MESIN PENCARI AYAT AL QURAN MENGGUNAKAN *INEXACT STRING MATCHING*

Agus Sofiyan Anwar[1], Zainal Abidin[2], Ririen Kusumawati[3]
[1]Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, UIN Maliki Malang
[2,3]Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim
Malang, Jalan Gajayana 50 Malang 65144, Telepon/Faksimile (0341) 558933
Email: agusofiyanwar@yahoo.com[1], br52s@yahoo.com[2], rizn.kusumawati@gmail.com[3]

ABSTRAK

Dengan adanya teknologi digital, al Quran yang dahulu berupa teks manual sekarang sudah dapat dijumpai versi digitalnya. Hal tersebut memicu pengembangan perangkat lunak yang membantu mendapatkan informasi dari teks al Quran, seperti: pencarian ayat berdasarkan kata, frase maupun tema, terjemahan al Quran, tafsir al Quran. Dalam kaitannya dengan pencarian ayat berdasarkan kata atau frase, pada umumnya perangkat lunak yang ada menggunakan teknik exact string matching, yaitu teknik pencarian ayat yang sesuai dengan kata inputan secara tepat. Teknik tersebut sangat sesuai jika pemakai perangkat lunak mengetikkan kata atau frase yang akan dicari dengan benar. Tetapi jika pemakai salah dalam mengetikkan kata inputan, perangkat lunak tidak memberikan solusi atau kemungkinan-kemungkinan dari ayat yang dimaksud. Penelitian ini memadukan teknik stemming dan teknik exact string matching. Stemming berperan sebagai preprocessing untuk exact string matching. Stemming digunakan untuk menemukan kata dasar dari kata berimbuhan dengan cara menghilangkan semua imbuhan baik yang terdiri dari prefiks, sufiks, infiks, konfiks, transfiks, maupun interfiks, namun pada penelitian ini hanya menghilangkan prefiks dan sufiks saja, sebagai contoh jika kata berimbuhan adalah يسطرون *maka kata dasarnya adalah* سطر*. Exact string matching adalah pencocokan string secara tepat dengan susunan karakter dalam string yang dicocokkan memiliki jumlah maupun urutan karakter yang sama, sebagai contoh kata* سطر *akan menunjukkan kecocokan hanya dengan kata* سطر*. Dalam kaitannya dengan pencarian ayat, hasil stemming akan digunakan sebagai kata kunci (keyword) pencarian pada database indeks al Quran. Perpaduan tersebut dimaksudkan untuk meningkatkan hasil pencarian ayat, dan selanjutnya dapat dikategorikan sebagai teknik inexact string matching. Hasil uji coba membuktikan bahwa teknik inexact string matching dapat diimplementasikan untuk mendukung pencarian ayat al Quran dengan nilai F-measure tertinggi pada data uji coba adalah 100 % dan nilai F-measure terendah adalah 66.66 %. Uji coba juga membuktikan bahwa teknik inexact string matching lebih banyak memberikan solusi/kemungkinan dari ayat yang dimaksud dari pada teknik exact string matching.*

Kata kunci: *Arabic Stop Word, Arabic Stemming, Exact String Matching, Inexact Matching.*

1. PENDAHULUAN

Era digital berkembang ditandai dengan munculnya tiga teknologi, yaitu: komputer, komunikasi, dan multimedia. Hal tersebut membawa kepada perubahan besar yang pada umumnya memberikan kemudahan bagi kita. Permasalahan yang dapat dipecahkan dengan cara manual kini dapat dipecahkan dengan bantuan komputer, bahkan beberapa permasalahan

yang sulit dipecahkan dengan cara manual kini dapat dipecahkan dengan bantuan komputer, seperti masalah optimasi. Pada awalnya komputer hanyalah merupakan perlengkapan elektronik yang belum berfungsi. Komputer dapat membantu manusia untuk memecahkan masalah yang komplek dan dapat membantu membedakan hal-hal yang ambigu/samar karena adanya cabang ilmu komputer, yaitu kecerdasan buatan (*artificial intelligence*).

Kecerdasan buatan dapat diterapkan diberbagai aspek kehidupan manusia, salah satu penerapannya pada dunia informatika adalah kecerdasan buatan pada mesin pencari (*search engine*). Pada awalnya bentuk mesin pencari adalah *desktop search engine* yang mencari informasi pada database yang berada di PC lokal. Seiring dengan perkembangan teknologi sekarang sudah dapat dijumpai *web search engine* yang mencari informasi pada database yang berada di server-server di seluruh dunia.

Perkembangan teknologi digital telah membawa perubahan gaya hidup, salah satu contohnya adalah al Quran yang dahulu berupa buku teks sekarang sudah dapat dijumpai versi digitalnya, baik yang berupa teks maupun yang sudah dalam bentuk database. Perubahan tersebut memicu pengembangan perangkat lunak untuk membantu mendapatkan informasi dari teks al Quran, seperti: pencarian ayat berdasarkan kata, frase maupun tema, terjemahan al Quran, tafsir al Quran.

Pada umumnya perangkat lunak pencarian ayat yang ada menggunakan teknik *exact string matching*, yaitu teknik pencarian ayat yang sesuai dengan kata inputan secara tepat. Teknik tersebut sangat sesuai jika pemakai perangkat lunak mengetikkan kata atau frase yang akan dicari dengan benar. Tetapi jika pemakai salah dalam mengetikkan kata inputan, perangkat lunak tidak memberikan solusi atau kemungkinan-kemungkinan dari ayat yang dimaksud. Suatu contoh jika potongan teks ayat yang akan dicari adalah والیتلطف, maka teknik *exact string matching* tidak dapat menemukan, karena tidak ada satupun ayat didalam al Quran yang secara tepat mengandung potongan teks ayat tersebut, yang ada adalah ولیتلطف. Kesalahan terletak pada huruf yang kedua, yaitu huruf *alif* (ا). Seharusnya tanpa huruf *alif* (ا), jadi setelah huruf yang pertama, yaitu huruf *waw* (و) langsung huruf *lam* (ل). Kesalahan pengetikan tersebut disebabkan karena pengucapan/*lafadz* ولیتلطف (ada didalam al Quran) sama dengan pengucapan/*lafadz* والیتلطف (tidak ada didalam al Quran), namun penulisannya berbeda. Dengan kasus yang sama, teknik *inexact string matching* masih dapat memberikan solusi atau kemungkinan dari ayat yang dimaksud, karena sebelum dilakukan pencarian, terlebih dahulu teks والیتلطف dicari akar katanya. Akar kata dari والیتلطف adalah لطف, sehingga jika dilakukan pencarian dengan menggunakan kata لطف tersebut, akan ditemukan didalam al Quran, karena kesalahan penulisan yaitu huruf *alif* (ا) tidak lagi berpengaruh pada proses pencarian atau dengan kata lain diabaikan.

Permasalahan yang diangkat dalam penelitian ini adalah pencarian ayat al Quran berbasis akar kata dengan menyertakan sisipan atau dengan kata lain hanya menghilangkan awalan dan akhiran saja. Hal tersebut dilakukan atas dasar bahwa jika kata yang akan dicari akarnya mengandung sisipan, dapat dipastikan bunyi akar kata atau pola akar kata yang sesungguhnya berbeda dengan kata sebelumnya yang belum dicari akar katanya karena susunan karakternya berubah.

2. DASAR TEORI

2.1. Pembentukan Kata Dalam Bahasa Arab

Afiks atau imbuhan didefinisikan sebagai bunyi yang ditambahkan pada sebuah kata untuk membentuk kata baru yang artinya berhubungan dengan kata yang pertama (Wikipedia A, 2010). Penambahan

afiks dapat dilakukan didepan, dibelakang, disisipkan, didepan dan dibelakang, atau sebagai unsur perangkai didalam kata majemuk dan berada ditengah namun berupa vokal. Penambahan yang diimbuhkan didepan disebut prefiks, yang diimbuhkan di belakang disebut sufiks, yang diimbuhkan sebagai sisipan disebut infiks, yang diimbuhkan didepan dan belakang disebut konfiks, yang menjadi unsur perangkai disebut interfiks, sedang yang berupa vokal ditengah disebut sebagai transfiks. Penambahan afiks itu sendiri juga mengalami dua pengubahan, yaitu pengubahan gramatikal dan pengubahan leksikal (Supriyadie, M, 2008).

Contoh prefiks, sufiks, infiks, konfiks, transfiks, dan interfiks dalam bahasa Arab ditunjukkan pada **tabel 1**.

2.2. *Filtering*/Penyaringan

Stop word disebutkan sebagai nama yang diberikan untuk kata-kata yang disaring, sebelum atau sesudah pengolahan bahasa alami yang dalam hal ini berupa teks. Hans Peter Luhn adalah salah satu pelopor dalam *information retrieval* (IR) yang menggunakan konsep tersebut dalam desainnya. *Stop word* dikendalikan oleh input manusia dan tidak otomatis. Tidak ada daftar yang pasti untuk *stop word* dan tidak semua alat NLP menggunakan sebuah *stoplist*. Beberapa peralatan menghindari untuk tidak menggunakannya, hal tersebut dilakukan untuk mendukung pencarian frase (Wikipedia B, 2010).

Stop word dapat dikatakan sebagai kata-kata yang memiliki frekuensi tinggi. Berikut ini contoh beberapa *arabic stop word*:

Tabel 1. Afiksasi Kata Arab

Afiks	Pengubahan Gramatikal		Pengubahan Leksikal	
Prefiks	س - ق - ط	أ - س - ق - ط	ض - ر - ب	م - ض - ر - ب
	s-q-th	a-squthu	dh-r-b	ma-dhrab
Sufiks	ج - ل - س	ج - ل - س - ت	ي - س- ر	ي - س - ر - ك
	j-l-s	jalas-tu	y-s-r	yusru-ka
Infiks	ض - ر - ب	ض - ا - ر - ب	ك - ت - ب	ك - ا - ت - ب
	dh-r-b	dha-a-raba	k-t-b	ka-a-tib
Konfiks	ل - م - س	أ - ل - م - س	ض - ر - ب	م - ض - ر - ب
	l-m-s	a-lmas-u	dh-r-b	ma-dhrab-un
Transfiks	د - ر - س	د - ا - ر - س	ت - ر - ك	م - ت - ر - و - ك
	d-r-s	d-a-r-a-s	t-r-k	m-a-t-r-u-u-k
Interfiks	وجه - حسن		وجه حسن	
	wajh-hasan		wajh-un-hasan	

Sumber: Supriyadie, M. 2008

Tabel 2. *Arabic Stop Word*

No	*Arabic Stop Word*
1	و
2	هو
3	هي
4	هم
5	هن
6	أنت
7	أنتما
8	أنتم
9	من
10	عن
11	في
12	على
13	تحت
14	فوق
15	أينما
16	بعد
17	هذه
18	ذلك
19	تلك
20	ليس

2.3. *Arabic Stemming*

Stemming merupakan suatu proses untuk menemukan kata dasar dari kata

berimbuhan dengan cara menghilangkan semua imbuhan baik yang terdiri dari prefiks, sufiks, infiks, konfiks, transfiks, maupun interfiks, namun pada penelitian ini hanya menghilangkan prefiks dan sufiks saja. Dalam kaitannya dengan pencarian ayat, hasil *stemming* akan digunakan sebagai kata kunci (*keyword*) pencarian pada database indeks al Quran.

Al Quran diturunkan di tanah Arab dan bahasanya menggunakan bahasa Arab. Oleh karena bahasa al Quran adalah bahasa Arab maka teknik *stemming* yang digunakan adalah teknik *stemming* kata arab. Berdasarkan penelitian-penelitian yang telah dilakukan, teknik *stemming* ada yang memerlukan kamus akar kata seperti Khoja *Arabic Stemmer* dan ada yang tidak memerlukannya seperti ISRI *Arabic Stemmer*. Pada penelitian ini digunakan teknik *stemming* yang tidak memerlukan kamus akar kata.

Teknik *stemming* yang digunakan adalah ISRI *Arabic Stemmer Algorithm* yang dikembangkan oleh *The Information Science Research Institute's* (ISRI). Algoritma ISRI dimulai dengan mendefinisikan sekumpulan tanda pengenal dan penggolongan atau pengelompokan *affix* (Taghva, K., Elkhoury, R. dan Coombs, J., 2005: 2), seperti ditunjukkan pada **tabel 3**.

Imbuhan-imbuhan seperti pada **tabel 3** itulah yang akan dihilangkan oleh *stemmer*. Selain itu juga didefinisikan beberapa kelompok pola kata seperti ditunjukkan pada **tabel 4**.

Tabel 3. ***Affix Sets***

Set	Description	Examples
D	Diacritics vowelizations	سٌ سِ سً سّ سْ سُ س سَ
*P*3	Prefixes of length three	ولل، وال، كال، بال
*P*2	Prefixes of length two	ال، لل
*P*1	Prefixes of length one	ل، ب، ف، س، و، ي، ت، ن، ا
*S*3	Suffixes of length three	تما، هما، تان، تين، كما
*S*2	Suffixes of length two	ون، ات، ان، ين، تن، كم، هن، نا، يا، ها، تم، كن، ني، وا، ما، هم
*S*1	Suffixes of length one	ة، ه، ي، ك، ت، ا، ن

Sumber: Taghva, K., Elkhoury, R. dan Coombs, J. 2005

Tabel 4. ***Arabic Patterns and Roots***

Set	Description	Examples
*PR*4	Length four patterns	فاعل ،فعول ،فعلة ،فعال ،فعيل، مفعل
*PR*53	Length five patterns and length three roots	تفاعل، افتعل، افعال، افاعل، فعالة، فعلان، فعولة، تفعلة، تفعيل، مفعلة، مفعول، فاعول، فواعل، مفعال، مفعيل، افعلة، فعانل، منفعل، مفتعل، فاعلة، مفاعل، يفتعل، تفتعل، فعالي، انفعل
*PR*54	Length five patterns and length four roots	تفعلل، افعلل، مفعلل، فعللة، فعلان، فعالل
*PR*63	Length six patterns and length three roots	استفعل، مفعالة، افتعال، افعوعل، مستفعل
*PR*64	Length six patterns and length four roots	افعلال، متفعلل

Sumber: Taghva, K., Elkhoury, R. dan Coombs, J. 2005

Berikut ini adalah ringkasan umum langkah-langkah dari proses *arabic stemmer algorithm*:

1. Hilangkan tanda pengenal yang mewakili huruf hidup (vokal).
2. Normalkan *hamza* yang muncul dalam beberapa bentuk pada kombinasi

berbagai huruf menjadi satu bentuk (أ).

3. Hilangkan awalan dengan panjang tiga dan dua.
4. Hilangkan penghubung (و) jika mendahului kata yang bermula dengan huruf (و).
5. Normalkan إ، أ، آ menjadi (ا).
6. Kembalikan akar kata jika panjangnya kurang dari atau sama dengan tiga huruf.
7. Mempertimbangkan empat keadaan bergantung pada panjang kata:
   a. Panjang = 4: Jika kata cocok dengan salah satu pola dari *PR*4 (**tabel 4**), ambil akar kata yang bersangkutan dan kembalikan. Sebaliknya, coba untuk menghilangkan awalan dan akhiran dari *S*1 dan *P*1. Pada perintah tersebut disediakan kata yang panjangnya tidak kurang dari tiga.
   b. Panjang = 5: Ambil akar kata dengan tiga karakter untuk kata-kata yang cocok dengan pola-pola dari *PR*53. Jika tidak ada yang cocok, coba untuk menghilangkan awalan dan akhiran, sebaliknya akar kata yang bersangkutan dengan panjang tiga dikembalikan. Jika kata panjangnya masih lima karakter, kata tersebut dicocokkan dengan *PR*54 untuk menentukan jika ia mengandung akar kata dengan panjang 4. Akar kata yang bersangkutan dikembalikan jika ditemukan.
   c. Panjang = 6: Ambil akar kata dengan panjang tiga, jika kata cocok dengan pola dari *PR*63. Sebaliknya, coba untuk menghilangkan akhiran. Jika akhiran telah dihilangkan dan menghasilkan kata dengan panjang lima, kirim balik kata lewat langkah 7b. Sebaliknya, coba untuk menghilangkan satu karakter awalan, dan jika berhasil, kirim hasil dengan panjang lima ke langkah 7b.
   d. Panjang = 7: Coba untuk menghilangkan satu karakter awalan dan akhiran. Jika berhasil, kirim hasil dengan panjang enam ke langkah 7c.

Langkah 7 pada dasarnya mengambil kata-kata panjang dan berturut-turut mencoba untuk memotong imbuhan-imbuhan karakter tunggal. Jika berhasil, ia membandingkan hasil kata yang lebih pendek dengan berbagai pola-pola pada level-level yang berbeda sampai ia cocok dengan salah satu pola dan mengambil kata yang bersangkutan, atau menjadi pendek sekali untuk menjadi akar kata yang berdiri sendiri.

Secara umum ISRI *arabic stemmer algorithm* menghilangkan imbuhan pada kata arab yang berupa awalan, akhiran dan sisipan. Melalui beberapa percobaan dan analisa kecil, penulis merasa perlu untuk mengabaikan beberapa langkah pada ISRI *arabic stemmer algorithm*. Hal ini dilakukan dengan dua alasan. Alasan pertama, tujuan utama atau *goal* dari penelitian ini adalah mencari ayat al Quran berdasarkan akar kata dan bukan semata-mata untuk mencari akar kata. Alasan kedua, teknik pencarian ayat pada penelitian ini menggunakan perpaduan antara dua teknik yang berbeda, yaitu teknik *stemming* dan teknik *exact string matching*. Dari percobaan dan analisa kecil, didapati bahwa pencarian ayat tetap dapat dilakukan walaupun hasil *stemming* belum mencapai 100 %, artinya belum sampai ditemukan akar kata yang sesungguhnya. Sebaliknya jika tanpa pengabaian beberapa langkah pada ISRI *arabic stemmer algorithm*, ada kemungkinan hasil pencarian ayat tidak sesuai dengan yang diharapkan. Salah satu contoh pengabaiannya adalah menganggap pola kata pada **tabel 4** sebagai akar kata. Jadi jika kata cocok dengan pola pada **tabel 4** maka proses pencarian akar kata

dihentikan dan kata yang bersangkutan dianggap sebagai akar kata. Penghentian tersebut disebabkan karena pola kata pada **tabel 4** kebanyakan mengandung sisipan. Pendek kata penghentian dilakukan jika kata mengandung sisipan. Jadi algoritma hanya menghilangkan awalan dan akhiran saja dan tidak menghilangkan sisipan. Hal tersebut dilakukan atas dasar bahwa jika kata yang akan dicari akarnya mengandung sisipan, dapat dipastikan bunyi akar kata atau pola akar kata yang sesungguhnya berbeda dengan kata sebelumnya yang belum dicari akar katanya karena susunan karakternya berubah, sebagai contoh kata *tafaa'ala* (تَفَاعَلَ) akar katanya adalah *fa'ala* (فَعَلَ), kata *maf'uulun* (مَفْعُوْلٌ) akar katanya adalah *fa'ala* (فَعَلَ). Pada kasus pencarian ayat, suatu contoh jika ingin mencari ayat yang mengandung kata مسطور sementara kata tersebut mengandung sisipan yaitu huruf و (cocok dengan pola kata مفعول), maka jika proses pencarian akar kata tidak dihentikan, akar kata yang didapat (akar kata yang benar) adalah سطر. Jika kata سطر digunakan sebagai kata kunci untuk pencarian ayat maka hasilnya adalah (وَالْقَلَمِ وَمَا **يَسْطُرُوْنَ**), sementara ayat yang dimaksud adalah ayat yang mengandung kata مسطور yaitu (وَكِتَابٍ **مَسْطُوْرٍ**). Sebenarnya kedua kata **يَسْطُرُوْنَ** dan **مَسْطُوْرٍ** merupakan turunan dari kata سطر atau dengan kata lain kedua kata tersebut akar katanya sama, yaitu kata سطر. Namun bagi orang awam (yang belum menguasai ilmu *shorof*), mereka melihat persamaan atau perbedaan antara kata **يَسْطُرُوْنَ** dan **مَسْطُوْرٍ** dari sisi pengucapan/*lafadz*, dan jika dilihat dari sisi pengucapan/*lafadz*-nya kedua kata **يَسْطُرُوْنَ** dan **مَسْطُوْرٍ** memang berbeda.

Inputan program berupa potongan teks ayat al Quran tanpa tanda baca (*harakat*), maka perlu dilakukan pengabaian untuk langkah pertama dari algoritma ISRI. Selanjutnya perlu dilakukan pengabaian untuk langkah kedua dan kelima dari algoritma ISRI, karena jika dinormalkan maka pola kata akan berubah.

Berdasarkan analisa mengenai panjang kata dalam ayat al Quran, maka perlu dilakukan penambahan satu keadaan (panjang kata) pada langkah ketujuh, yaitu kata dengan panjang 8 karakter.

Berikut ini adalah algoritma *arabic stemmer* dari ISRI setelah dilakukan pengabaian beberapa langkah:

1. Kembalikan kata jika panjangnya kurang dari atau sama dengan tiga huruf.
2. Hilangkan awalan dengan panjang tiga atau dua.
3. Hilangkan penghubung (و) jika mendahului kata yang bermula dengan huruf (و).
4. Kembalikan kata jika panjangnya kurang dari atau sama dengan tiga huruf.
5. Mempertimbangkan lima keadaan bergantung pada panjang kata:
   a. Panjang = 4: Jika kata cocok dengan salah satu pola dari *PR*4 (**tabel 4**), ambil kata yang bersangkutan dan kembalikan sebagai akar kata. Sebaliknya, coba untuk menghilangkan awalan atau akhiran dari *P*1 atau *S*1. Pada perintah tersebut disediakan kata yang panjangnya tidak kurang dari tiga.
   b. Panjang = 5: Jika kata cocok dengan salah satu pola dari *PR*53 (**tabel 4**), ambil kata yang bersangkutan dan kembalikan sebagai akar kata. Jika tidak ada yang cocok, coba untuk menghilangkan awalan atau akhiran dari *P*2 atau *S*2, serta kembalikan kata jika panjangnya tiga huruf. Jika kata panjangnya masih lima karakter, kata tersebut dicocokkan dengan *PR*54 dan jika ada salah satu yang cocok maka ambil kata yang bersangkutan dan kembalikan sebagai akar kata. Sebaliknya jika tidak ada yang cocok, coba untuk menghilangkan akhiran dari *S*1 dan

jika berhasil kirim balik kata yang bersangkutan ke langkah 7a. Jika belum berhasil, coba untuk menghilangkan awalan dari *P*1 dan jika berhasil kirim balik kata yang bersangkutan ke langkah 7a.

c. Panjang = 6: Jika kata cocok dengan salah satu pola dari *PR*63 (**tabel 4**), ambil kata yang bersangkutan dan kembalikan sebagai akar kata. Sebaliknya, coba untuk menghilangkan akhiran dari *S*3 dan jika berhasil ambil kata yang bersangkutan dan kembalikan sebagai akar kata. Jika belum berhasil, coba untuk menghilangkan akhiran dari *S*2 dan jika berhasil kirim balik kata yang bersangkutan ke langkah 7a. Jika belum berhasil, coba untuk menghilangkan akhiran dari *S*1 dan jika berhasil kirim balik kata yang bersangkutan ke langkah 7b. Jika belum berhasil, coba untuk menghilangkan awalan dari *P*1 dan jika berhasil kirim balik kata yang bersangkutan ke langkah 7b.
d. Panjang = 7: Coba untuk menghilangkan akhiran dari *S*3 dan jika berhasil kirim balik kata yang bersangkutan ke langkah 7a. Jika belum berhasil, coba untuk menghilangkan akhiran dari *S*2 dan jika berhasil kirim balik kata yang bersangkutan ke langkah 7b. Jika belum berhasil, coba untuk menghilangkan awalan atau akhiran dari *P*1 atau *S*1 dan jika berhasil kirim balik kata yang bersangkutan ke langkah 7c.
e. Panjang = 8: Coba untuk menghilangkan akhiran dari *S*3 dan jika berhasil kirim balik kata yang bersangkutan ke langkah 7b. Jika belum berhasil, coba untuk menghilangkan akhiran dari *S*2 dan jika berhasil kirim balik kata yang bersangkutan ke langkah 7c. Jika belum berhasil, coba untuk menghilangkan awalan atau akhiran dari *P*1 atau *S*1 dan jika berhasil kirim balik kata yang bersangkutan ke langkah 7d.

2.4. Mesin Pencari (*Search Engine*)

Search engine didefinisikan, "*A search engine is an information retrieval system designed to help find information stored on a computer system"* (Wikipedia C, 2010). Berdasarkan definisi tersebut, dapat dipahami bahwa mesin pencari adalah sistem pencarian informasi yang didesain untuk membantu menemukan informasi yang tersimpan dalam sistem komputer. Pada umumnya bentuk *search engine* adalah *Web search engine* yang mencari informasi pada *World Wide Web*. Selain itu ada juga *search engine* yang berbasis desktop. *Desktop search engine* didefinisikan, "*Desktop search is the name for the field of search tools which search the contents of a user's own computer files, rather than searching the Internet"* (Wikipedia D, 2010). Berdasarkan definisi tersebut, dapat dipahami bahwa pencarian desktop adalah nama bidang untuk peralatan pencarian yang mencari isi dari file-file komputer milik pemakai. Pencarian desktop didesain untuk menemukan informasi pada PC pemakai, yang mencakup *web browser histories, e-mail archives, text documents, sound files, images* dan *video*.

2.5. Pencocokan *String* (*String Matching*)

Pencocokan *string* diartikan sebagai sebuah permasalahan untuk menemukan pola susunan karakter *string* didalam *string* lain atau bagian dari isi teks (Nist, 2010).

Pencocokan *string* secara garis besar dapat dibedakan menjadi dua (Syaroni, M. dan Munir, R., 2005: K-7), yaitu:

1. *Exact String Matching*, merupakan pencocokan *string* secara tepat dengan susunan karakter dalam *string* yang dicocokkan memiliki jumlah maupun urutan karakter yang sama. Contoh: kata *see* akan menunjukkan kecocokan hanya dengan kata *see*.
2. *Inexact String Matching*, merupakan pencocokan *string* secara samar, maksudnya pencocokan *string* dimana *string* yang dicocokkan memiliki kemiripan yang bersifat samar, yaitu: keduanya memiliki susunan karakter yang berbeda (mungkin jumlah atau urutannya) tetapi *string-string* tersebut memiliki kemiripan baik kemiripan tekstual/penulisan (*approximate string matching*) atau kemiripan ucapan (*phonetic string matching*). *Inexact string matching* masih dapat dibagi lagi menjadi dua yaitu:
   a. *Approximate string matching* (pencocokan *string* berdasarkan kemiripan penulisan), merupakan pencocokan *string* dengan dasar kemiripan dari segi penulisannya, yaitu: jumlah karakter dan susunan karakter dalam teks. Contoh: *campuler* dengan *computer*, memiliki jumlah karakter yang sama tetapi ada dua karakter yang berbeda. Jika perbedaan dua karakter ini dapat ditoleransi sebagai sebuah kesalahan penulisan, maka dua *string* tersebut dikatakan cocok.
   b. *Phonetic string matching* (pencocokan *string* berdasarkan kemiripan ucapan), merupakan pencocokan *string* dengan dasar kemiripan dari segi pengucapannya meskipun ada perbedaan penulisan dua *string* yang dibandingkan tersebut. Contoh: *see* dengan *sea* dari tulisan berbeda tetapi dalam pengucapannya mirip sehingga dua *string* tersebut dianggap cocok.

Exact string matching bermanfaat jika pengguna ingin mencari *string* dalam dokumen atau teks yang sama persis dengan *string* masukan. Tetapi jika pengguna menginginkan pencarian *string* yang mendekati dengan *string* masukan atau terjadi kesalahan penulisan *string* masukan maupun dokumen objek pencarian, maka *inexact string matching* yang lebih bermanfaat. Beberapa algoritma *exact string matching* antara lain: algoritma *brute force*, *knuth-morris-pratt*, *boyer-moore* dan yang lainnya. Beberapa algoritma *inexact string matching* antara lain: *levenshtein edit distance*, *soundex*, *metaphone*, *caverphone* dan yang lainnya.

Konsep *inexact string matching* pada penelitian ini adalah memadukan antara teknik *stemming* dan teknik *exact string matching*. *Stemming* berperan sebagai *preprocessing* untuk *exact string matching*. *Stemming* digunakan untuk menemukan kata dasar dari kata berimbuhan dengan cara menghilangkan semua imbuhan baik yang terdiri dari prefiks, sufiks, infiks, konfiks, transfiks, maupun interfiks, namun pada penelitian ini hanya menghilangkan prefiks dan sufiks saja, sebagai contoh jika kata berimbuhan adalah يسطرون maka kata dasarnya adalah سطر. Hasil *stemming* yang berupa kata dasar tersebut akan digunakan sebagai kata kunci (*keyword*) pencarian pada database indeks al Quran dengan menggunakan teknik *exact string matching*.

2.6. Algoritma *Brute Force*

Munir (2004: 2) mengasumsikan bahwa *text* berada di dalam *array* *T*[1..*n*] dan *pattern* berada di dalam *array* *P*[1..*m*], maka algoritma *brute force* pencocokan *string* adalah sebagai berikut:

1. Mula-mula *pattern P* dicocokkan pada awal teks *T*.
2. Dengan bergerak dari kiri ke kanan, bandingkan setiap karakter di dalam *pattern P* dengan karakter yang bersesuaian di dalam teks *T* sampai:
   a. Semua karakter yang dibandingkan cocok atau sama (pencarian berhasil).
   b. Dijumpai sebuah ketidakcocokan karakter (pencarian belum berhasil).

Bila *pattern P* belum ditemukan kecocokannya dan teks *T* belum habis, geser *pattern P* satu karakter ke kanan dan ulangi langkah 2.

Contoh:

Text : We have good pleasure today
Pattern : have

We **have** good pleasure today
s = 0 have
s = 1 have
s = 2 have
s = 3 **have**

3. METODOLOGI PENELITIAN

Proses pencarian ayat al Quran terdiri dari tiga proses utama, yaitu: *filtering*, *inexact string matching* dan *database connection*.

Dalam proses *filtering*, teks potongan ayat al Quran yang telah dimasukkan, diuraikan menjadi kata-kata yang terpisah. Kata-kata terpisah tersebut dicocokkan dengan daftar *arabic stop word* yang telah didefinisikan sebelumnya. Selanjutnya diambil satu kata terdepan yang tidak cocok dengan daftar *arabic stop word*.

Dalam proses *inexact string matching*, kata hasil *filtering* dihilangkan awalannnya dengan panjang 3 atau 2 huruf. Jika kata bermula dengan huruf *waw* dan kata tersebut didahului oleh penghubung *waw*, maka penghubung tersebut dihilangkan. Jika panjang kata lebih dari 3 huruf, maka dipertimbangkan berdasarkan panjang huruf. Kata yang telah dihilangkan awalan dan akhirannya, selanjutnya diambil dan dikembalikan sebagai akar kata. Setelah ditemukan akar katanya, selanjutnya menginisialisasi *array* dengan panjang sesuai dengan jumlah karakter pada akar kata untuk menyimpan indeks posisi karakter yang cocok. Akar kata yang telah ditemukan kemudian dicocokkan dengan teks ayat al Quran yang ada pada database indeks al Quran. Jika akar kata ditemukan pada teks ayat al Quran, maka *array* yang telah diinisialisasikan sebelumnya diisi dengan indeks posisi karakter yang cocok.

Database Connection adalah proses yang menghubungkan program dengan database indeks al Quran.

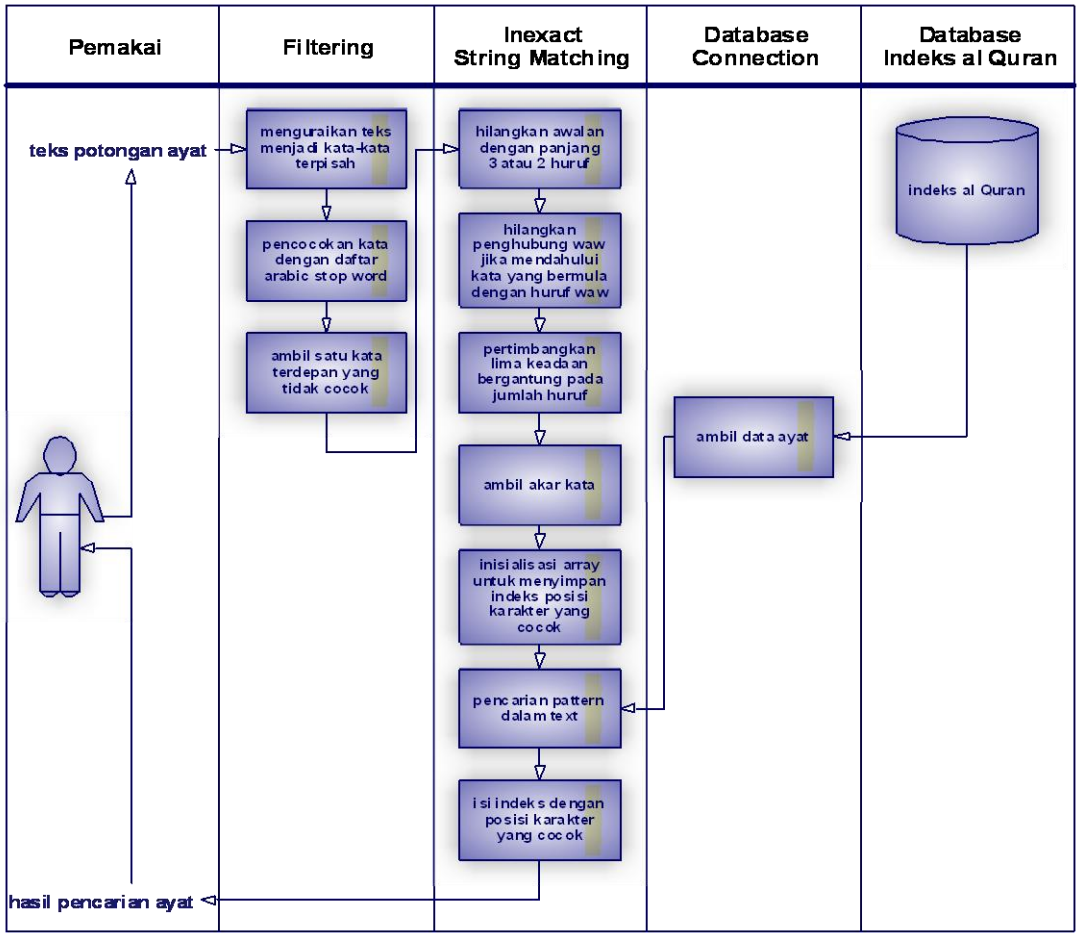
Block diagram proses pencarian ayat al Quran dapat dilihat pada **gambar 1**.

4. UJI COBA

Uji coba perbandingan dilakukan antara hasil *inexact string matching* dengan kamus *fathurrahman* dan perbandingan antara hasil *exact string matching* dengan kamus *fathurrahman*. Kamus *fathurrahman* yang ditulis oleh Al Husni adalah kamus untuk mencari ayat al Quran yang disusun berdasarkan akar kata. Perbandingan hasil *inexact string matching* dengan kamus tersebut tidak lain merupakan uji validasi hasil pencarian ayat.

Hasil uji coba perbandingan antara *inexact string matching* dengan kamus *fathurrahman* menunjukkan bahwa nilai *F-measure* tertinggi adalah 100 % dan nilai *F-measure* terendah adalah 66.66 %. Dari hasil uji coba dapat diketahui bahwa masih ada hasil pencarian ayat yang tidak tepat, hal tersebut disebabkan karena susunan karakternya sama namun bukan berasal dari akar kata yang sama, yaitu kata يلطف dan kata الطفل. Data uji coba perbandingan antara hasil *inexact string matching* dengan kamus *fathurrahman* dapat dilihat pada **tabel 5**.

Hasil uji coba perbandingan antara *exact string matching* dengan kamus *fathurrahman* menunjukkan bahwa nilai *F-measure* tertinggi adalah 100 % dan nilai *F-measure* terendah adalah 0 %. Dari hasil uji coba dapat diketahui bahwa solusi/kemungkinan yang dihasilkan oleh teknik *exact string matching* lebih sedikit dari pada teknik *inexact string matching*. Bahkan untuk kasus tertentu, dimana pemakai salah dalam mengetikkan teks potongan ayat, teknik *exact string matching* tidak dapat memberikan solusi/kemungkinan dari ayat yang dimaksud, seperti teks زادتهم ايمانا seharusnya diketik زادتهم إيمانا. Kesalahan terletak pada kata ايمانا yang mana huruf pertama seharusnya meggunakan huruf *hamzah* dibawah *alif* (إ) dan bukan huruf *alif* (ا), jadi yang benar adalah إيمانا. Data uji coba perbandingan antara hasil *exact string matching* dengan kamus *fathurrahman* dapat dilihat pada **tabel 6**.



Gambar 1. *Block Diagram* Proses Pencarian Ayat

Tabel 5. Perbandingan Hasil *Inexact String Matching* Dengan Kamus *Fathurrahman*

No	Potongan Teks Ayat	Hasil Perbandingan						
		Inexact String Matching		Kamus *Fathurrahman*		Keterangan		
		Rincian	Jumlah	Rincian	Jumlah	*Precision*	*Recall*	*F-measure*
1	ألم نشرح لك صدرك	**[6 : 125] [16 : 106] [20 : 25] [39 : 22] [94 : 1]**	5	**[16 : 106] [39 : 22] [6 : 125] [94 : 1] [20 : 25]**	5	100 %	100 %	100 %
2	والله سريع الحساب	**[2 : 202] [3 : 19] [3 : 199] [5 : 4] [6 : 165] [7 : 167] [13 : 41] [14 : 51] [24 : 39] [40 : 17]**	10	**[2 : 202] [24 : 39] [3 : 19] [3 : 199] [5 : 4] [14 : 51] [40 : 17] [6 : 165] [7 : 167] [13 : 41]**	10	100 %	100 %	100 %
3	زادتهم ايمانا	**[8 : 2] [9 : 124] [9 : 125]**	3	**[9 : 124] [8 : 2] [9 : 125]**	3	100 %	100 %	100 %
4	إنه كان حوبا كبيرا	**[4 : 2]**	1	**[4 : 2]**	1	100 %	100 %	100 %
5	وليتلطف	**[18 : 19]** [24 : 31]	2	**[18 : 19]**	1	50 %	100 %	66.66 %
6	ولا على الأعرج حرج	**[24 : 61] [48 : 17]**	2	**[24 : 61] [48 : 17]**	2	100 %	100 %	100 %
7	إن الخمر	**[2 : 219] [5 : 90] [5 : 91] [12 : 36] [12 : 41] [24 : 31] [47 : 15]**	7	**[47 : 15] [12 : 36] [12 : 41] [5 : 90] [2 : 219] [5 : 91] [24 : 31]**	7	100 %	100 %	100 %

Tabel 6. Perbandingan Hasil *Inexact String Matching* Dengan *Exact String Matching*

No	Potongan Teks Ayat	Hasil Perbandingan						
		Exact String Matching		Kamus *Fathurrahman*		Keterangan		
		Rincian	Jumlah	Rincian	Jumlah	*Precision*	*Recall*	*F-measure*
1	ألم نشرح لك صدرك	**[94 : 1]**	1	[16 : 106] [39 : 22] [6 : 125] **[94 : 1]** [20 : 25]	5	100 %	20 %	33.33 %
2	والله سريع الحساب	**[2 : 202] [24 : 39]**	2	**[2 : 202] [24 : 39]** [3 : 19] [3 : 199] [5 : 4] [14 : 51] [40 : 17] [6 : 165] [7 : 167] [13 : 41]	10	100 %	20 %	33.33 %
3	زادتهم ايمانا		0	[9 : 124] [8 : 2] [9 :	3	0 %	0 %	0 %

				125]				
4	إنه كان حوبا كبيرا	**[4 : 2]**	1	**[4 : 2]**	1	100 %	100 %	100 %
5	واليتلطف		0	[18 : 19]	1	0 %	0 %	0 %
6	ولا على الأعرج حرج	**[24 : 61]** **[48 : 17]**	2	**[24 : 61]** **[48 : 17]**	2	100 %	100 %	100 %
7	إن الخمر		0	[47 : 15] [12 : 36] [12 : 41] [5 : 90] [2 : 219] [5 : 91] [24 : 31]	7	0 %	0 %	0 %

5. KESIMPULAN

Berdasarkan hasil uji coba yang telah dilakukan dapat ditarik kesimpulan bahwa:

1. Teknik *inexact string matching* dapat diimplementasikan untuk mendukung pencarian ayat al Quran dengan nilai *F-measure* tertinggi pada data uji coba adalah 100 % dan nilai *F-measure* terendah adalah 66.66 %. Hasil uji coba menunjukkan bahwa masih ada hasil pencarian ayat yang tidak tepat, hal tersebut disebabkan karena susunan karakternya sama namun bukan berasal dari akar kata yang sama, yaitu kata يتلطف dan kata الطفل.
2. Solusi/kemungkinan yang dihasilkan oleh teknik *exact string matching* lebih sedikit dari pada teknik *inexact string matching*. Bahkan untuk kasus tertentu, dimana pemakai salah dalam mengetikkan teks potongan ayat, teknik *exact string matching* tidak dapat memberikan solusi/kemungkinan dari ayat yang dimaksud, seperti teks زادتهم ايمانا seharusnya diketik زادتهم إيمانا.

6. SARAN

Bagi peneliti selanjutnya diharapkan melakukan eksperimen lebih jauh lagi untuk meningkatkan hasil pencarian ayat dengan menghilangkan sisipan pada proses pencarian akar kata.

Penelitian ini diharapkan dapat memicu penelitian-penelitian/riset-riset lain yang berbasis pada *arabic stemmer algorithm* maupun *string matching algorithm*.

DAFTAR PUSTAKA

1. Al Husni. *Fathu-ar-Rahman Lithalibi Ayati-al-Quran*. Indonesia: Maktabah Dahlan.
2. Munir, R. 2004. *Strategi Algoritmik, Algoritma Pencarian String (String Matching)*. Bandung: Departemen Teknik Informatika Institut Teknologi Bandung.
3. Nist. 2010. *Dictionary of Algorithms and Data Structures. National Institute of Standards and Technology*. *http://www.itl.nist.gov/div897/sqg/dads/*. Diakses tanggal 09 Juni 2010.
4. Supriyadie, M. 2008. *Pembentukan Kata Dalam Bahasa Arab*. *http://supriyadie.wordpress.com/2008/06/11/pembentukan-kata-dalam-bahasa-arab*. Diakses tanggal 06 Juni 2010.
5. Syaroni, M. dan Munir, R. 2005. *Pencocokan String Berdasarkan Kemiripan Ucapan (Phonetic String Matching) Dalam Bahasa Inggris*. Yogyakarta: Seminar Nasional Aplikasi Teknologi Informasi (SNATI).
6. Taghva, K., Elkhoury, R. dan Coombs, J. 2005. *Arabic Stemming Without A Root Dictionary*. Las Vegas: Information Science

Research Institute University of Nevada.
7. Wikipedia A. 2010. *Afiks*. *http://id.wikipedia.org/wiki/Afiks*. Diakses tanggal 06 Juni 2010.
8. Wikipedia B. 2010. *Stop words*. *http://en.wikipedia.org/wiki/Stop_words*. Diakses tanggal 12 Juli 2010.
9. Wikipedia C. 2010. *Search Engine (Computing)*. *http://en.wikipedia.org/wiki/Search_engine_(computing)*. Diakses tanggal 24 Mei 2010.
10. Wikipedia D. 2010. *Desktop Search*. *http://en.wikipedia.org/wiki/Desktop_search*. Diakses tanggal 24 Mei 2010.