PENCARIAN AYAT – AYAT AL-QUR'AN BERDASARKAN KONTEN MENGGUNAKAN *TEXT MINING* BERBASIS APLIKASI DESKTOP

Aditya Herdianto
Jurusan Teknik Informatika, Dosen Pembimbing
Politeknik Elektronika Negeri Surabaya
Institut Teknologi Sepuluh Nopember Surabaya
Kampus PENS-ITS Keputih Sukolilo Surabaya 60111
Telp (+62)31-5947280, 5946114, Fax. (+62)31-5946114
Email: joedith_archilles@yahoo.com

Abstrak

Dalam kehidupan manusia, banyak permasalahan tentang agama yang sulit untuk terpecahkan. Karena bentuk dari Al-Qur'an adalah konvensional, sehingga sulit untuk dipelajari. Ayat - ayat yang ada pada Al-Qur'an sifatnya terpencar sehingga dibutuhkan waktu lama untuk mencari sebuah permasalahan. Proses pencarian ayat - ayat Al-Qur'an dengan cara konvensional ataupun dengan Al-Qur'an digital yang selama ini ada di internet tidak cukup membantu, jika hasil yang kita inginkan adalah ayat - ayat tertentu yang sesuai dengan masalah yang kita hadapi. Dengan begitu dibutuhkan sebuah sistem untuk mengenali, mencari dan mengelompokkan masalah yang dibutuhkan oleh *user*. Sehingga sistem tersebut dapat menampilkan ayat - ayat Al-Qur'an sebagai referensi dan solusi. Dengan masalah diatas, maka dapat digunakan sebuah proses pengenalan teks yang disebut *Text Mining*. Dengan proses tersebut maka masalah yang dibutuhkan *user* dilakukan dengan beberapa metode yaitu, *parsing*, *steamming*, dan *morphing*. Sehingga dapat mengenali masalah yang dibutuhkan oleh *user* dan ayat - ayat yang berhubungan dengan masalah *user* dapat langsung ditampilkan.

1. Latar Belakang

Permasalahan agama merupakan permasalahan yang sangat komplek. Untuk menyelesaikan permasalahan tersebut diperlukan pemahaman dalam mengkaji Al-Qur'an. Bila dilakukan tanpa pemahaman Al-Qur'an, maka akan menimbulkan perpecahan. Dalam Al-Qur'an setiap permasalahan tidak mengacu pada satu ayat ataupun satu surat. Sedangkan jumlah ayat dan surat di dalam Al-Qur'an, sangatlah banyak. Sehingga bila dilakukan pencarian secara manual terlalu banyak memakan waktu, dibuatlah sebuah sistem yang dapat melakukan pencarian lebih cepat dan ayat - ayat yang dibutuhkan dapat langsung ditemukan.

Latar belakang pengambilan judul proyek akhir ini adalah sulitnya mengimplementasikan Al-Qur'an pada kehidupan sehari – hari terutama dalam pemecahan masalah oleh beberapa kalangan masyarakat, dikarenakan bentuk Al –Qur'an yang konvensional sulit untuk dipelajari. Ayat – ayat dalam Al – Qur'an disusun tidak berdasarkan sebuah permasalahan, sehingga sifatnya yang berpencar, sehingga memakan waktu lama untuk pencarian ayat – ayat yang dibutuhkan.

Proses pencarian ayat - ayat Al-Qur'an dengan cara konvensional ataupun dengan Al-Qur'an digital yang selama ini ada di internet (misalnya: www.al-qurandigital.com) tidak cukup membantu, jika hasil yang kita inginkan adalah ayat - ayat tertentu yang sesuai dengan masalah yang kita hadapi. Dengan begitu dibutuhkan sebuah sistem untuk mengenali, mencari dan mengelompokkan masalah yang diinputkan oleh *user*. Sehingga sistem tersebut dapat menampilkan ayat - ayat Al-Qur'an sebagai referensi dan solusi.

M Syaiful Rizal dengan Tugas Akhir Pencarian ayat – ayat Al – Qur'an tentang Permasalahan Akidah berdasarkan content menggunakan Text Mining [9] telah berhasil membuat sistem untuk pencarian ayat ayat Al-Qur'an, hanya saja proses penyelesaian masalah berdasarkan akidah. Jadi ayat dan surat yang ada pada sistem tidak menyeluruh di dalam Al-Qur'an. Ini terjadi karena pembatasan masalah yang ada pada sistem hanya menyangkut masalah akidah. Pada Tugas Akhir ini penulis menyempurnakan sistem yang telah ada dan membuat sebuah sistem yang lebih sempurna karena mencakup semua cakupan ayat dan surat di Al-Qur'an. Selain itu, pada sistem sebelumnya hanya bisa digunakan pada *web service*. Sedangkan sistem ini menggunakan aplikasi

desktop. Karena lebih efisien dan dapat digunakan tanpa koneksi internet.

Dengan masalah diatas, maka dapat digunakan sebuah proses pengenalan teks yang di sebut *Text Mining*. Dengan proses tersebut maka masalah yang di inputkan *user* dilakukan dengan beberapa *method* seperti, *parsing*, *steamming*, dan *morphing* sehingga dapat mengenali masalah yang ada.

Dengan proyek akhir ini diharapkan dapat membantu mencarikan referensi berupa ayat dan terjemahan Al – Qur'an terhadap suatu permasalahan. Proyek Akhir ini dibuat berbasis desktop sehingga lebih mudah diakses. Karena tidak perlu koneksi terlebih dahulu ke internet.

2. Dasar Teori

Teori-teori yang digunakan dalam penyelesaian proyek akhir akan dibahas dalam bab ini sesuai kaitannya dengan *text mining*. Serta membahas software - software yang digunakan dalam pembuatan proyek akhir ini.

2.1 Text Mining

Text mining adalah proses menambang data berupa teks dengan sumber data biasanya dari dokumen dan tujuannya adalah mencari kata - kata yang mewakili dalam dokumen sehingga dapat dilakukan analisa keterhubungan dalam dokumen. Data teks akan diproses menjadi data numerik agar dapat dilakukan proses lebih lanjut. Sehingga dalam *text mining* ada istilah *preprocessing data*, yaitu proses pendahulu yang diterapkan terhadap data teks yang bertujuan untuk menghasilkan data numerik.

Pada proses *preprosesing* ada beberapa tahapan yang dilakukan, antara lain:

- Penghapusan *Format* dan *Mark-Up*

Jika dokumen yang digunakan bukan berupa teks murni maka tahap ini dilakukan. Karena dokumen teks yang biasanya kita lihat berupa format non-teks seperti *html, pdf* atau dalam bentuk *word*. Format ini mengharuskan sebuah teks dilengkapi unsur - unsur tambahan untuk dapat menghasilkan tampilan yang *friendly user*. Informasi - informasi itu dihilangkan karena dianggap tidak perlu dan tidak mencerminkan isi sebuah dokumen teks.

- Penghapusan Tanda Baca dan Angka

Tanda baca juga dianggap tidak penting, karena kebetulan dalam penelitian yang dilakukan pada proyek akhir ini tidak memperhatikan keterkaitan kata, kalimat, ataupun sejenisnya jadi kata dianggap berdiri sendiri.

- Pengubahan dari Huruf Besar ke Huruf Kecil semua

Dimaksudkan agar mudah dalam proses pencarian dalam sistem yang akan dibuat.

- *Parsing* dan *Stemming*

Penguraian kata ke dalam bentuk tunggal dan pembentukan kata kedalam bentuk dasarnya, sehingga kata - kata yang mempunyai bentuk kata dasar yang sama akan dikelompokkan.

- Pembobotan

Dimulai dengan perhitungan jumlah kata dalam setiap dokumen, yang kemudian akan dihitung menggunakan skema pembobotan yang dikehendaki.

2.2 Metode Korelasi

Untuk mencari keterkaitan antara masalah dengan ayat Al – Qur'an maka digunakan metode korelasi. Metode ini menggunakan penghitungan matematis yang menghasilkan sebuah bilangan yang menunjukkan keterkaitan antara masalah dan ayat ayat Al-Qur'an.
Rumus korelasi adalah:

$$Correl(X,Y) = \frac{\sum (x-\bar{x})(y-\bar{y})}{\sqrt{\sum (x-\bar{x})^2 \sum (y-\bar{y})^2}}$$

Dimana x dan y adalan *mean sample* rata - rata (array1) dan nilai rata - rata (array2).

Mengembalikan koefisien korelasi dari array1 dan array2 rentang sel. Gunakan koefisien korelasi untuk menentukan hubungan antara dua properti. Sebagai contoh, Anda dapat memeriksa hubungan antara temperatur rata-rata lokasi dan penggunaan AC.

Jika sebuah array atau argumen referensi berisi teks, nilai-nilai logis, atau sel-sel kosong, nilai-nilai tersebut diabaikan, namun sel-sel dengan nilai nol akan disertakan mengembalikan CORREL # N/A.

2.3 Bahasa Pemrograman JAVA

Bahasa pemrograman Java dapat dibedakan menjadi dua jenis, yaitu applet dan aplikasi.

Keterangan tentang dua jenis dari bahasa tersebut adalah sebagai berikut:

- Applet

Adalah program yang dibuat dengan Java, dapat diletakkan pada *web server* dan diakses melalui *web browser*. Dalam hal ini *browser* yang digunakan adalah yang memiliki kemampuan Java (misalnya: *Netscape Navigator*, *Internet Explorer*, dan *HotJava*).

- Aplikasi

Adalah program yang dibuat dengan Java dan bersifat perangkat umum. Aplikasi adalah program yang dibuat dengan Java yang bersifat umum. Aplikasi dapat dijalankan secara langsung, tidak perlu perangkat lunak *browser* untuk menjalankannya. Aplikasi dapat dieksekusi secara langsung setelah dilakukan kompilasi terlebih dahulu.

Class - *class* yang menyimpan fungsi proses String dan Karakter pada bahasa pemrograman yang digunakan adalah sebagai berikut:

- *Class* java.lang.String
- *Class* java.lang.StringBuffer
- *Class* java.lang.Character
- *Class* java.util.StringTokenizer

Tabel 2.1 *Tabel konstruktor class string*

Konstruktor	***Keterangan***
String()	*Menciptakan objek String() yang berisi string kosong (jumlah karakter sama dengan).*
String(byte[] v, int ofset, int jumlah)	*Menciptakan objek String yang berasal dari array yang dirujuk oleh "v". Offset menentukan posisi dalam array yang akan diberikan ke objek string. Int jumlah menentukan jumlah karakter yang ingin di ambil dari array dari posisi offset.*
String (char v [])	*Menciptakan objek string yang diiambil dari array "v".*
String(char[] v , int ofset, int jumlah)	*Sama dengan String(byte[] v, int offset, int jumlah).*
String (String v)	*Menciptakan objek string dari objek string "v".*

Tabel 2.2 *Tabel metode class string*

Metode	**Keterangan**
Copy Value Of (char data[])	Menghasilkan kelas berobjek String yang berisi data yang sama dengan array data
Copy Value Of (char data[],int offset,int jum)	Serupa dengan Copy Value Of (char data[]) namun dimulai dari posisi *offset* dan jumlaj karakter *jum*
ValueOf(boolean b)	Menghasilkan objek string yang berisi true kalau argument berupa true begitu juga sebaliknya
ValueOf(char c)	Menghasilkan objek string yang berisi sebuah karakter yang sesuai dengan c
ValueOf(double d)	Menghasilkan objek string yang berisi sebuah bilangan yang sesuai dengan d
ValueOf(float f)	Menghasilkan objek string yang berisi sebuah bilangan yang sesuai dengan f
ValueOf(int i)	Menghasilkan objek string yang berisi sebuah bilangan yang sesuai dengan i
ValueOf(Object obj)	Menghasilkan objek string yang berisi sebuah objek yang sesuai dengan obj, biasanya menggunakan to_string()
CharAt(int indeks)	Menghasilkan karakter dalam posisi index
Compare to(String s)	Menghasilkan nilai bertipe int yang berupa : • Bilangan positif kalau string ini lebih besar daripada string • Nol jika string ini sama dengan string s • Bilangan negatif kalau string ini lebih kecil daripada string s *Catatan* : Pengertian lebih besar atau lebih kecil dari perbandingan karakter.
Concat (String s)	Menghasilkan objek string yang meru[pakan gabungan dari string ini dengan string s
EndWith(String s)	Menghaslkan nilai true kalau string ini berakhiran dengan string s
Equals (Object obj)	Menghasilkan nilai true jika string sama dengan

	object obj
equaslIgnoreCase(String s)	Menghasilkan nilai true jika string berakhiran dengan string s tanpa memperhatikan hurug besar atau kecil
Get bytes()	Menghasilkan array bertipe byte yang nilai dari setiap karakter dalam string ini.
Indexof(int ch)	Menghasilkan nilai bertipe int yang menyatakan posisi pertama untuk karakter yang nilainya ch
Indexof(String s)	Menghasilkan nilai bertipe int yang menyatakan posisi pertama untuk substring yang nilainya s
Indexof(String s, int index)	Menghasilkan nilai int dari posisi pertama substring dari string s dimulai dari posisi index
LastIndex(int ch)	Menghasilkan nilai bertipe int yang menyatakan posisi terakhir untuk karakter yang nilainya ch
LastIndex(String s)	Menghasilkan nilai bertipe int yang menyatakan posisi terakhir untuk substring yang nilainya s
LastIndex(String s, int index)	Menghasilkan nilai int dari posisi pertama substring dari string s dimulai dari posisi index. Jika tidak ditemukan nilainya -1
Length ()	Menghasilkan int panjang string
Replace(char lama,char baru)	Menghasilkan string dengan nilai dimana karakter *lama* diganti dengan karakter *baru*
startWith(String s)	Menghasilkan nilai true kalau string ini berawalan dengan string s
startWith(String s,int subset)	Menghasilkan nilai true kalau string ini berawalan dengan string s dimulai dari posisi offset
Substring (int index)	Menghasilkan objek string yang berisi substring dimulai dari karakter pada posisi index hingga karakter terakhir.
Substring (int indexawal,indexakhir)	Menghasilkan objek string yang berisi substring dimulai dari karakter pada posisi *indexawal* hingga karakter index *indexakhir*
toCharArray()	Menghasilkan array bertipe karakter yang berisi karakter dari objek string
To_string()	Menghasilkan objek string ini
Trim()	Menghasilkan objek string yang menghilngkan seluruh spasi ataupun karakter control yang terletak di awal dan di akhir ini.
countTokens()	Memberikan nilai balik berupa int yang menyatakan jumlah token yang belum diproses
hasMoreElements()	Memberikan nilai balik bertipe booelean yang menyatakan belum ada token yang di proses
nextElement()	Memberikan nilai balik bertipe object yang menyatakan token berikutnya
nextToken()	Memberikan nilai balik berupa String yang menyatakan token berikutnya
nextToken(String delim)	Memberikan neilai balik bertipe String yang menyatakan token berikutnya dengan pemisah token berupa *delim*

2.4 *Stemming* Bahasa Indonesia

Stemming adalah proses untuk menggabungkan atau memecahkan setiap varian - varian suatu kata menjadi kata dasar. *Stem* (akar kata) adalah bagian dari kata yang tersisa setelah dihilangkan imbuhannya (awalan dan akhiran). Metode *stemming* memerlukan *input* berupa *term* yang terdapat dalam dokumen. Sedangkan *output*nya berupa *stem*. Algoritma ini diawali dengan pembacaan tiap kata

dari sampel. Sehingga *input* dari algoritma ini adalah sebuah kata yang kemudian dilakukan :

- Pemeriksaan kemungkinan bentuk kata. Setiap kata diasumsikan memiliki 2 awalan (*prefiks*) dan 2 akhiran (*sufiks*). Sehingga bentuknya menjadi:

Prefiks 1 + Prefiks 2 + (morphing) Kata dasar + Sufiks 2 + Sufiks 1

Seandainya kata tersebut tidak memiliki imbuhan sebanyak imbuhan di atas, maka imbuhan yang kosong diberi tanda x untuk *prefiks* dan diberi tanda xx untuk *sufiks*.

- Pemotongan dilakukan secara berurutan sebagai berikut:

AW : A W AK : A K KD : K D

Keterangan langkah pemotongan adalah sebagai berikut:

a. AW I, hasilnya disimpan pada p1
b. AW II, hasilnya disimpan pada p2
c. AK I, hasilnya disimpan pada s1
d. AK II, hasilnya disimpan pada s2

Pada setiap tahap pemotongan di atas diikuti dengan pemeriksaan di *database* (kumpulan masalah) apakah hasil pemotongan itu sudah berada dalam bentuk dasar. Kalau pemeriksaan ini berhasil maka proses dinyatakan selesai dan tidak perlu melanjutkan proses pemotongan imbuhan lainnya.

2.5 Aturan Imbuhan

2.5.1 Imbuhan Awalan

Yang dimaksud dengan imbuhan awalan adalah imbuhan yang diletakkan di depan kata dasar. Jenis – jenis yang termasuk imbuhan awalan adalah imbuhan ‘me-‘, ‘pe-‘, ‘per-‘‘di-‘, ‘ke-‘, ‘se-‘, ‘ke-‘, ‘ber-‘ dan ‘ter-‘.

2.5.2 Pemotongan Akhiran Imbuhan

Yang dimaksud dengan akhiran imbuhan adalah imbuhan yang diletakkan di belakang kata dasar. Jenis - jenis yang termasuk dalam akhiran imbuhan :

• Akhiran imbuhan ‘-kan’

• Akhiran imbuhan ‘-an’

• Akhiran imbuhan ‘-i’

• Akhiran imbuhan ‘-lah’

• Akhiran imbuhan ‘-nya’

• Akhiran imbuhan ‘-kah’

2.5.3 Pemotongan Imbuhan Gabungan

Yang dimaksud dengan imbuhan gabungan adalah imbuhan yang diletakkan didepan dan dibelakang kata dasar. Jenis-jenis yang termasuk dalam imbuhan gabungan:

• Imbuhan gabungan ‘ber-an’ dan ‘ber-kan’

• Imbuhan gabungan di-kan’ dan ‘di-i’

• Imbuhan gabungan ‘diper-kan’ dan ‘diper-i’

• Imbuhan gabungan ‘ke-an’

• Imbuhan gabungan ‘me-kan’ dan ‘me-i’

• Imbuhan gabungan ‘memper-kan’ dan ‘memper-i’

3. Perencanaan dan Pembuatan

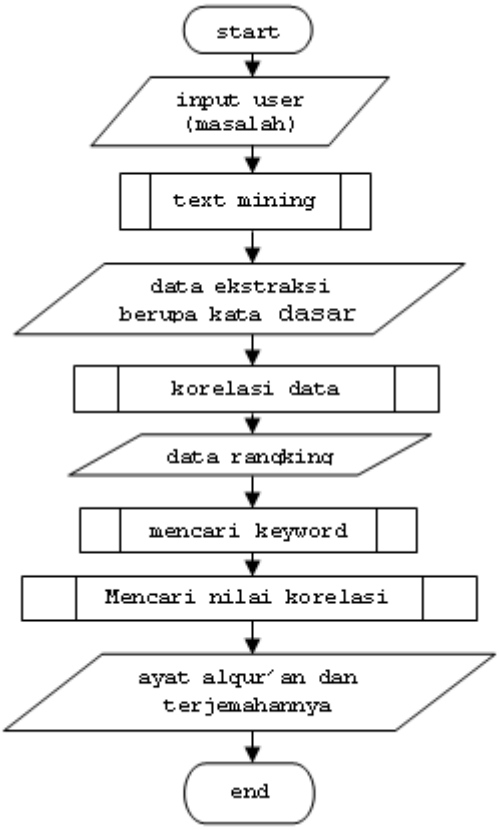
Sebelum pembuatan akan dilakukan perencanaan untuk alur pada sistem yang terdiri dari Deskripsi Umum, Rancangan Umum, Perancangan Database, Perancangan Proses, Perancangan Interface dan Implementasi

3.1 Dekripsi Umum Perangkat Lunak

Proyek akhir ini adalah sebuah aplikasi yang dibuat untuk memberikan alternatif pencarian ayat - ayat Al-Qur’an berdasarkan masalah yang di inputkan user dan merupakan masalah umum yang dihadapi manusia. Pada proyek akhir ini digunakan sebuah metode yaitu, *text mining*. Yang berfungsi untuk mengenali masalah yang di inputkan oleh user. Mulai dari *parsing*, *stemming*, pembobotan, dan mencari *keyword*. Fasilitas utama dalam proyek akhir ini adalah menampilkan ayat Al-Qur’an beserta artinya berdasar dari permasalahan yang dinputkan oleh user.

3.2 Perancangan Aplikasi Secara Umum

Berikut adalah flowchart umum pada proyek ini :

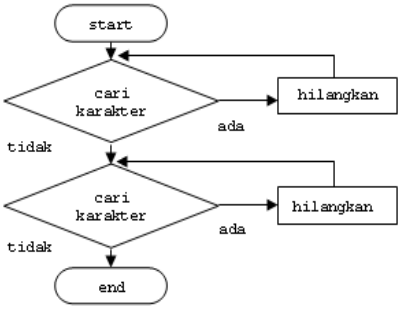


Gambar 3.1 *Flowchart Umum*

Proses pada proyek akhir ini adalah :

1) Proses preprocessing → terdapat metode untuk menghilangkan tanda baca, angka, simbol, mark up, dan membuat semua kata seragam yaitu dengan menjadikannya huruf kecil semua.
2) Stemming kalimat → yaitu metode untuk memecah kalimat menjadi kata – kata yang berdiri sendiri. Karena dalam text mining kata dalam kalimat tidak memiliki keterkaitan.
3) Stemming kata → proses ini adalah memotong imbuhan dan awalan sehingga kata – kata hanya dalam bentuk kata dasar. Sehingga pembobotannya valid.
4) Mencari nilai korelasi → mencari nilai keterkaitan antara masalah dengan ayat Al – Qur'an.
5) Pembobotan → merangking atau mengelompokkan kata dasar.
6) Mencari keyword → keyword adalah kata – kata yang jumlahnya lebih dari 2 dan merupakan kata kunci.
7) Menampilkan hasil → menampilkan ayat – ayat Al – Qur'an dan arti berdasar keyword, dengan di dikelompokkan pertampilan *(paging)* sebanyak 5 ayat perhalaman.

Selain proses di atas ada proses lain yaitu, proses untuk menghilangkan karakter tanda baca. Flowchart proses tersebut adalah:



Gambar 3.2 *Flowchart menghilangkan tanda baca*

Dari flowchart di atas dapat kita lihat bahwa pada awalnya sistem mencari tanda baca dan angka, kemudian menghilangkannya. Prosesnya akan berjalan rekursif hingga tak ada tanda baca dan angka lagi yang ada pada masalah.

3.3 Perancangan Database

Karena menggunakan konsep *Text Mining* secara murni maka database hanyalah berisi tentang tabel master Al-Qur'an dan daftar kata *stop list* dan tidak ada relasi tabel. Berbeda dengna Tugas Akhir sebelumnya yang menggunkan relasi database untuk identifikasi masalah.

Tabel 3.1 *Tabel entitas pada tabel ayat Al-Qur'an*

Id_Filed	Deskripsi	Type
Id_ayat	Primary Key untuk ayat	Varchar(10)
Surat	Surat al-Qur'an	Varchar(10)
Ayat	Ayat al-Qur'an	Varchar(10)
Arab	Tampilan ayat Al – Qur'an dalam arab	Varchar(10)
Terjemahan	Tampilan ayat Al – Qur'an dalam indonesia	Varchar(10)

Tabel 3.2. *Tabel entitas pada tabel stop list*

Id_Filed	Deskripsi	Type
Id_stop	Primary Key untuk stop list	Varchar(10)

Stop_list	Kata yang akan di stop list	Varchar(10)

3.4 Perancangan Proses

3.4.1 Preprocessing

Dalam Preprocessing terdapat beberapa langkah yang dilakukan yaitu:

- Menerima Input Masalah Dan Menyimpannya Di Database

  User bisa menginputkan masalah dengan 3 cara, yaitu:
  1) mengetikkan pada *TextArea* pada form isian.
  2) Mengetikkan masalah pada file ber-extension *.txt
  3) Kombinasi dari cara pertama dan kedua.

  Oleh karena itu, harus ada sebuah metode yang dapat melayani ketiga cara diatas. Yaitu menangkap parameter text, operasi file dan mengabungkan keduanya. Setelah melakukan proses pengenalan input, selanjutnya input tersebut akan di simpan dalam database.

  Form input masalah adalah sebagai berikut:

Gambar 3.3 *Form Input Masalah*

- Menghilangkan Tanda Baca, Angka, Simbol, Mark Up

  Dalam Text Mining tanda baca, angka, dan mark up tidak dibutuhkan. Karena kata- kata dalam metode text mining dianggap berdiri sendiri. Sehingga hal – hal diatas dihapuskan untuk effiesiensi proses. Proses yang ada adalah sebagai berikut :

1) Mengecek kalimat masalah perkarakter.
2) Jika karakter tersebut adalah tanda baca, angka, simbol, maka karakter tersebut di diganti dengan karakter kosong ( ).

- Membuat Semua Kata Huruf Kecil Semua

  Penulisan huruf besar dan kecil tidak diperhatikan dalam *Text Mining (Non CaseSensitive).* Dengan begitu kita harus menyeragamkan huruf pada kalimat. Hal itu berfungsi pada pembobotan dan pengecekan *keyword* pada database, yang notabene memperhatikan huruf besar dan kecil (*case sensitive*).

3.4.2 *Stemming* Kalimat

User memasukkan masalah berupa kalimat, padahal *keyword* yang dicari berupa kata. Hal ini menyebabkan kalimat masukan dari user harus menjalani proses *stemming* menjadi ”kata”. Dalam proses ini digunakan spasi sebagai acuan pemotongan. Seperti yang kita tahu bahwa dalam kalimat kata yang satu dengan yang dipisahkan oleh spasi.

Proses dalam stemming kalimat adalah sebagai berikut :

1) Mencari posisi spasi
2) Memotong kalimat menjadi 2 bagian dengan spasi yang ditemukan sebagai acuan posisi pemotongan.
3) Bagian pertama dari proses pemotongan menjadi kata baru, dan sisanya menjadi kalimat baru yang akan dipotong selanjutnya.
4) Kembali ke proses 1 sampai tidak lagi ditemukan spasi.

3.4.3 Memecah Akhiran -*Ullah*

Dalam Al-Qur’an terkadang kita menemukan kta dengan akhiran –*Ullah*. Misal : Rasulullah,kitabullah,waliyullah, dll. Padahal akhiran –*Ullah* ini tidak terdapat dalam kaidah bahasa Indonesia. Kata ini merupakan serapan dari bahasa Arab yang berarti ”Milik Allah”. Rasulullah artinya ”Rasulnya Allah”. Bila diperhatikan dengan seksama semua yang ada di dunia ini pasti milik Allah sehingga tanpa penegasanpun kita sudah yakin bahwa objek yang dimaksud merupakan ciptaan Allah.

Oleh karena itu penggunaan akhiran –*Ullah* tidak memiliki arti sendiri jika digabung dengan kata yang lain. Sehingga kata –*Ullah* bisa dihilangkan, tanpa merubah arti kata semula. Pemotongan kata *Ullah* ini tidak disertai *morphing*, yaitu proses perubahan bentuk karena imbuhan dan akhiran.

Contoh :

Rasulullah = rasul + Ullah

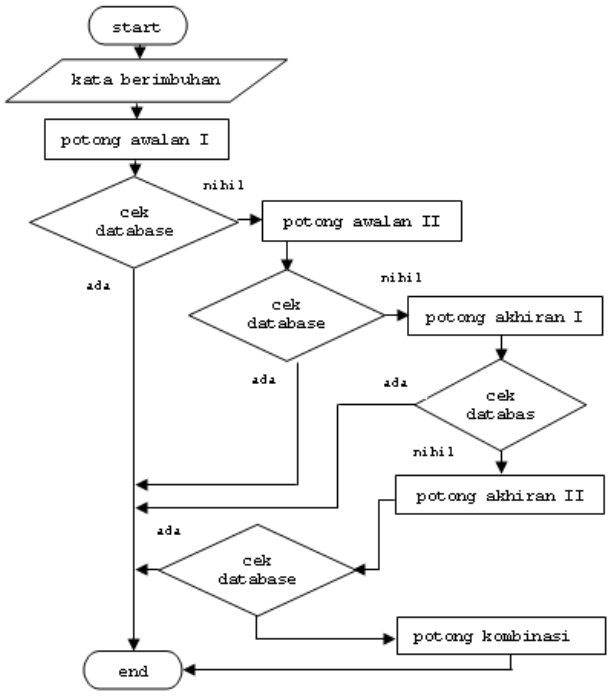
Kitabullah = kitab + Ullah

3.4.4 *Stemming* Kata

Stemming kata adalah proses mengubah sebuah kata berimbuhan menjadi kata dasar. Hal ini berguna untuk mencari kata terbanyak yang di ungkapkan oleh user, yang nantinya digunakan menjadi sebuah keyword. Ada beberapa proses dalam *Stemming* ini yaitu :

1. Mengecek AwalanI (aw1)
2. Morphing I (digunakan untuk Awalan 1)
3. Mengecek Awalan II
4. Mengecek akhiran I
5. Mengecek akhiran II
6. Mengecek kombinasi (gabungan antara awalan dan akhiran)

Flowchart untuk pemotongan kata adalah sebagai berikut:



Gambar 3.4 *Flowchart Stemming*

3.4.5 Pembobotan

Setelah semua kata diasumsikan sudah dalam bentuk kata dasar, maka selanjutnya adalah proses pembobotan. Proses ini adalah mengelompokkan kata yang sama dan mencari jumlahnya. Hal ini berfungsi untuk mengetahui kata - kata apa saja yang paling banyak muncul, yang nantinya bisa masuk seleksi untuk menjadi *keyword*. Berikut adalah proses dalam pembobotan :

1) Melakukan pengecekan apakah sebuah kata sama dengan kata yang lain.
2) Jika sama maka, menambahkan parameter jumlah kata pertama, sedangkan jumlah kata kedua di *ground.*
3) Meranking kata berdasar jumlah.

3.4.6 Korelasi

Untuk mencari keterkaitan antara masalah dengan ayat Al-Qur'an maka digunakan metode korelasi. Metode ini menggunakan penghitungan matematis yang menghasilkan sebuah bilangan yang menunjukkan keterkaitan antara masalah dan ayat ayat Al-Qur'an. Jadi setiap masalah akan dikorelasikan terhadap Al-Qur'an. Proses pada korelasi yaitu:

1. Mengi*nput* suatu masalah.
2. Melakukan proses ekstraksi terhadap masalah.
3. Melakukan filter *stoplist* pada masalah.

4. Melakukan proses korelasi terhadap ayat ayat Al-Qur'an.
5. Mendapatkan nilai korelasi.

3.4.7 Mencari *Keyword*

Setelah kata - kata tersebut dikelompokkan dan dirangking, proses selanjutnya adalah mencari dan menentukan *keyword* untuk masalah yang di inputkan oleh user. *Keyword* adalah sebuah kata yang muncul 2 kali atau lebih. Namun sebelumnya kata yang tidak perlu disimpan di stop list.

Berikut proses dalam mencari *keyword*:

1) Mengambil kata yang muncul lebih dari minimal 1 kali.
2) Memeriksa kata tersebut dalam tabel stop list, jika ada maka di hilangkan kata tersebut.
3) Jika tidak ada, maka kata tersebut menjadi keyword.

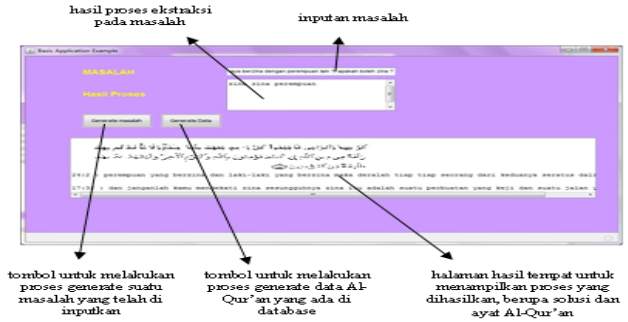
3.4.8 Menampilkan Hasil

Selanjutnya, setelah *keyword* ditemukan maka proses yang harus dilakukan adalah menampilkan hasil berupa ayat Al-Qur'an, terjemahan dalam Bahasa Indonesia dan keterangan. Menampilkan hasil ini ada 2 macam karena *keyword*nya ada 2 jenis. Jenis pertama adalah jika *keyword* berupa kata kunci masalah dalam tabel masalah, atau termasuk rukun iman dalam tabel rukun iman. Berikut adalah proses dalam menampilkan hasil :

a. Mengambil indeks ayat Al-Qur'an yang termasuk dalam membahas masalah berdasar *keyword*.
b. Menampilkan ayat Al-Qur'an dalam arab, terjemahan Bahasa Indonesia, dan keterangan.
c. Melakukan proses *pagin*, yaitu menampilkan hasil 5 ayat per halaman.

3. 5 Perancangan *Interface* (Tatap Muka)

Interface berupa halaman aplikasi *desktop* yang diakses langsung yang terdiri dari:



Gambar 3.5 *Layout aplikasi desktop*

3. 6 Implementasi *Interface* (Tatap Muka)

*Input*an masalah di atas adalah halaman untuk mengisikan masalah. Sisipan adalah masalah yang sudah tersimpan dalam format *.txt. Dengan hal ini *user* bisa mengetikkan nama file langsung yang akan di*upload*.

Halaman hasil adalah ayat Al-Qur'an yang berhubungan dengan masalah yang *input*kan oleh user.

4. Pengujian dan Analisa Sistem

4.1 Pengujian Sistem

Uji coba fungsionalitas dilakukan untuk melihat apakah fungsifungsi dasar aplikasi berjalan sebagai mana mestinya. Hasil uji coba ditunjukkan dengan hasil *screenshoot aplikasi.* Berikut adalah macam uji coba dalam bab ini :

4.1.1 Keyword Berupa Kata Dasar

File masukan misalnya :

Gambar 4.1 *Hasil dari masukan dengan keyword kata dasar*

Dari masukan masalah diatas maka dapat diketahui bahwa keyword masalah adalah kata 'kafir'. Hal ini karena kata kafirada dalam index masalah dalam database dan muncul sebanyak 2 kali. Setelah itu barulah ditampilkan ayat – ayat Al-Qur'an yang berhubungan 'kafir'. Dan dapat dilihat juga bahwa percobaan berhasil.

4.1.2 Keyword dengan Awalan

4.1.2.1 Awalann ”me –”

Gambar 4.2 *Halaman hasil dengan keyword awalan me -*

Dari proses ditas kita lihat bahwa kata melihat menjadi lihat denga rumus : me + lihat. Dan dapat dilihat juga bahwa percobaan berhasil.

4.1.2.2 Awalan ”mem –”

Berikut contoh masukan untuk keyword yang mengandung awalan mem-:

Gambar 4.3 *Halaman hasil dengan keyword awalan mem -*

Dari proses di atas di dapatkan sebuah keyword yaitu ’pilih’. Dikarenakan sistem menangkap kata memilih, yang muncul 2 kali. Kemudian kata tersebut diurai menjadi kata dasarnya seperti rumus berikut:

Memilih = me + pilih (proses morphing dengan menambah huruf p menjadi huruf pertama)

Dari hasil tampilan hasil pada gambar 4.3 dapat disimpulkan bahwa percobaan berhasil.

4.1.2.3 Awalan ”men –”

Berikut contoh masukan untuk keyword yang mengandung awalan men- :

Gambar 4.4 *Halaman hasil dengan keyword awalan men -*

Dari proses diatas dapat diketahui bahwa keyword yang didapat adalah tegur. Salah satu kata yang menentukan adalah menegur yang diuraikan menjadi tegur seperti rumus berikut: *men + (t)+egur (terjadi morphing dengan huruf t menjadi huruf depan)*. Dari tampilan hasil di atas dapat disimpulkan bahwa percobaan berhasil.

4.1.2.4 Awalan ”meng –”

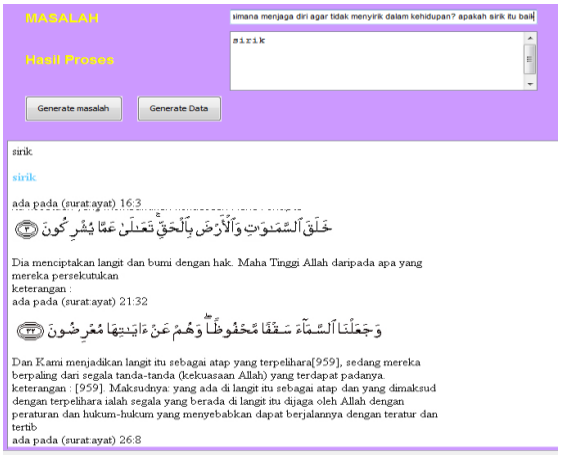
Berikut contoh masukan untuk keyword yang mengandung awalan meng- :

Gambar 4.5 *Halaman hasil dengan keyword awalan meng -*

Dari tampilan hasil di atas dapat disimpulkan bahwa percobaan berhasil.

4.1.2.5 Awalan ”meny –”

Berikut contoh masukan untuk keyword yang mengandung awalan meny-:



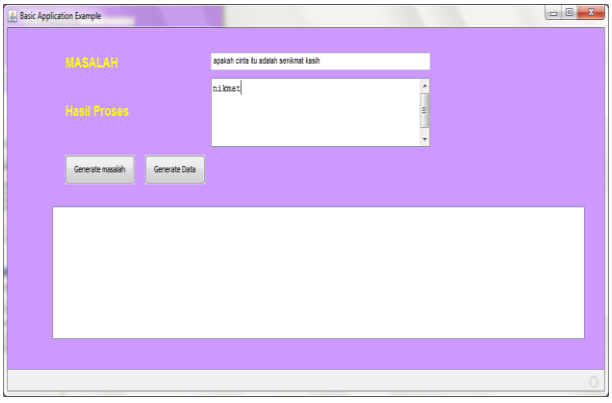
***Gambar 4.6** Halaman hasil dengan keyword awalan meny -*

Dari proses kita bisa tahu bahwa kata menyirik dipotong menjadi sirik dengan rumus : *meny + (s) irik (adanya morphing dengan menambahkan huruf pertama kata dasar)*

Dari tampilan hasil di atas dapat disimpulkan bahwa percobaan berhasil.

4.1.2.6 Awalan "se", "ke", "di"

Sifat dari 3 jenis awalan ini sama persis, yaitu tidak terjadi *Morphing*. Berikut adalah contoh *input* masalah sebuah masalah dimana keywordnya berimbuhan jenis ini:



***Gambar 4.7** Halaman masukan dengan keyword awalan se-, ke-, di-*

Dari kalimat masalah di atas kita, akan mendapati tampilan proses pemotongan sebagai berikut :

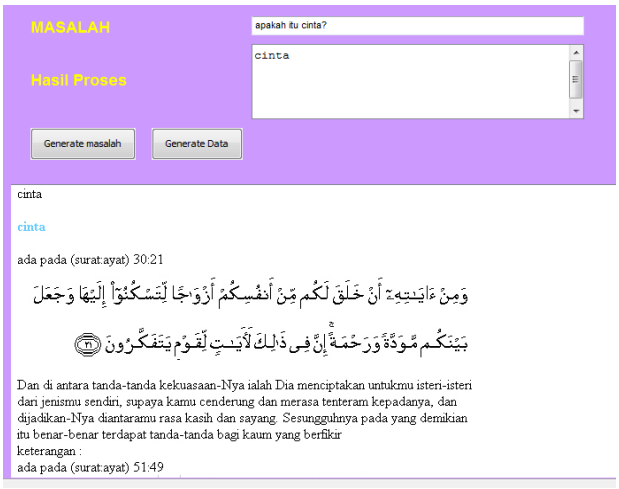
Dari proses diatas dapat dilihat pemotongan pada awalan se,ke,di tidak terjadi morphing.

Senikmat = se + nikmat

Kehati = ke + hati

Dicipta = di + cipta

Dari rumus tersebut didapatkan hasil sebagai berikut:



***Gambar 4.8** Halaman hasil dengan keyword awalan se-, ke-, di-*

Dari tampilan hasil diatas dapat di simpulkan bahwa percobaan berhasil.

4.1.2.7 Contoh untuk kata yang memiliki 2 Awalan

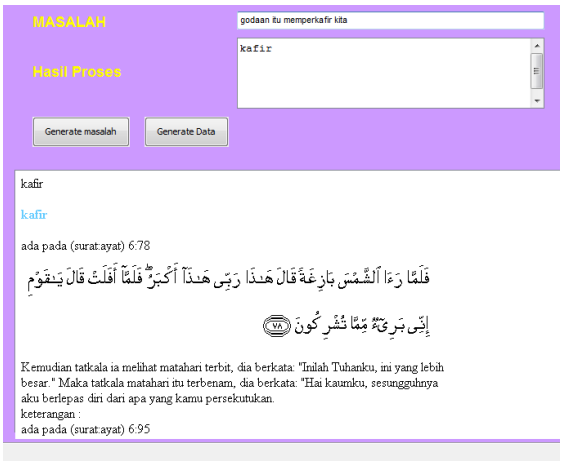
Berikut adalah contoh penggunaan kata yang memiliki 2 awalan misal : diperkafir dan memperkafir yang harusnya menghasilkan kata 'kafir'.

Dari kalimat ada 2 kata berimbuhan dengan kata dasar kafir. Dalam proses ini tidak ada morphing

Diperkafir = di + per + kafir

Memperkafir = mem + per + kafir

Dari rumus tersebut didapatkan hasil sebagai berikut:



Gambar 4.9 *Halaman hasil dengan keyword 2 awalan*

Dari tampilan hasil diatas dapat di simpulkan bahwa percobaan berhasil.

4.1.3 Keyword dengan Akhiran

Sebuah kata dengan akhiran sebenarnya hanya cukup dipotong saja sebanyak karakter akhiran. Jadi program *stemming* pada proyek akhir ini tidak mengecek jenis akhiran, namun jika beberapa karekter pada akhir kata dihapus dan sisanya ternyata cocok pada indeks di database, maka kata tersebut dianggap *keyword*.

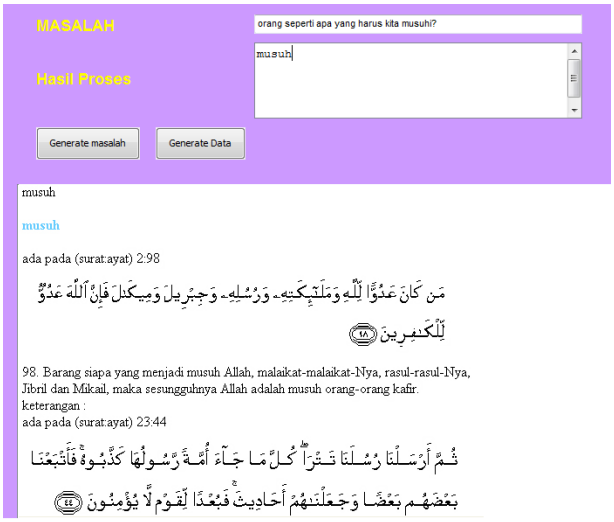
4.1.3.1 Akhiran dengan 1 karakter

Berikut adalah contoh tampilan input untuk kata yang memilki akhiran 1 karakter (misalkan: akhiran - i):

Musuhi = musuh + i

Dari proses diatas dapat diambil sebuah kesimpulan bahwa sebuah kata yang memiliki akhiran i hanya terjadi proses pemotongan tanpa morphing.

Berikut tampilan sistem dengan pemenggalan akhiran:



Gambar 4.10 *Halaman hasil dengan keyword yang mempunyai akhiran 1 karakter*

Dari tampilan hasil di atas dapat disimpulkan bahwa percobaan berhasil.

4.1.3.2 Akhiran dengan 2 karakter

Berikut adalah contoh tampilan input untuk kata yang memilki akhiran 2 karakter (misalkan: akhiran -an atau –mu):

Ciptaan = cipta + an

Nikmatmu = nikmat + mu

Dari proses di atas maka dapat diambil beberapa kesimpulan bahwa proses stemming pada akhiran –an dan –mu tidak terjadi morphing.

Gambar 4.11 *Halaman hasil dengan keyword yang mempunyai akhiran 2 karakter*

4.1.3.3 Akhiran dengan 3 karakter

Pada sebuah kata yang memiliki awalan 3 karakter juga memiliki sifat seperti akhiran dengan 1 dan 2 karakter. Tidak ada proses morphing pada proses *stemming*. Berikut adalah contoh masukan pada sebuah masalah yang *keyword*nya memiliki akhiran dengan 3 karakter.

AmpunkanNya = ampun + kan + Nya

Hasilnya adalah:

Gambar 4.12 *Halaman hasil dengan keyword yang mempunyai akhiran 3 karakter*

4.1.4 Kata yang memiliki awalan dan akhiran

Yang dimaksud dengan kondisi ini adalah jika sebuah kata memiliki awalan dan akhiran sekaligus, baik itu 1 atau 2 awalan dan 1 atau 2 akhiran. Proses dalam gabungan ini secara umum tidak terjadi morphing, jadi tinggal mengecek awalan dan akhirannya, yang kemudian dipotong. Satu – satunya kondisi yang terjadi *morphing* adalah kondisi pada sebuah imbuhan pen – an. Misalnya adalah :

Penataan = pen + (t)ata+an

Terjadi morphing dengan menambahkan karakter *"t"* sebagai huruf pertama kata dasar.

Namun ada juga kondisi dimana kombinasi pen –an tidak terjadi morphing. Misalkan :

Pencarian = pen + cari + an

Berikut adalah contoh masukan untuk sebuah masalah dimana kata *keyword* memiliki akhiran dan awalan sekaligus:

Gambar 4.13 *Halaman hasil dengan keyword yang mempunyai awalan dan akhiran*

Dari tampilan hasil di atas dapat disimpulkan bahwa percobaan berhasil.

4.2 Analisa Sistem

Setelah melakukan pengujian sistem, didapatkan analisa bahwa sistem telah berjalan sesuai dengan yang diharapkan.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Berdasarkan hasil percobaan dan analisa yang dilakukan pada Proyek Akhir ini maka dapat disimpulkan bahwa :

1) Pada sistem ini, data ayat yang di dalam Al-Qur'an sudah tersimpan di dalam database.
2) Solusi dari permasalahan yang dibutuhkan *user* dapat langsung ditampilkan tiap lima ayat.
3) Dengan metode korelasi, semua masalah dapat dikenali oleh sistem.
4) Pada sistem ini, bahasa yang harus digunakan *user* harus menggunakan bahasa Indonesia baku. Jika tidak, maka data solusi yang ditampilkan tidak sesuai dengan harapan *user*.
5) Nilai korelasi menentukan makna masalah dari inputan user.

5.2 Saran

Berikut merupakan beberapa saran untuk pengembangan sistem di masa yang akan datang, berdasar pada hasil perancangan, implementasi, dan uji coba yang telah dilakukan. :

1) Menyempurnakan sistem dengan pencarian waktu yang lebih cepat
2) Sistem ini hanya dapat digunakan dengan Bahasa Indonesia yang baku, jadi di masa mendatang akan diharapkan bisa

6. Daftar Pustaka

1) *Abdul Kadir,"Bahasa Pemograman JAVA",Andi Yogyakarta, Tahun 2004*.
2) *Buya Hamka , "tafsir Al qur'an",majelis ulama indonesia, 1950*
3) *Brigitte Mathiak dan Silke Eckstein, "Five Steps to Text Mining in Biomedical Literatur", http://www.cs.tubs.de/idb/,*, 2007
4) *Help Microsoft Excel, "Correlation Method", Microsoft Corp, 2010*
5) *Igg Adwijaya, "Text Mining dan Knowledge Discovery", EMC Corporations, 2006*
6) *JISC, " What Text Mining Can Do ", http://www. jisc.ac.uk/publications*, 2006
7) *Marti Hearst," What Is The Text Mining", hearst@sims.berkeley.edu, 2003*
8) *Mohamad Hairiyanov," Data Mining Email ",ITB – Bandung, 2004*
9) *Rizal, M. Syaiful, "Pencarian ayat – ayat Al – Qur'an tentang Permasalahan Akidah berdasarkan content menggunakan Text Mining"*
10) *http://www.alqurandigital.com/, "Al – Qur'an Digital, 2004, diakses pada tanggal 12 September 2009*