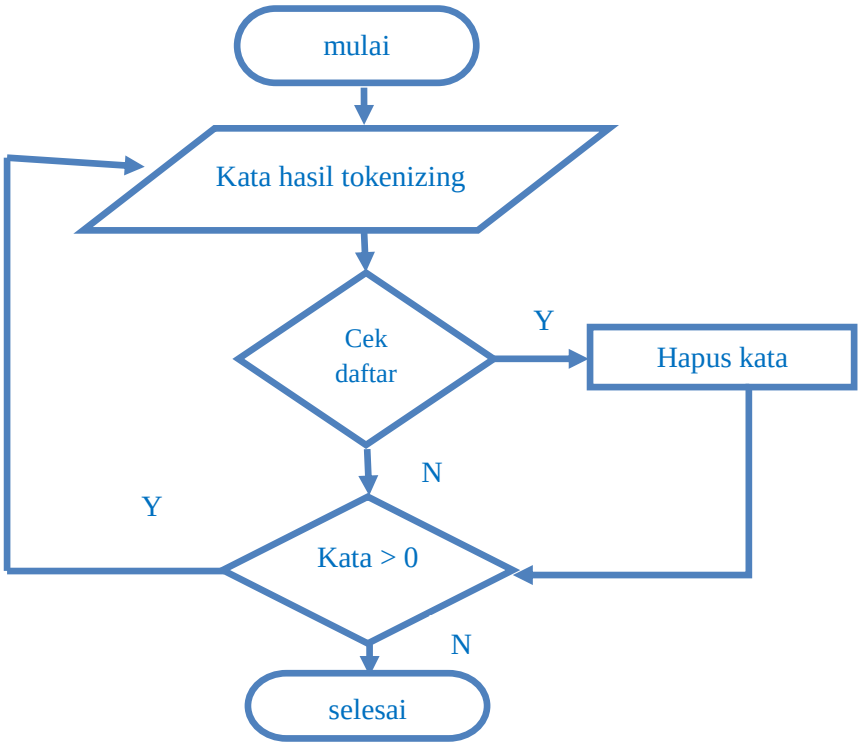
Gambar 3.6 Dagram Alir case folding dan tokenizing

Pada tahap case folding dan tokenizing diatas proses yang dilakukan adalah sebagai berikut:

1. Memasukan dokumen kedalam aplikasi
2. Mengubah semua karakter dalam dokumen menjadi huruf kecil
3. Menghapus semua karakter dan simbol selain huruf abjad a-z
4. Memotong dokumen menjadi kata berdasarkan spasi
5. Memunculkan hasil kata

3.2.2.3 Filtering

Tahap filtering adalah tahap mengambil kata-kata penting dari hasil token Algoritma yang dipakai adalah stoplist (membuang kata yang kurang penting) atau wordlist (menyimpan kata yang penting). Stoplist/stopword adalah kata kata yang tidak deskriptif yang dapat dibuang dalam pedektan bag-of-word. Dalam tahap pembuangan kata kata yang tidak penting adalah kata hasil parsing dicek dengan kamus (kumpulan kata) stopword. Jika kata parsing ada yang sama dengan stopword maka kata akan dibuang/dihapus. Diagram alir filtering dapat dilihat pada **Gambar 3.7.**



Gambar 3.7 Diagram Alir dari Proses Filtering

Pada tahap filtering proses yang dilakukan adalah sebagai berikut :

1. Mengambil kata hasil dari proses case folding dan tokenizing.
2. Mengecek kata di dalam tabel stopword, jika ada kata yang sama dengan tabel stopword maka akan dihapus, tetapi jika tidak ada maka proses berakhir.

3.2.2.4 Stemming

Stemming adalah pengembalian kata dasar dari kata kata yang telah mengalami imbuhan, sisipan dan awalan. Merupakan proses pemetaan dan penguraian berbagai bentuk (variants) dari suatu kata menjadi kata dasarnya (system). Proses ini disebut juga sebagai conflation. Proses stemming secara luas sudah digunakan di dalam information retrieval (pencarian informasi) untuk meningkatkan kualtas informasi yang didapatkan. Kualitas informasi yang dimaksud misalnya untuk mendapatkan hubungan antara varian kata yang satu dengan yang lainnya.

Kumpulan kata hasil proses sebelumnya yaitu filtering dicek satu persatu apakah kata tersebut terdapat dalam kamus kata yang disimpan didatabase ataukah tidak. Jika kata tersebut terdapat dalam kamus kata maka kata tersebut sudah termasuk bentuk dasar, tapi jika tidak terdapat dalam kamus kata maka akan dilakukan pemotongan imbuhan atau akhiran berdasarkan algoritma stemming Nazief-Andriani, Algoritma ini merupakan hasil penelitian internal UI (Universitas Indonesia) dan tidak dipublish secara umum (Nazif, 1996).

Algoritma ini merupakan gabungan antara algoritma menghilangkan imbuhan dan brute force stemming. Namun algoritma ini mempunyai dua masalah, yang pertama kemampuannya tergantung dari besarnya database kata dasar, dan yang kedua, hasil stemming tidak selalu optimal untuk apliksi information retrieval. dapat pula dilihat pada **gambar 3.7** yang menunjukkan diagram alir algoritma stemming nazief-andriani. Dalam hal ini maka kelengkapan daftar kamus kata sangat

menentukan ketepatan perubahan kata ke bentuk akarnya, disamping algoritma yang digunakan.

[[[DP+]DP+]DP+]kata dasar[[+DS][+PP][+P]]

DP : Awalan

DS : Akhiran

PP : Kata ganti kepunyaan

P : Partikel

Tanda kurung menandakan imbuhan bersifat opsional. Langkah-langkah algoritma Nazief dan Andriani dalam men-stemming sebuah kata telah dijelaskan pada bab II. Berikut ini contoh penerapan algoritma naxief dan andriani

Berkepenting**an**>

Berkepenting + **an (DS) >**hapus akhiran –**an**

Ber (DP) + **ke**penting > hapus awalan **–ber**

Ke (DP) + penting > hapus awalan –**ke**

Penting > root word

Mempertimbang**kannyapun**>

Mempertimbang**kannya** + **pun** (P) > hapus partikel –**pun**

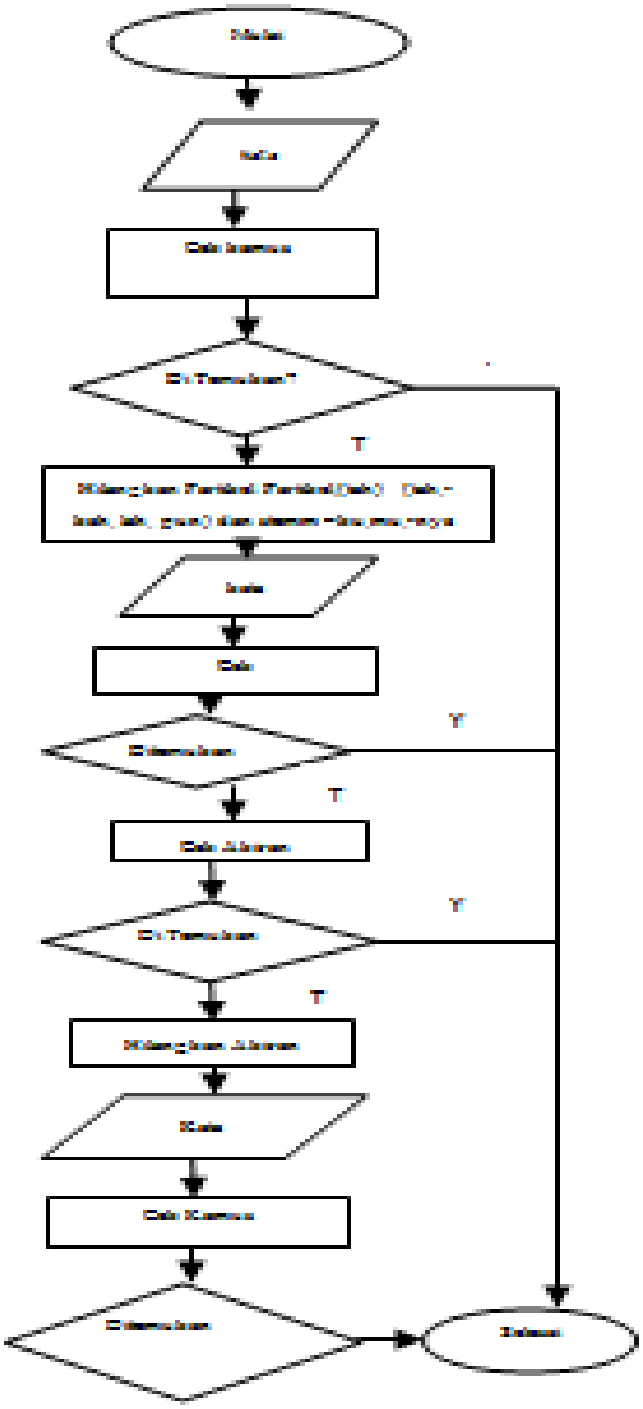
Memeprtimbang**kan** + **nya** (pp) > hapus possessive pronoun **–nya**

Mempertimbang + **kan (**DS) > hapus derivation suffix **–kan**

Mem (DP) + **per**timbang > hapus derivation prefix **-mem**

Per (DP) + timbang > hapus derivation prefix **–per**

Timbang > root word



Gambar 3.8 Diagram AlirAlgoritma Stemming Nazief-Andriani

Proses tahapan *stemming* nazief dan andriani yang terjadi pada gambar 3.8:

1. Kata dicari di dalam daftar kamus. Bila kata tersebut ditemukan di dalam kamus maka dapat diasumsikan kata tersebut adalah kata dasar sehingga algoritma dihentikan.
2. Bila kata di dalam langkah pertama tidak ditemukan di dalam kamus, maka diperiksa apakah sufiks tersebut yaitu sebuah partikel (“-tah”,”-lah”,”-pun” atau “-kah”). Bila ditemukan maka partikel tersebut dihilangkan.
3. Pemeriksaan dilanjutkan pada kata ganti milik(“-ku”,”-mu”,”-nya”).bila ditemukan maka kata ganti tersebut dihilangkan.
4. Memriksa akhiran (“-i”,”-an”,”-kan”). Bila ditemukan maka akhiran tersebut dihilangkan.

   Hingga langkah ke-4 dibutuhkan ketelitian untuk memeriksa apakah akhiran “-an” merupakan hanya bagian dari akhiran “-kan” dan memeriksa lagi apakah partikel (“-tah”,”-lah”,”-pun” atau “-kah”) dan kata ganti milik (“-ku”,”-mu”,”-nya”) yang telah dihilangkan pada langkah 2 dan 3 bukan merupakan bagian dari kata dasar.
5. Memeriksa awalan (“se-“,”ke-“,”di-“,”te-“,”be-“,”pe-“,”me-“). Bila ditemukan, maka awalan tersebut dihilangkan. Pemeriksaan dilakukan dengan berulang mengingat adanya kemungkinan multi-prefik.Langkah ke-5 ini juga membutuhkan ketelitian untuk memeriksa kemungkinan peluluhan awalan, perubahan prefix yang disesuaikan dengan huruf awal kata dan aturan kombinasi prefix-sufix yang diperbolehkan.
6. Setelah menyelesaikan semua langkah diatas dengan sukses, maka algoritma akan mengembalikan kata dasar yang ditemukan.

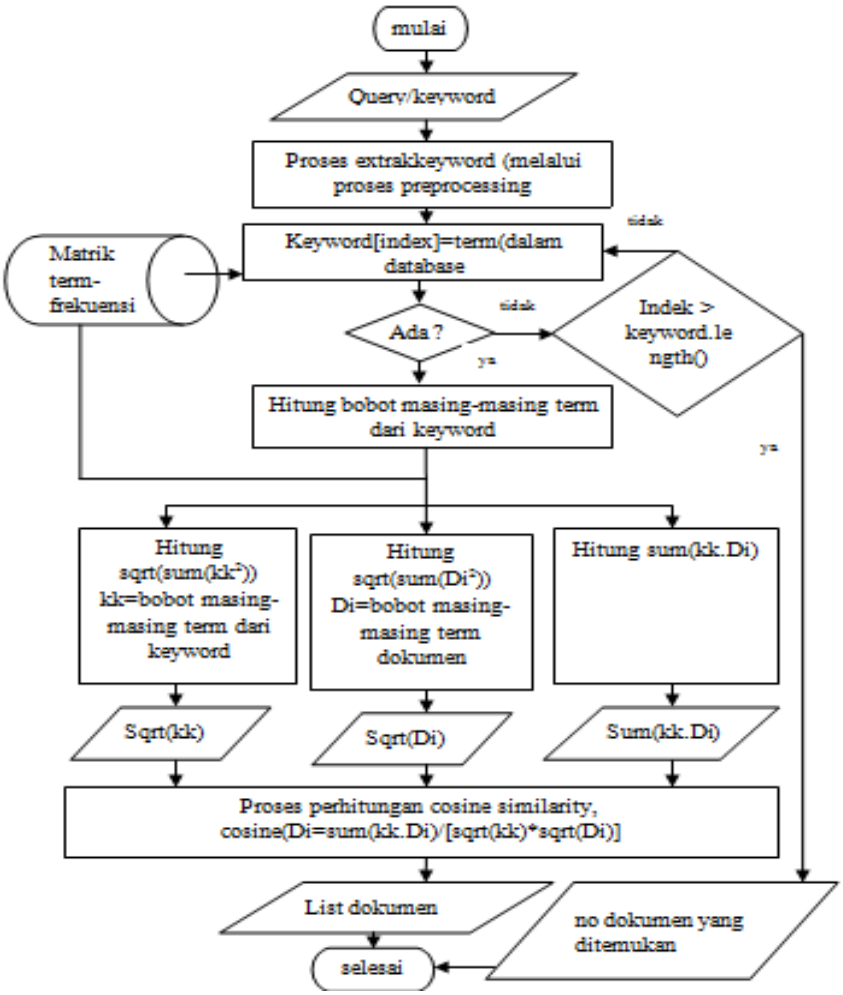
3.2.2.5 Pembobotan Dengan Algoritma TF-IDF(Term Frequency-Inversed DocumentFrequency)

Algoritma pembobotan TF_IDF merupakan algoritma pembobotan yang merupakan perpaduan antara pembobotan local, yaitu *term frequency* (idf) dengan pembobotan global yaitu *inverse document*

frequency (idf). Pembobotan TF dilakukan dengan menghitung jumlah kemunculan kata dalam terjemahan ayat , sedangkan pembobotan TF-IDF merupakan nilai *log* dari jumlah terjemahan ayat keseluruhan dibagi dengan jumlah ayat yang mengandung suatu kata, dan nilai dari pembobotan TF –IDF diperoleh dengan mengalikan nilai TF dengan IDF yang telah diperoleh. Dari bobot TF-IDF tersebut dilakukan *shorting*, bobot yang lebih tinggi memiliki tingkat kemiripan yang lebih dibandingkan dengan bobot yang rendah. Meski dari pengurutan bobot ini dapat diketahui padanan terjemahan ayat yang sesuai, hanya saja masih dimungkinkan adanya terjemahan ayat yang memiliki bobot yang sama, sehingga untuk meningkatkan tingkat akurasi kmiripan terjemahan ayat perlu dilakukan tahap selanjutnya, yaitu representasi model ruang vector dan ukuran kemiripan cosine similarity.

3.2.2.6 Vektor Space model

Setelah di lakukan proses pengurutan terhadap bobot (w) TF-IDF di lakukan dengan mengukur nilai kemiripan antar terjemahan ayat, tapi sebelumnya perlu menggunakan algoritma *vector space model* antar terjemahan, sebagaimana yang telah di tunjukkan **gambar 3.9**



Gambar 3.9 Diagram Alir Proses Algoritma Vektor Space Model

Gambar 3.10 Diagram Alir algoritma *vector space model*

Gambar 3.9 Diagram Alir Proses Algoritma vector space model

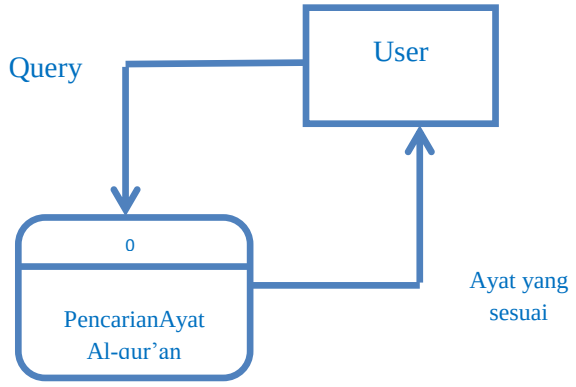
3.2.2.7 Cosine Similarity

Ukuran kemiripan/similaritas yang digunakan dalam penelitian ini adalah *cosine similarity*. Perhitungan jarak cosinus antara veltor query dengan vector tiap ayat al-qur'an dalam database dilakukan dengan rumus yang terdapat pada **persamaan 2.6** di bab sebelumnya. Hingga diperoleh nilai kemiripan tersebut antara 0 sampai 1 di *shorting* mulai nilai terbesar karena semakin emndekati 1 berarti tingkat kemiripan lebih tinggi.

3.2.3 Data Flow Diagram

3.2.3.1 Context Diagram Sistem

Context diagram pada **gambar 3.10** ini merupakan gambaran sistem atau aplikasi secara garis besar yang disebut sebagai top level. Dimana user menginputkan query atau keyword ke dalam sistem pencarian, query inilah yang akan diproses dan kemudian akan mendapatkan hasil berupa ayatdanterjemahan al-qur'an yang relevan, didalamnya terdapat informasi berupater jemahan dan Ayat al-qur'anyang relevan dan mirip dengan query inputannya sebagai hasil dari proses pencarian tersebut.



Gambar 3.10 Context Diagram Sistem Pencarian Ayat Al-qur'an

3.2.3.2 Diagram Berjenjang

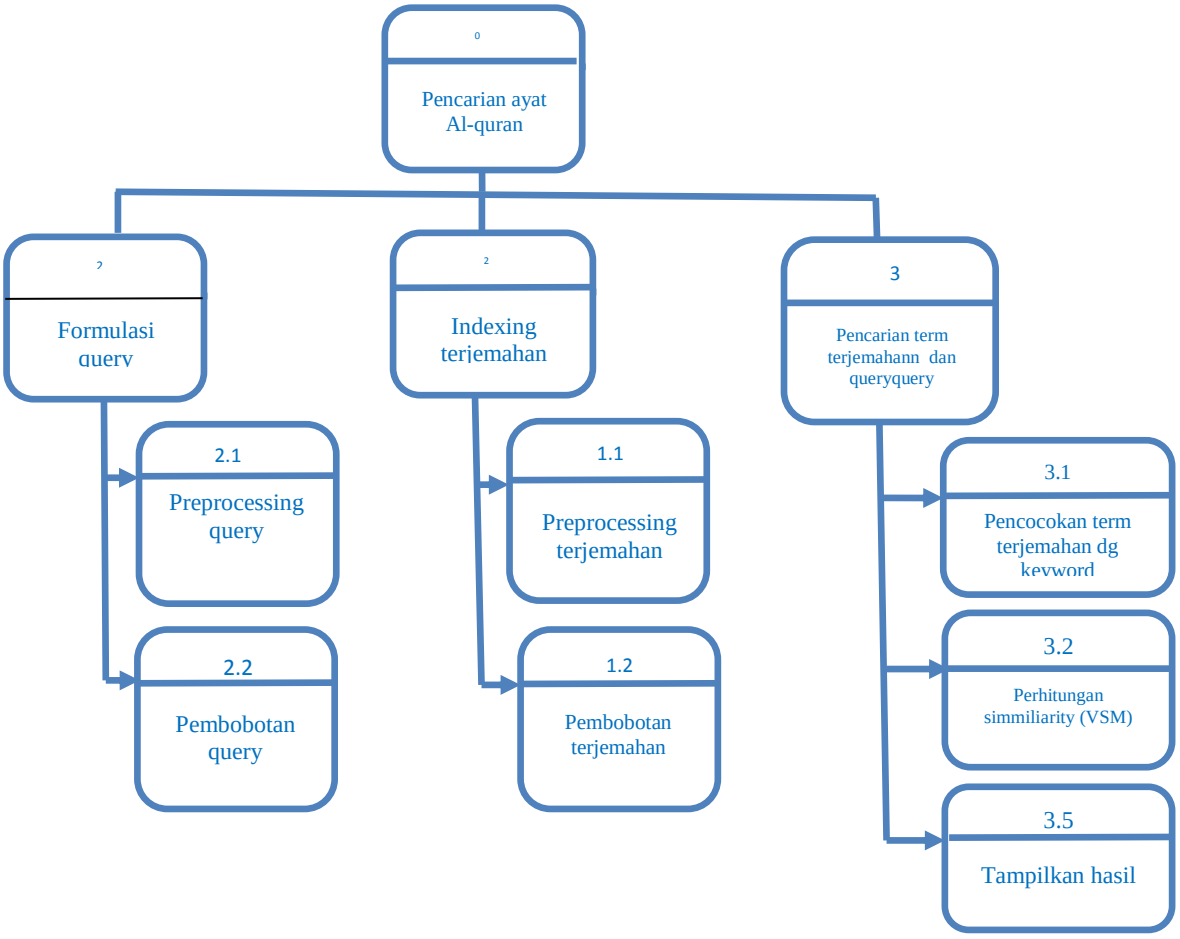
Gambar 3.11 dibawah ini merupakan gambar diagram berjenjang dari sistem pencarian ayat al-qur'an yang terdiri dari 3 level, yaitu :

- Top level : Sistem Pencarian Ayat Al-qur'an secara global
- Level 0 : merupakan hasil break down dari proses global dari sistem yang terdiri dari beberapa sub proses, yaitu:
    - Proses indexing ayat al-qur'an

      Proses indexing terdapat sub proses didalamnya, diantaranya proses preprocessing dan pembobotan
    - Proses formulasi query / keyword yang diinput user

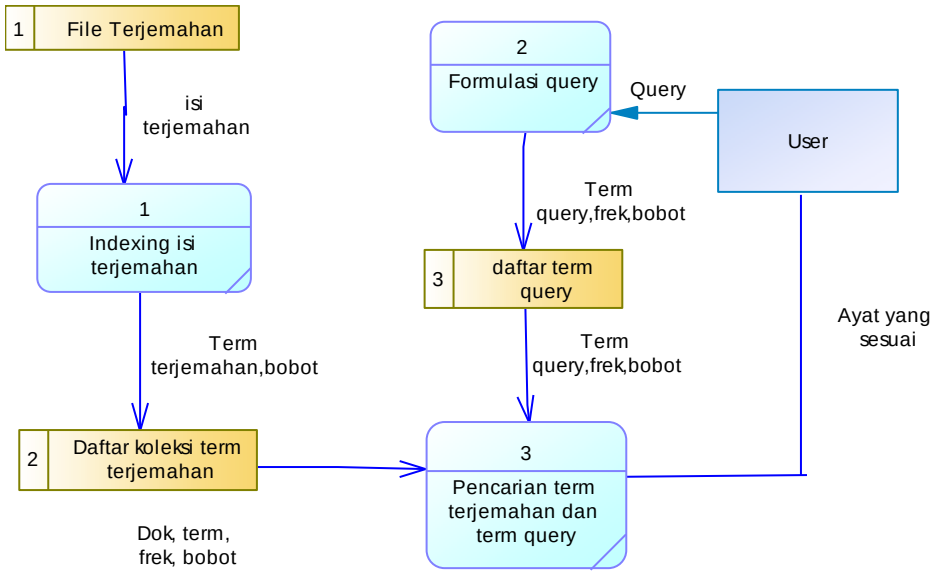
      Proses ini juga mengandung beberapa sub proses, yaitu proses preprocessing query dan pembobotan
    - Proses pencarian

      Dalam proses pencarian terdapat sub proses diantaranya yaitu perhitungan simmiliarity antara dokumen dengan query, perankingan hasil simmiliarity, proses pencarian dokumen mirip dengan *vector space model*.



3.2. Gambar 3.11 Diagram Berjenjang Sistem Pencarian Ayat Al-qur'an

Dari gambaran umum aplikasi pada contex diagram kemudian dijabarkan lebih dalam pada *data flow daiagram (DFD).*Pada DFD level 0 ditunjukkantahapan proses yang dilakukan dalam pencarian yaitu mualai dari:Prosespertama :indexing dokumen isi terjemahan, kedua: pemprosesan query inputan user, ketiga: proses pencarian. Dari proses tersebut user mendapatkan informasi ayat al-qur'an yang sesuai dengan inputan user berupa ayat yang sesuaidenganinputan yang dimasukkan user.

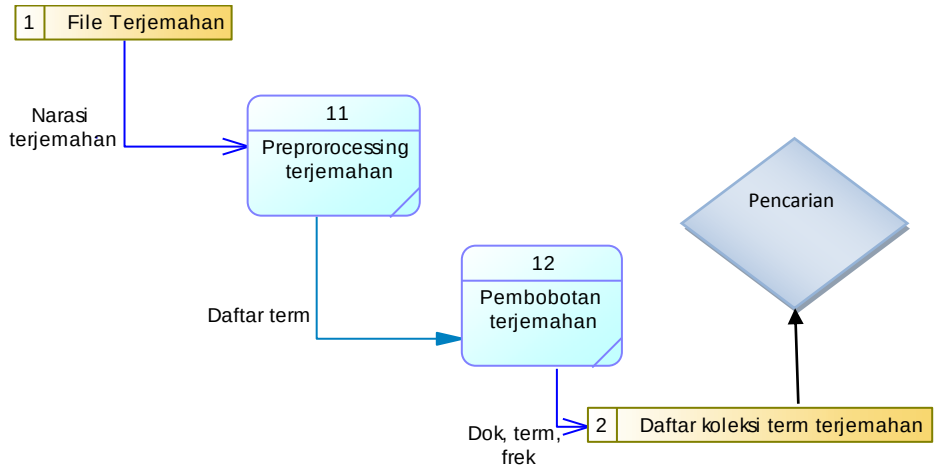


Gambar 3.12 Data Flow Diagram Level 0

Dari tiap-tiap tahapan proses dari **gambar 3.12** dapat dijabarkan lebih detail dalam DFD level 0.

3.2.3.4 Data Flow Diagram Level 1.1 Proses Indexing

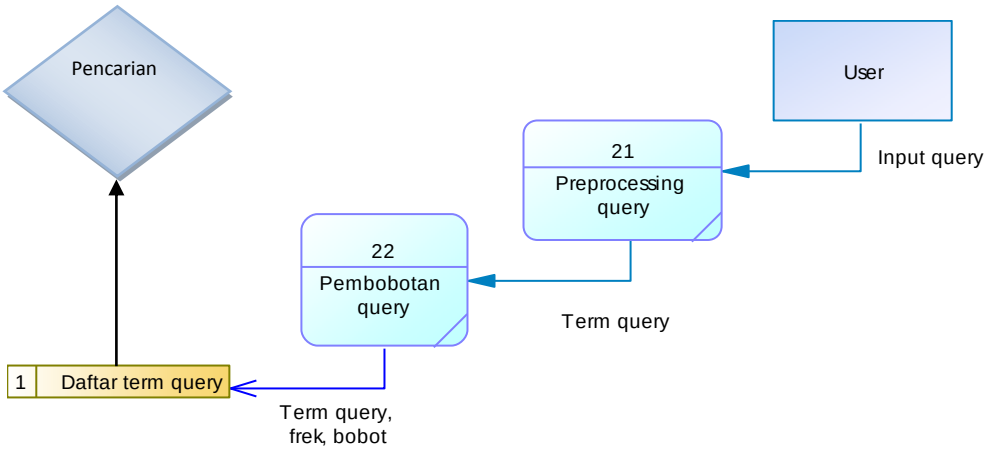
Gambar 3.13 dibawah ini merupakan DFD level 1 dari proses indexing terjemahan yang mana dalam proses ini dilakukan beberapa urutan proses hingga menghasilkan output berupa kumpulan term dan bobot dari tiap term yang akan dilanjut ke proses transformation.



Gambar 3.13 DFD Level 1 Proses Indexing

3.2.3.5 Data Flow Diagram Level 1.2 Proses Formulasi

Pada DFD level 2 proses formulasi query terdapat dua tahap proses yaitu proses preprocessing query sehingga didapatkan daftar term query, lalu dilakukan proses kedua yaitu pembobotan term query, dimana hasilnya akan disimpan dalam sebuah tabel di database untuk proses pencarian. Lihat **gambar 3.14**

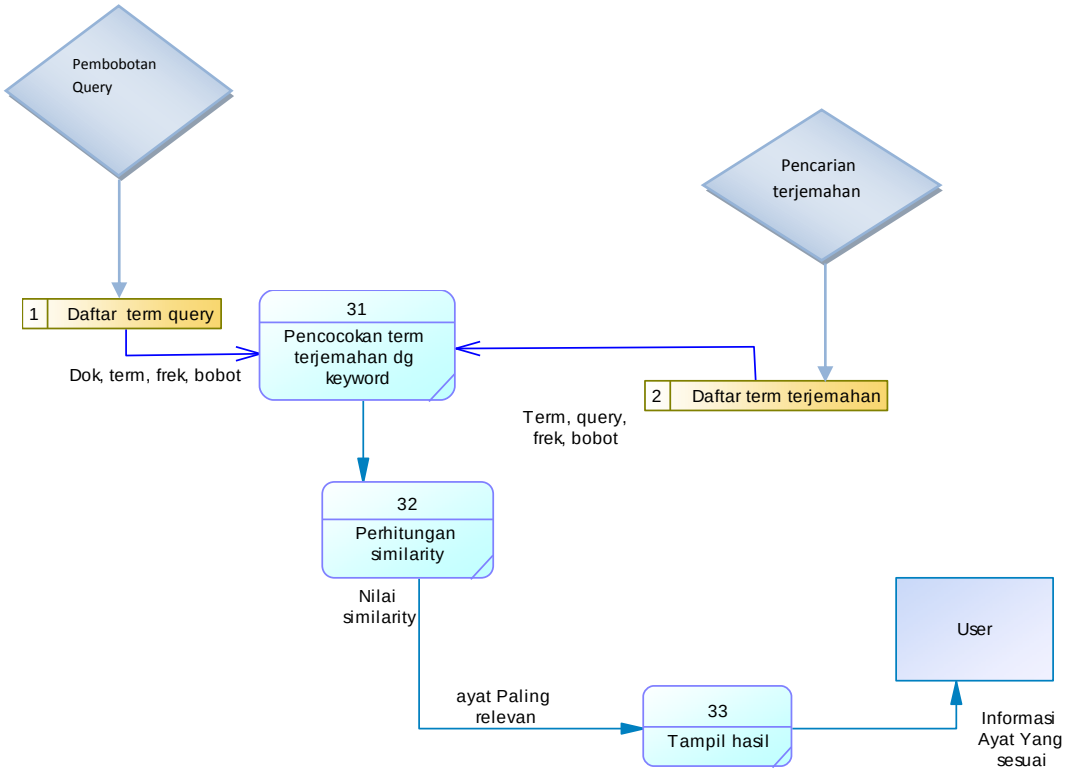


Gambar 3.14 DFD Level 1 Proses Formulasi Query

3.2.3.6 Data Flow Diagram Level 1 Proses Pencarian

Tahapan yang ketiga sebagaimana **gambar 3.15** merupakan tahapan pencarian dokumen yang sekiranya relevan dengan query dan pencarian ayat yang mirip. Daftar term query hasil proses query dilakukan pencocokan dengan

daftar tem hasil indexing dan akan dihitung nilai simmiliarity-annya terhadap koleksi term yang didapat dari tahapan indexing. Kemudian dilakukan perankingan berdasarkan nilai simmiliarity terbesar. Proses pencarian terjemahan dan ayat al-qur'an yang mirip lainnya dilakukan dengan algoritma *vector space model*. Hasil dari tahapan pencarian ini nantinya berupa list terjemahandanayatyang relevan dan terjemahan lain yang mirip.



Gambar 3.15 DFD Level 1 Proses Pencarian

3.2.4 Representasi Data

Query atau kalimat yang diinputkan user: "shalat"

Berikut dibawah ini tahapan-tahapan prosesya :

A. Kasus

Sebagai contoh dilakukan pencarian ayat yang sesuai denga teks Q, Ayat yang ada yaitu A1, A2, dan A3. Teks Q dan ayat-ayat tersebut diproses sehingga diperoleh urutan ayat yang sesuai dengan teks Q.

Quey :shalat

Ayat1 : Kamu mohonlah pertolongan dalam kesabaran dan shalat karena ia adalah sungguh berat kecuali, kepada orang orang yang merendah;

Ayat **2** : Dan dirikanlah shalat dan berikanlah zakat dan berlututlah bersama orang-orang yang berlutut.

Ayat 3 :Hanya Engkaulah yang kami sembah, dan hanya kepada Engkaulah kami meminta pertolongan.

Text Processing dan text Transformation

Masing-masing Q, A1, A2, dan A3 diproses sesuai dengan urutan tahap text mining, Berikut ini contoh hasil dari tiap proses dalam tahap *prepocessing dan transformation*, dimana tahap (1) merupakan *case folding, (2) tokenizing, (3)filtering, (4) stemming*

Quey :shalat

Ayat 1

1. Tahap *case folding* : kamu mohonlah pertolongan dalam kesabaran dan shalat karena ia adalah sungguh berat kecuali kepada orang orang yang merendah

2.Tahap*tokenizing:*|kamu||mohonlah||pertolongan||dalam||kesabaran||dan||shalat||karena||ia||adalah||sungguh||berat||kecuali||kepada||orang||orang||yang| |merendah|

3.*Tahap Filtering* : mohonlah pertolongan kesabaran shalat berat orang orang merendah

4. *Tahap stemming* : mohon tolong sabar shalat berat orang orang rendah

Ayat 2 : 1. Tahap *case folding :* dan dirikanlah shalat dan berikanlah zakat dan berlututlah bersama orang-orang yang berlutut

2**.** Tahap *tokenizing:*

|dan||dirikanlah||shalat||dan||berikanlah||zakat||dan||berlututlah|
|bersama||orang||orang||yang||berlutut|

3**.***Tahap Filtering* : dirikanlah shalat zakat berlututlah orang orang berlutut

4**.***Tahap stemming* :dirikan shalat zakat lutut orang orang lutut

Ayat 3 : 1**.** Tahap *case folding :* hanya engkaulah yang kami sembah dan hanya kepada engkaulah kami meminta pertolongan

2. Tahap *tokenizing*:
   |hanya||engkaulah||yang||kami||sembah||dan||hanya||kepada||engkaulah||kami||meminta||pertolongan|
3. *Tahap Filtering* : sembah meminta pertolongan
4. Tahap stemming :sembah minta tolong

Tahapan dalam menghitung term dan query dalam perhitungan Tf-Idf :

1. Term frequency (tf) merupakan frekuensi kemunculan term (t) pada dokumen (d).
2. Document frequency (df) adalah banyaknya dokumen dimana suatu term (t) muncul.
3. Menghitung invers document frequency (idf) menggunakan rumus persmaan (2.3).

Dibawah ini **tabel 3.1** merupakan tabel perhitungan TF-IDF tiap termdalam masing-masing dokumen

Tabel 3.1 Tabel Perhitungan TF-IDF

term		**Tf**					
	Q	**A1**	**A2**	**A3**	**df**	**d/df**	**idf**
mohon	0	1	0	0	1	3	1,477121

Lanjutan **Tabel 3.1** Tabel Perhitungan TF-IDF

term		**Tf**					
	Q	**A1**	**A2**	**A3**	**df**	**d/df**	**idf**
tolong	0	1	0	1	2	1,5	1,176091
sabar	0	1	0	0	1	3	1,477121
shalat	1	**2**	1	0	2	1,5	1,176091
berat	0	1	1	0	2	1,5	1,176091
rendah	0	1	1	0	2	1,5	1,176091
dirikan	0	0	1	0	1	3	1,477121
zakat	0	0	1	0	1	3	1,477121
lutut	0	0	2	0	1	3	1,477121
sembah	0	0	0	1	1	3	1,477121
minta	0	0	0	1	1	3	1,477121

Selanjutnya dilakukan perhitungan pembobotan masing-masing term dokumen serta term query dari tiap dokumen menggunakan rumus (2.4)

Tabel 3.2 Tabel Perhitungan Pembobotan

W			
WQ	**A1**	**A2**	**A3**
0	1,477121	0	0
0	1,176091	0	1,176091
0	1,477121	0	0

Lanjutan **Tabel 3.2** Tabel Perhitungan Pembobotan

W			
WQ	**A1**	**WQ**	**A1**
1,17609126	2,352183	1,176091	0
0	1,176091	1,176091	0
0	2,352183	2,352183	0
0	1,176091	1,176091	0
0	0	1,477121	0
0	0	1,477121	0
0	0	2,954243	0
0	0	0	1,477121
0	0	0	1,477121

Setelah didapatkan bobot masing-masing term dokumen dan query. Selanjutnya dilakukan perhitungan simmiliarity antar term dokumen dengan term query menggunakan persamaan rumus cosine simmiliarity (2.5).

Tabel 2.10 perhitungan manual dot product antara document dengan query $d_i \bullet q$:

Tabel 3.3 Tabel Perhitungan dot product document dengan query

Term	**wA1*wq**	**wA2*wq**	**wA3*wq**
Mohon	0	0	0
Tolong	0	0	0
Sabar	0	0	0

Lanjutan **Tabel 3.3** Tabel Perhitungan dot product document dengan query

Term	**wA1*wq**	**wA2*wq**	**wA3*wq**
shalat	2,766381	1,3831906	0
berat	0	0	0
orang	0	0	0
rendah	0	0	0
dirikan	0	0	0
zakat	0	0	0
lutut	0	0	0
sembah	0	0	0
minta	0	0	0
$d_i \bullet q$	2,766381	1,3831906	0

Tabel 3.4 Tabel Perhitungan vector document query dan cosine

term	**(WA1^2)**	**(WA2^2)**	**(WA3^2)**	**(WQ^2)**
mohon	2,181887201	0	0	0
tolong	1,38319065	0	1,383191	0
sabar	2,181887201	0	0	0
shalat	5,532762599	1,383191	0	1,383191

Lanjutan Tabel 3.4 Tabel Perhitungan vector document query dan cosine

Term	**(WA1^2)**	**(WA2^2)**	**(WA3^2)**	**(WQ^2)**
Berat	1,38319065	1,383191	0	0
Orang	5,532762599	5,532763	0	0
Rendah	1,38319065	1,383191	0	0
Dirikan	0	2,181887	0	0
Zakat	0	2,181887	0	0
Lutut	0	8,727549	0	0
Sembah	0	0	2,181887	0
Minta	0	0	2,181887	0
$\sum (Wd_i^2, Wq^2)$	19,57887155	22,77366	5,746965	1,383191
$\sqrt{\sum (Wd_i^2, Wq^2)}$ $\|\|d_i\| \times \|\|q\|\|$	4,424801865	4,772175	2,397283	1,176091
$\frac{d_i \bullet q}{\|d_i\| \times \|q\|}$	5,203970797	5,612514	2,819423	0
Cosine	0,531590473	0,246448	0	0

Nilai cosine dari tiap dokumen terhadap query yang selanjutnya dilakukan **perankingan** berdasarkan nilai cosine tertinggi, seperti pada tabel dibawah ini

Tabel 3.5 Perangkingan

Rangking	Dokumen	similariti
1	Ayat 1	0,531590473
2	Ayat 2	0,246448
3	Ayat 3	0

Tahap pencarian ayat yang mirip dengan dokumen yang memiliki similiaritas tertinggi dengan algoritma vektor space model.

Dari proses sebelumnya didapatkan sebuah dokumen yang memiliki nilai similiarity tertinggi yaitu dokumen 1. Selanjutnya dokumen tersebut akan direpresentasikan sebagai cluster pertama dalam proses vektor space model.

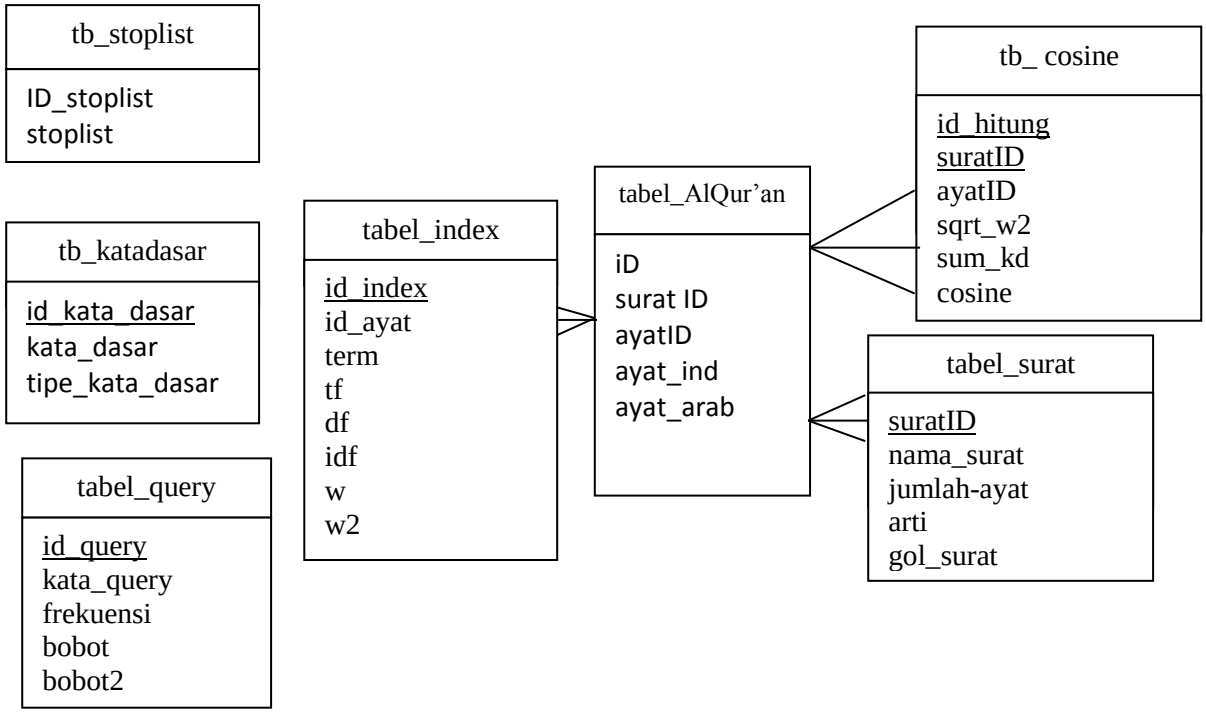
Setelah mengalami proses vektor space model didapatkan:

Output dokumen relevan dan yang mirip: Ayat 1, Ayat2

Dokumen kurang relevan (tidak dijadikan output) : Ayat 3

3.2.5 Desain database

Berikut ERD desain database yang merepresentasikan relasi antar entity-nya.



Gambar 3.16 Entity Relational Diagram Sistem Pencarian Ayat Al-Qur'an

Berikut tabel-tabel yang digunakan dalam pembangunan sistem pencarian Ayat Al-qur'an :

a.Tabel stopword (stoplist)

Tabel stopword ini digunakan untuk penyimpanan kata kata dengan frekuensi kemunculan tinggi atau kata yang irrelevant dengan dokumen, seperti: “yang”,”di”,”dan”. Terdapat satu field dalam table ini, yaitu colom kata_stopword yang digunakan untuk penyimpanan kata yang akan dihilangkan pada proses filtering. Sehingga dapat dilakukan pengecekan terhadap tiap kata dari input dokumen, jika kata tersebut ditemukan dalam table stopword ini maka kata tersebut tidak diproses ke tahap selanjutnya.data ini diperoleh dari situs http://static.hikaruyuuki.com/wp-content/uploads/stopword_list_tala.txt. Struktur table ini dapat dilihat pada **table 3.1**

Tabel 3.6 Struktur table stopword

Nama field	Tipe	Keterangan
ID_stoplist stoplist	Int (10) varchar (50)	

b. Tabel kata dasar

Tabel kata dasar berisi kumpulan kata dasar berdasarkan kamus bahasa Indonesia. Dalam table ini juga disertai keterangan kata dasar tersebut, sehingga diketahui kata yang berupa kata kerja atau kata yang berupa kata sifat. Table ini digunakan pada proses stemming untuk mengecek apakah kata tersebut sudah merupakan kata dasar. Jika tidak terdapat dalam daftar kata dasar maka akan dilakukan proses stem terhadap kata tersebut. Data ini diperoleh dari situs http://bahtera.org/.Struktur table ini dapat dilihat pada **table 3.2**

Tabel 3.7 Struktur table kata_dasar

Field	Type	Size	Keterangan
Id_ktdasar	Int	11	ID dari kata dasar
Kata_dasar	Varchar	20	Kata dalam bentuk akar
Tipe_kata_dasar	Varchar	20	Tipe dari kata dasar tersebut

c. Tabel Al-Qur'an

Tabel tabel_alquran berfungsi untuk penyimpanan daftar ayat dan terjemahan. Dalam tabel ini terdapat id_ayat yang merupakan primary key. Sedangkan field id sebagai penunjuk urutan skripsi dimasukkan ke sistem, id ini bersifat incremental. Tabel tabel_alqur'an ini memiliki struktur seperti terlihat pada tabel 3.3 dibawah ini :

Tabel 3.8 Struktur table AlQur'an

Field	Type	Size	Keterangan
Id	Int	11	Incremental
suratID	Int	11	ID dari surat
ayatID	Int	11	ID dari ayat
Ayat_ind	text		Berisi terjemahan ayat alqur'an
Ayat-arab	text		Berisi ayatalquran

d. tabel index

Tabel tabel_index berfungsi untuk penyimpanan daftar term terjemahan hasil proses preprocessing, nilai TF (term frequency)masing-masing term dari tiap ayat, nilai IDF(inverse document frequency) dari perhitungan mrnggunakan rumus(2.3) serta nilai bobot tiap term yang didapat dari perhitungan perkalian tf dan idf rumus (2.4). Tabel ini memiliki struktur sebagai berikut :

Tabel 3.9 Struktur tabel tabel_index

Field	Type	Size	Keterangan
Id_index	Int	11	Incremental
suratID	Int	11	Id dari surat
Id_ayat	Int	11	Id dari ayat
Term	Varchar	50	Kata hasil preprocessing ayat
Tf	Int	11	Nilai frekuensi term dalam tiap dokumen
Df	Int	11	Nilai banyaknya terjemahan yang mengandung suatu term
Idf	Double		Hasil inverse document frekuensi tiap term
W	Double		Bobot dari tiap term
W2	Double		Nilai kuadrat bobot tiap term

e.Tabel query

Tabel tabel_query digunakan untuk penyimpanan sementara query inputan user yang telah dilakukan preprocessing dan bobot dari tiap term dalam query untuk nantinya digunakan dalam proses perhitungan cosine similiarity dengan term terjemahan. Adapun struktur tabel_query adalah sebagai berikut

Tabel 3.10 Struktur tabel tabel_query

Field	Type	Size	Keterangan
Id_query	Int	11	Incremental
Kata_query	Varchar	50	Term query
Frekuensi	Int	11	Jumlah tiap term query
Bobot	Double		Bobot dari tiap term query
Bobot2	Double		Nilai kuadrat bobot tiap term query

f. Tabel perhitungan cosine

Tabel tabel_hitung_cosine berfungsi dalamperhitungan simmiliarity tiap ayat terhadap query,untuk mempermudah dalam perhitungan cosine similiarity dan perankingan dalam mendapatkan hasil dokumen yang memliki nilai similiarity tertingggi. Tabel ini memiliki struktur tabel seperti berikut :

Tabel 3.11 Struktur tabel tabel_hitung_cosine

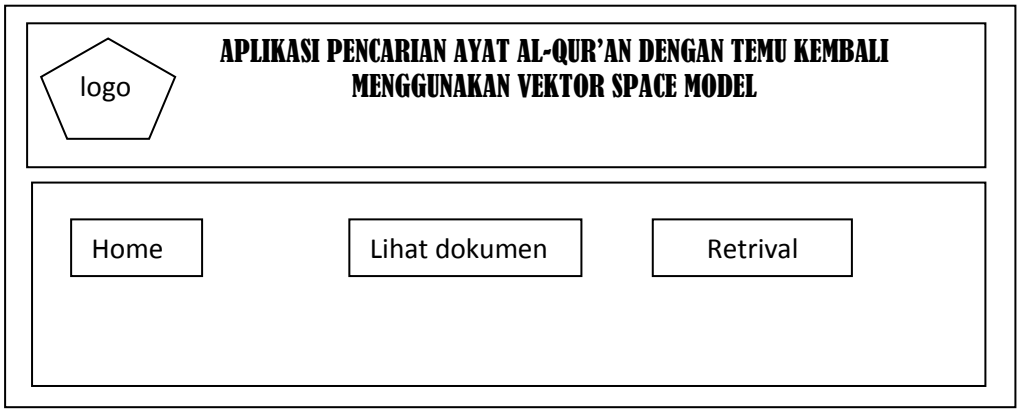
Field	Type	Size	Keterangan
Id_hitung	Int	11	Incremental
suratID	Int	11	Id dari surat
ayatID	Int	11	Id dari ayat
Sqrt_w2	Double		Panjang vector document
Sum_kd	Double		Nilai jumlah dari perkalian bobot term query dengan term documents. $(d_i \cdot q)$
Cosine	Double		Nilai cosine

3.2.6. Perancangan Interface

Aplikasi ini menggunakan bahasa pemrograman PHP menggunakan editplus3 sebagai edtor PHP-nya. Pada sistem pencarian ini terdapat beberapa halaman, sebagaimana berikut ini:

3.2.6.1.Antarmuka Halaman awal (home)

Halaman awal ini merupakan tampilan awal ketika user mulai menjalankan aplikasi pencarian. Terdapat informasi-informasi tentang aplikasi. Rancangan interface halaman ini adalah seperti ditunjukkan pada **gambar 3.17**



Gambar 3.17 Antarmuka halaman awal system

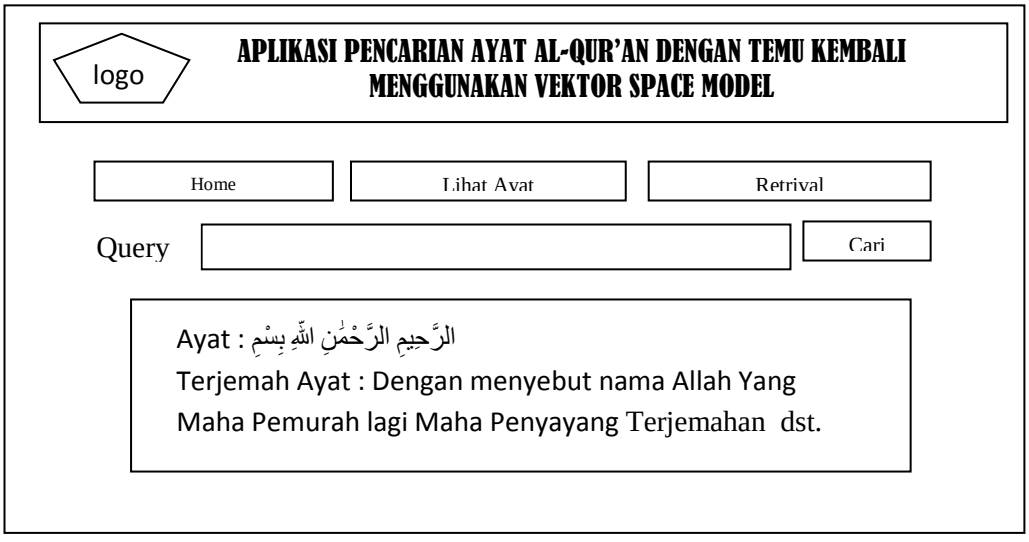
3.2.6.2.Antarmuka Pencarian

Halaman ini merupakan tampilan pencarian bagi user untuk mencari ayat Al-qur'an yang diinginkan dengan menginputkan query. Perancangan tampilan interface halaman ini bisa dilihat pada **gambar 3.18** dibawah ini

Gambar 3.18 Antarmuka halaman Pencarian

3.2.6.3.Halaman Lihat Dokumen

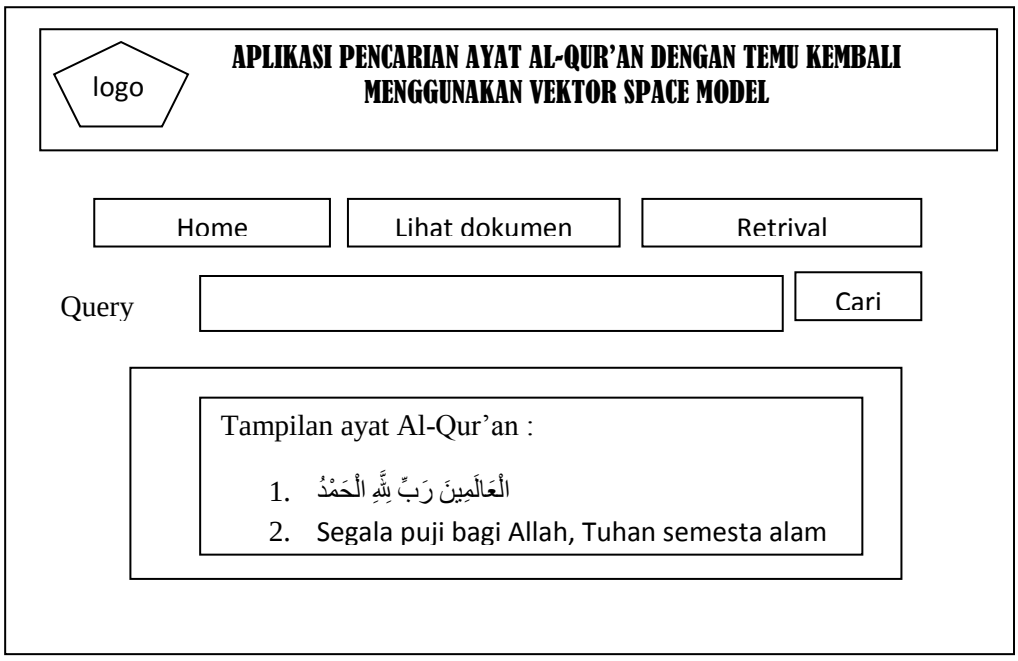
Halaman ini merupakan tampilan bagi user yang menampilkan seluruh ayat al-qur'an danterjemahan yang ada dalam al-qur'an . Perancangan tampilan interface halaman seperti dibawah ini.



Gambar 3.19 Antarmuka halaman lihat dokumen

3.2.6.4. Halaman Hasil Pencarian

Halaman hasil menampilkan hasil dari proses sistem aplikasi pencarian ayat al-qur'an, setelah diinputkan query oleh user. Berupa ayat yang paling relevan dengan query dari user dan beberapa list ayat dan terjemahan al-qur'an yang mirip lainnya. Tampilan dari halaman hasil ini seperti gambar dibawah ini.



Gambar 3.20 Antarmuka halaman hasil pencarian

3.2.7 Perancangan uji coba

Skenario sistem pencarian ayat al-qur'an yang dibangun ini difokuskan kepada hasil terjemahan ayat al-qur'an yang ditemukan. Pengujian dilakukan secara uji empiris. Dengan melakukan langah-langkah berikut:

1. Input query berupa tema yang diinginkan dan melihat hasil pencarian
2. Mencatat hasil ayat-ayat yang dimuncukan sistem pencarian
3. Melakukan analisis terhadap hasil uji coba.

Analisis dilakukan dengan menghitung nilai *Recall* adalah rasio antara dokumen relevan yang berhasil ditemukembalikan (*diretrieved*) dari seluruh dokumen relevan yang ada di dalam sistem, sedangkan *precision* adalah rasio dokumen relevan yang berhasil ditemukembalikan dari seluruh dokumen yang berhasil ditemu-kembalikan. (Grossman, 2002), recall dan accuracy menggunakan

persamaan rumus (2.6, 2.7 dan 2.8) dari hasil pencarian. Hal ini dilakukan secara berulang. Untuk menilai perbandingan keakurasian system ini menggunakan ‘Tafsir Rahmat penulis H.Omer bakry Percetakan Mutiara” yang sudah dijelaskan ayat-ayat yang relevan sesuai dengan tema. Dari hasil uji coba analisis dapat mendapatkan hasil pencarian yang dianggap relevan, dengan nilai precission, recall dan akurasi yang tinggi.