

Pembangunan Sistem Pencarian Ayat Al-Qur'an Berbasis Mobile Menggunakan Algoritma Metaphone Berdasarkan Kemiripan Pengucapan dan Algoritma Levenshtein Distance Untuk Mengukur Tingkat Similarity

Eka Putri¹, Dr. Moch. Arif Bijaksana², Said Al Faraby³

¹Fakultas Informatika, Universitas Telkom, Bandung

¹ekaputeri@students.telkomuniversity.ac.id, ²arifbijaksana@telkomuniversity.ac.id,

³saidalfaraby@telkomuniversity.ac.id

Abstrak

Al-Qur'an merupakan kitab suci utama bagi umat Islam yang ditulis menggunakan bahasa Arab. Sebagai kitab suci yang diturunkan dalam bahasa Arab, menyebabkan orang awam yang tidak memahami bahasa Arab semakin sulit untuk melakukan pencarian ayat Al-Qur'an. Maka dari itu dibutuhkan sistem yang dapat membantu pengguna dalam melakukan pencarian ayat Al-Qur'an yang ingin dicarinya. Sedangkan sistem yang ada saat ini hanya melakukan pencarian dengan teknik Exact String Matching. Dimana jika terdapat kesalahan dalam penulisan, maka sistem tersebut tidak akan menampilkan hasil yang dicari. Dan saat ini sistem berbasis mobile semakin berkembang, baik itu untuk informasi maupun hal-hal lainnya. Hal ini disebabkan sistem berbasis mobile lebih cepat, mudah dan efisien. Oleh karena itu untuk mengatasi permasalahan-permasalahan yang ada, dibutuhkan sistem pencarian ayat Al-Qur'an berbasis mobile dengan teknik fonetik atau pencocokan kata berdasarkan pengucapan (Phonetic String Matching) dengan menggunakan algoritma metaphone serta dapat diukur tingkat similaritinya dengan menggunakan algoritma levenshtein distance. Hasil dari penelitian tugas akhir ini didapatkan nilai MAP (Mean Average Precision) 76.34% , nilai recall sebesar 81.99%, serta nilai korelasi sebesar 89.47%.

Kata kunci : Al-Qur'an, Phonetic String Matching, Metaphone, Levenshtein Distance

Abstract

The Qur'an is the main holy book for Muslims which is written in Arabic. As a holy book which is revealed in Arabic, it makes it difficult for lay people who do not understand Arabic to search for the Qur'anic verses. Therefore we need a system that can help users in searching for the Qur'anic verses that they want to look for. While the current system only searches with the Exact String Matching technique. Where if there is an error in writing, then the system will not display the results sought. And now mobile-based systems are growing, both for information and other things. This is because mobile-based systems are faster, easier and more efficient. Therefore, to overcome the existing problems, we need a mobile-based Al-Qur'an verse search system with phonetic techniques or Phonetic String Matching using metaphone algorithms and the similarity level can be measured using the levenshtein distance algorithm. The results of this thesis research obtained the value of MAP (Mean Average Precision) 76.34 %, recall value of 81.99 %, and the correlation value of 89.47 %.

Keywords: Al-Qur'an, Phonetic String Matching, Metaphone, Levenshtein Distance

1. Pendahuluan

Latar Belakang

Al-Qur'an adalah kitab suci utama dalam agama Islam, yang umat muslim percaya bahwa kitab ini diturunkan oleh Allah SWT dan diwahyukan dalam bahasa Arab kepada Nabi Muhammad SAW. Sebagai wahyu yang diturunkan dengan bahasa Arab, maka proses pencarian ayat Al-Qur'an pun harus menggunakan bahasa Arab. Namun hal tersebut cukup menyulitkan bagi seseorang yang tidak terlalu paham atau bahkan tidak bisa menuliskan bahasa Arab. Dan dengan menggunakan teknik Exact String Matching, yaitu pencocokan string secara tepat dengan susunan karakter dalam string yang dicocokkan memiliki jumlah maupun urutan karakter dalam string yang sama. Teknik tersebut sangat sesuai jika pengguna mengetikkan kata yang akan dicari dengan benar. Tetapi jika pengguna mengetikkan kata inputan dengan salah, maka sistem tidak akan mengeluarkan outputan yang diinginkan oleh pengguna tersebut. Sebagai contoh apabila pengguna ingin mencari potongan ayat "Allohuakbar", maka teknik

Exact String Matching tidak dapat menemukan potongan ayat yang dicari pengguna tersebut. Dikarenakan tidak adanya potongan ayat dalam Al-Qur'an transliterasi Latin yang secara tepat mengandung potongan ayat tersebut, karena potongan ayat yang ada dalam Al-Qur'an transliterasi Latin itu adalah "Allahu Akbar". Kesalahan penge-tikan tersebut disebabkan oleh pengucapan/lafadz yang sama tetapi cara penulisannya berbeda. Dan seiring dengan perkembangan teknologi informasi pada saat ini khususnya dalam penggunaan perangkat mobile, tidak menutup kemungkinan akan menambah permasalahan yang ada.

Oleh karena itu, untuk mengatasi permasalahan-permasalahan diatas. Maka pada penelitian ini dibutuhkan sistem pencarian ayat Al-Qur'an berbasis mobile dengan teknik fonetik atau pencocokan kata berdasarkan pengucapan (Phonetic String Matching) yang lebih sesuai dengan pelafalan orang Indonesia dengan menggunakan algoritma metaphone dan algoritma levenshtein distance untuk mengukur tingkat similaritynya. Pada penelitian tugas akhir ini, penulis memilih menggunakan algoritma metaphone dikarenakan dalam proses pengkodean fonetis dilakukan dengan menerapkan aturan yang biasa digunakan dalam pengucapan. Sehingga semua karakter dari kode hasil pengkodean fonetis berupa huruf dan lebih mendekati kebenaran. Sedangkan penulis memilih algoritma levenshtein distance dikarenakan algoritma levenshtein distance dapat digunakan untuk mengukur nilai kesamaan atau kemiripan antara dua buah kata (string).

Topik dan Batasannya

Berdasarkan latar belakang yang telah diurai di atas, maka permasalahan yang akan dikaji dalam penelitian ini dapat di rumuskan sebagai berikut :

1. Bagaimana merancang sistem pencarian ayat Al-Qur'an berbasis mobile menggunakan pencocokan kata berdasarkan kemiripan pengucapan dengan algoritma metaphone dan algoritma levenshtein distance untuk mengukur tingkat similarity?
2. Bagaimana perhitungan performansi sistem pencarian ayat Al-Qur'an menggunakan algoritma metaphone dan levenshtein distance untuk mengukur tingkat similarity?

Adapun yang menjadi batasan masalah dari penelitian ini adalah:

1. Algoritma metaphone yang diusulkan hanya dapat diterapkan pada query masukan string latin.
2. Dataset yang digunakan hanya teks Al-Qur'an Juz 1, Juz 2, Juz 3, Juz 4, Juz 5 dan Juz 30.
3. Menggunakan aturan alih aksara Arab-Latin sesuai dengan pelafalan orang Indonesia.
4. Sistem yang dibuat hanya menerapkan hukum tajwid iqlab, ikhfa dan idhgam.
5. Sistem yang dibuat belum bisa mengatasi query lebih dari satu kata yang antar katanya tersambung.

Tujuan

Penelitian ini bertujuan untuk membangun sistem pencarian ayat Al-Qur'an berbasis mobile berdasarkan kemiripan pengucapan dengan menggunakan algoritma metaphone dan levenstein distance untuk mengukur tingkat similaritynya. Serta untuk mengukur performansi yang dihasilkan oleh sistem tersebut.

Organisasi Tulisan

Adapun organisasi tulisan yang ada pada jurnal tugas akhir ini adalah pendahuluan, studi terkait, sistem yang dibangun, evaluasi, kesimpulan dan daftar pustaka. Dimana pendahuluan menjelaskan perencanaan serta isi dari penelitian secara keseluruhan, meliputi latar belakang, rumusan masalah, batasan masalah dan tujuan. Studi terkait memuat studi atau literatur yang pernah dilakukan untuk mendukung penelitian yaitu mengenai phonetic string matching, metode metaphone dan levenshtein distance. Sistem yang dibangun berisikan perancangan dan sistem yang dihasilkan dalam penelitian. Evaluasi menjelaskan hasil dari pengujian yang dilakukan, analisis terhadap hasil tersebut dan juga penjelasan skenario pengujian. Kesimpulan menjelaskan kesimpulan yang diambil dari hasil pengujian dan analisis hasil pengujian serta memuat saran yang dapat dijadikan dasar dalam pengembangan penelitian selanjutnya. Dan daftar pustaka berisi literatur-literatur yang membantu pengerjaan penelitian ini dan lampiran yang berisi data-data pendukung.

2. Studi Terkait

2.1 Al-Qur'an

Al-Qur'an merupakan wahyu Allah SWT yang disampaikan kepada Nabi Muhammad SAW sebagai pedoman hidup umat manusia. Secara bahasa Al-Qur'an artinya bacaan atau yang dibaca [1]. Al-Qur'an terdiri atas 114

surat, 30 juz dan 6236 ayat [10]. Secara umum, Al-Qur'an terbagi menjadi 30 bagian yang dikenal dengan nama juz. Dan setiap surat dalam Al-Qur'an terdiri atas sejumlah ayat, mulai dari surat-surat yang terdiri atas 3 ayat yaitu surat Al-Kautsar, An-Nasr dan Al-Asr hingga surah yang mencapai 286 ayat yaitu surat Al-Baqarah.

2.2 Pedoman Alih Aksara Arab ke Latin

Alih aksara huruf Arab ke huruf Latin dalam ejaan bahasa Indonesia diatur dalam Surat Keputusan Bersama Menteri Agama dan Menteri P dan K Nomor 158 tahun 1987 – Nomor: 0543 b/u1987 [7]. Dimana terdapat banyak kata dari bahasa Arab dengan aneka ragam lafal dan tulisan walaupun berasal dari kata yang sama. Beberapa penulis juga menggunakan sumber berbahasa Inggris beserta alih aksaranya. Dalam bahasa Inggris, yang sering digunakan adalah alih aksara Qalam. Kadang-kadang, perbedaan alih aksara tersebut dengan alih aksara kritis Indonesia menimbulkan kesalahpahaman dan kekeliruan pembacaan.

2.3 Pencocokan String

Pencocokan String menurut Dictionary of Algorithms and Data Structures, National Institute of Standards and Technology (NIST), diartikan sebagai sebuah permasalahan untuk menemukan pola susunan karakter string di dalam string lain atau bagian dari isi teks [9].

1. Exact string matching, merupakan pencocokan string secara tepat dengan susunan karakter dalam string yang dicocokkan memiliki jumlah maupun urutan karakter dalam string yang sama.
2. Inexact string matching atau Fuzzy string matching, maksudnya pencocokan string secara samar, maksudnya pencocokan string dimana string yang dicocokkan memiliki kemiripan dimana 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).

2.4 Phonetic String Matching

Pencocokan string berdasarkan (phonetic string matching) merupakan pencocokan string dengan dasar kemiripan dari segi pengucapannya meskipun ada perbedaan penulisan dua string yang dibandingkan tersebut. Dan beberapa algoritma phonetic string matching antara lain adalah soundex, metaphone, caverphone, phonex, NYSI-IS, Jaro-Winkler, dan lain-lain [9].

1. Algoritma Soundex, Algoritma Soundex pertama kali dipatenkan oleh Margaret O'Dell and Robert C. Russell pada tahun 1918, tetapi algoritma ini kemudian disempurnakan lagi. Algoritma soundex menghasilkan kode fonetik dengan panjang empat karakter untuk semua panjang string masukan.
2. Algoritma Metaphone, merupakan algoritma yang melakukan penanganan secara khusus terhadap setiap fonem (satuan bunyi bahasa) dalam kata.
3. Algoritma Double Metaphone, merupakan algoritma penyempurnaan dari metaphone dan algoritma terdahulu, yang ditulis oleh Lawrence Philips. Ditulis double karena algoritma ini mengembalikan primary dan secondary code untuk sebuah string.
4. Algoritma Caverphone, merupakan algoritma yang masih baru dan berusaha menyempurnakan algoritma-algoritma terdahulu.

2.5 Algoritma Metaphone

Algoritma metaphone dipublikasikan oleh Lawrence Philips dalam sebuah artikel berjudul "Hanging on the Metaphone" dalam jurnal "Computer Language" vol 7 nomor 12, Desember 1990, pp. 39-43. Algoritma metaphone memiliki beberapa kelebihan dibandingkan dengan algoritma-algoritma lainnya, diantaranya adalah algoritma metaphone menghasilkan kode fonetis yang panjang karakternya berbeda sesuai dengan panjang string masukan [9]. Selain itu dalam proses pengkodean fonetis dilakukan dengan menerapkan aturan yang biasa digunakan dalam pengucapan. Sehingga semua karakter dari hasil pengkodean fonetis berupa huruf dan lebih mendekati kebenaran.

Langkah-langkah dalam algoritma dapat dilihat pada buku "Practical Algorithms for Programmers" [2]. Pada algoritma metaphone vokal diabaikan kecuali pada awal kata. Dalam algoritma ini, mereduksi alfabet menjadi enam belas suara konsonan yaitu : B, X, S, K, J, T, F, H, L, M, N, P, R, O, W, dan Y. Dan berikut ini merupakan aturan dalam pemberian kode fonetis untuk algoritma metaphone yang telah diubah sesuai dengan pelafalan dalam aksara Arab yang telah ditransliterasikan dalam aksara Latin dan juga pemadanan ke kode fonetis:

Tabel 1. Pemadanan ke Kode Fonetis

Aksara Latin	Padanan
b	b
c	c
d, dh	d
f, p, v	f
g, gh	g
h, kh	h
k, q	k
l	l
m	m
r	r
w	w
y	y
s, sh, sy, ts	s
t, th	t
j, z, zh, dz, zy	z
nb (Iqlab)	mb
n, ng (Ikhfa)	n
ny, nn, nm, nw, nl, nr (Idhgam)	y, n, m, w, l, r

2.6 Levenshtein Distance

Algoritma Levenshtein Distance atau Edit Distance adalah algoritma pencarian jumlah perbedaan string yang ditemukan oleh Vladimir Levenshtein, seorang ilmuwan Rusia, pada tahun 1965 [3]. Algoritma ini merupakan perkembangan dari dynamic programming dimana algoritma ini melakukan perhitungan dengan menggunakan matriks pembandingan dan memberi output jumlah perbedaan di antara dua string yang disebut dengan distance dengan menggunakan fungsi [5]:

$$lev\ a, b(i, j) = \begin{cases} \max(i, j) & \text{if } \min(i, j) = 0 \\ \min \begin{cases} lev\ a, b(i-1, j) + 1 \\ lev\ a, b(i, j-1) + 1 \\ lev\ a, b(i-1, j-1) + 1 (a_i \neq b_j) \end{cases} & \text{otherwise} \end{cases}$$

Gambar 1. Fungsi Levenshtein Distance

Nilai distance antara dua string ini ditentukan oleh jumlah minimum dari operasi-operasi perubahan yang diperlukan untuk melakukan transformasi dari suatu string menjadi string lainnya. Operasi-operasi tersebut adalah penyisipan (insertion), penghapusan (deletion), atau penukaran (substitution) [6].

2.7 Similaritas

Pada penelitian ini dilakukan lima tahap untuk melakukan penghitungan kedekatan antara string query dengan korpus Al-Qur'an yang telah dicocokkan kode fonetisnya. Kelima tahapan ini merupakan hasil dari representasi dari algoritma levenshtein distance. Dan berikut ini merupakan kelima tahapannya [4]:

1. Menghitung Nilai ED (Edit Distance) Melakukan perhitungan nilai edit distance pada kata atau string yang berasal dari kata awal di dokumen yang telah dicocokkan kode fonetisnya. Dimana untuk menghitung jaraknya menggunakan fungsi levenshtein distance.
2. Menghitung Nilai PED (Phonetic Edit Distance) Melakukan perhitungan nilai phonetic edit distance pada kata atau string yang sudah diubah kedalam kode fonetis dan telah dicocokkan kode fonetisnya.

3. Menghitung Nilai EDscr (Edit Distance Score)

$$EDscr = \frac{\maxLeng(s1, s2) - ED}{\maxLeng(s1, s2)} \quad (1)$$

Dimana:

ED = hasil dari nilai ED

$\maxLeng(s1,s2)$ = nilai string terpanjang (antara kode fonetis string query masukan dan kode fonetis dataset).

4. Menghitung Nilai PEDscr (Phonetic Edit Distance Score)

$$PEDscr = \frac{\maxLeng(s1, s2) - PED}{\maxLeng(s1, s2)} \quad (2)$$

Dimana:

PED = hasil dari nilai PED

$\maxLeng(s1,s2)$ = nilai string terpanjang (antara kode fonetis string query masukan dan kode fonetis dataset).

5. Menghitung Nilai FOM (Figure of Merit)

$$FOM = \frac{PEDscr + \frac{EDscr}{10}}{1, 1} \quad (3)$$

Dimana:

EDscr = hasil dari nilai EDscr

PEDscr = hasil dari nilai PEDscr

2.8 Matriks Evaluasi

Matriks evaluasi dilakukan untuk mengukur nilai performansi dari sistem yang telah dibuat. Parameter pengujian yang digunakan untuk evaluasi pada penelitian ini adalah mengetahui nilai MAP (Mean Average Precision), recall dan korelasi. MAP, recall dan korelasi dapat dinyatakan sebagai berikut:

$$AP = \frac{1}{|R|} \cdot \sum_{i=1}^n prec(i) \cdot relevance(i) \quad (4)$$

Dimana:

R = jumlah dokumen yang relevan

$prec(i)$ = presisi di top i dokumen

$relevance(i)$ = 1 jika relevan dan 0 jika tidak relevan

$$MAP = \frac{1}{|Q|} \cdot \sum_{Q_1 \in Q} AP(Q_1) \quad (5)$$

Dimana:

Q = jumlah query pada AP

$AP(Q_1)$ = hasil AP pada setiap query

$$Recall = \frac{|TP|}{|TP| + |FN|} \quad (6)$$

Dimana:

TP = jumlah output yang dihasilkan sistem

FN = jumlah yang benar (menurut manual) tapi tidak dimunculkan sistem

$$Korelasi = \frac{\sum xy}{\sqrt{(\sum x^2)(\sum y^2)}} \quad (7)$$

Dimana:

x = variabel data 1

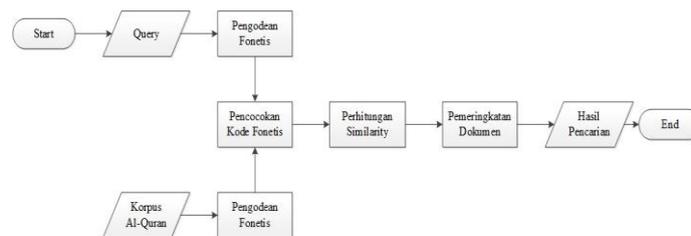
y = variabel data 2

2.9 Android

Menurut Nazrudin (2012), Android adalah sebuah sistem operasi pada handphone yang bersifat terbuka dan berbasis pada sistem operasi Linux. Android bisa digunakan oleh setiap orang yang ingin menggunakannya pada perangkat mereka. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang akan digunakan untuk bermacam piranti bergerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat piranti lunak ponsel. Kemudian untuk mengembangkan android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan piranti keras, piranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile dan Nvidia [8].

3. Sistem yang Dibangun

Dalam penelitian ini dibangun sistem pencarian ayat Al-Qur'an berdasarkan kemiripan pengucapan. Dan berikut ini merupakan gambaran umum dari sistem yang telah dibangun:



Gambar 2. Gambaran Umum Sistem

3.1 Dataset

Dataset yang digunakan pada penelitian tugas akhir ini adalah korpus teks Al-Qur'an dengan aksara latin atau Al-Qur'an tranliterasi latin sesuai dengan pelafalan orang indonesia. Pada penelitian ini surat Al-Qur'an pada Juz 1, Juz 2, Juz 3, Juz 4, Juz 5 dan juz 30 dipilih sebagai dataset yang akan digunakan sistem. Dataset tersebut akan dibangun dalam bentuk text file dengan format .txt. Dalam file tersebut terdapat beberapa atribut yaitu nama surat, nomor surat, nomor ayat dan teks ayat Al-Qur'an aksara Latin. Serta file dengan format .txt yang berisi atribut nomor surat, nomor ayat dan teks ayat Al-Qur'an aksara Arab.

3.2 Pengkodean Fonetis

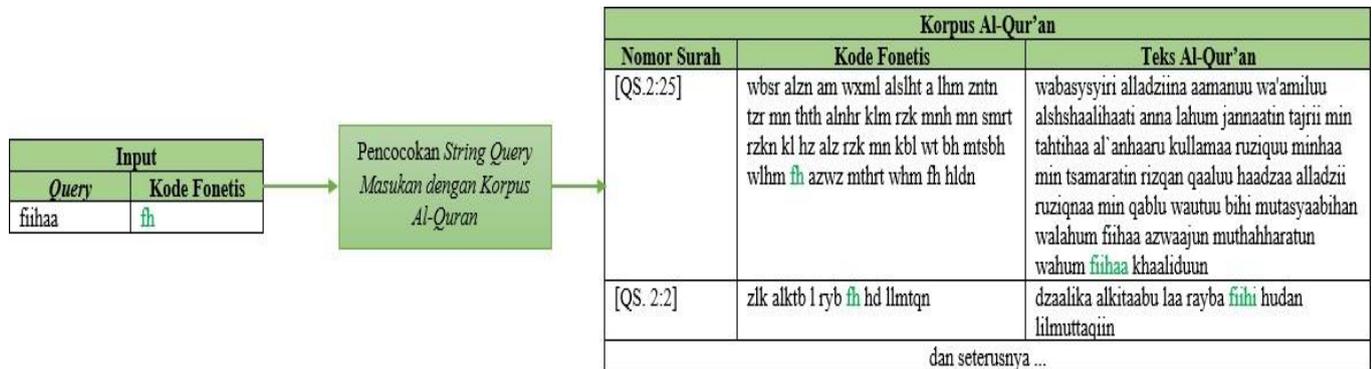
Proses yang terjadi dalam tahap ini adalah pemberian kode fonetis sesuai dengan aturan algoritma metaphone yang dilakukan pada string query masukan dan string yang ada dalam korpus Al-Qur'an. Contoh: fiihaa

1. Langkah pertama, huruf pertama dari string query masukann "fiihaa" adalah f. Dimana hasil kode dari huruf f adalah f.
2. Langkah kedua, lakukan pengkodean pada huruf berikutnya yaitu i. Dikarenakan huruf i merupakan huruf vocal dan letak dari huruf i tidak diawal kata, maka huruf i diabaikan.
3. Langkah ketiga, dikarenakan masih terdapat huruf i. Maka huruf i diabaikan.
4. Langkah keempat, terdapat huruf h. Dan hasil kode dari huruf h adalah h.
5. Langkah kelima, terdapat huruf vocal kembali yaitu a. Maka dari itu huruf a diabaikan.

6. Langkah keenam, dikarenakan masih terdapat huruf a. Maka huruf a diabaikan. Jadi hasil dari pengkodean fonetis dari string query masukan "fiihaa" adalah fh.

3.3 Pencocokan Kode Fonetis

Proses yang terjadi dalam pencocokan kode fonetis ini adalah mencocokkan kode fonetis antara string query masukan dengan kode fonetis dataset Al-Qur'an. Dimana kode yang memiliki kecocokan akan dikeluarkan sebagai sebuah output pada sistem.



Gambar 3. Pencocokan Kode Fonetis

3.4 Perhitungan Similarity

1. Menghitung nilai ED (Edit Distance) antara string fiihaa dan [2:25] fiihaa.

Tabel 2. Nilai Edit Distance

	String 1	f	i	i	h	a	a
String 2	0	1	2	3	4	5	6
f	1	0	1	2	3	4	5
i	2	1	0	1	2	3	4
i	3	2	1	0	1	2	3
h	4	3	2	1	0	1	2
i	5	4	3	2	1	1	2

2. Menghitung nilai PED (Phonetic Edit Distance) antara string fh dan [2:2] fh.

Tabel 3. Nilai Phonetic Edit Distance

	String 1	f	h
String 2	0	1	2
f	1	0	1
h	2	1	0

3. Menghitung nilai Edscr (Edit Distance Score).

$$ED_{scr} = \frac{6 - 2}{6} = 0.66666667(8)$$

4. Menghitung nilai PEDscr (Phonetic Edit Distance Score).

$$PED_{scr} = \frac{2 - 0}{2} = 1(9)$$

5. Menghitung nilai FOM (Figure of Merit).

$$FOM = \frac{1 + \frac{0.666666667}{10}}{1,1} = 0.96969697(10)$$

Jika diubah dalam bentuk persen, maka hasil similaritynya menjadi 96.97

3.5 Pemeringkatan Dokumen

Tahapan dilakukannya pengurutan dari hasil pencocokan kode fonetis tersebut. Pengurutan dilakukan berdasarkan besar nilai similarity dari nilai yang paling besar hingga nilai yang paling kecil.

3.6 Hasil Pencarian

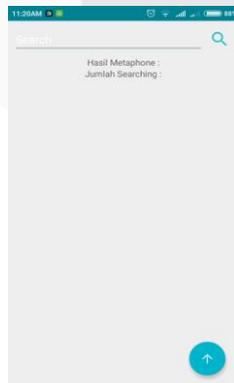
Keluaran dari sistem yang berupa ayat-ayat Al-Qur'an dimana berisikan informasi mengenai nama surat, nomor surat, nomor ayat, teks ayat Al-Qur'an aksara Arab, teks ayat Al-Qur'an aksara Latin dan hasil metaphone yang memiliki kemiripan kode fonetis antara string query masukkan dengan string outputan sistem.

3.6.1 Tampilan Awal



Gambar 4. Tampilan Awal

3.6.2 Tampilan Pencarian Ayat Al-Qur'an



Gambar 5. Tampilan Pencarian Ayat Al-Qur'an

3.6.3 Tampilan Hasil Pencarian Ayat Al-Qur'an Query fiihaa



Gambar 6. Tampilan Hasil Pencarian Ayat Al-Qur'an Query fiihaa

4. Evaluasi

4.1 Hasil Pengujian

Tabel 4. Hasil Pengujian

No	Sistem	MAP	Recall	Korelasi
1	Lafzi	87.71%	95.61%	-
2	IslamiCity	71.94%	87.37%	-
3	TA Eka menggunakan Algoritma Metaphone dan Levenshtein Distance	76.34%	81.99%	89.47%
4	TA Mining menggunakan Algoritma Double Metaphone dan Jaro Winkler	84.36%	92.81%	89.44%
5	TA Ulfa menggunakan Algoritma Soundex dan Cosine Similarity	81.81%	80.81%	86.15%
6	TA Oca menggunakan Algoritma Soundex dan Damerau Levenshtein Distance	77.89%	90.72%	82.36%

Tabel 5. Hasil Pengujian Sistem Berdasarkan Panjang Kode Fonetis

No	Panjang Kode Fonetis	MAP	Recall
1	2	75.34%	89.26%
2	3	100%	79.80%
3	4	60%	60%
4	5	100%	100%
5	6	61.34%	72.22%
6	8	67.77%	90%
7	9	53.61%	80%
8	10	100%	100%

4.2 Analisis Hasil Pengujian

4.2.1 Hasil Pengujian dan Analisis Skenario Pertama

Pada skenario pertama ini dilakukan pengujian untuk mengetahui nilai AP (Average Precision) dan MAP (Mean Average Precision) dari sistem berdasarkan variasi dari query masukan pengguna. Adapun tahapan-tahapan yang dilakukan untuk mendapatkan hasil AP (Average Precision) dan MAP (Mean Average Precision) adalah sebagai berikut:

1. Input query dengan 55 macam variasi penulisan seperti yang tertera pada lampiran 2 berdasarkan dari hasil kuesioner.

2. Sistem akan mengurutkan hasil yang dicari berdasarkan kemiripan dengan query secara descending, dimana hasil yang menempati posisi paling atas merupakan hasil yang memiliki nilai similarity tertinggi atau dengan kata lain memiliki nilai kemiripan paling baik.
3. Didapatkan kata yang muncul dari hasil keluaran sistem sesuai dengan inputan querynya.
4. Perhitungan AP (Average Precision) didapatkan berdasarkan presisi di top i dokumen dibagi dengan jumlah dokumen yang relevan, dan hasilnya dibagi dengan jumlah dokumen yang relevan.
5. Perhitungan MAP (Mean Average Precision) didapatkan berdasarkan hasil AP pada setiap query dibagi dengan jumlah query pada setiap query.
6. Didapatkan nilai MAP (Mean Average Precision) seperti pada tabel 4.

Dari skenario pengujian pertama didapatkan hasil MAP (Mean Average Precision) dari setiap variasi querynya. Dan MAP yang didapatkan dari setiap variasi querynya menghasilkan sebuah nilai MAP dari keseluruhan variasi query yaitu 76.34%. Hal ini menunjukkan bahwa sistem dapat menangani bentuk query yang diinputkan user secara bervariasi.

Dari keseluruhan variasi query yang telah diuji, query khairun memiliki MAP terkecil sebesar 50%. Hal ini disebabkan karena pada variasi penulisan dari query khairun yaitu hoyrun dan khayrun menghasilkan kode fonetis yang berbeda dengan kata khairun. Dimana kata hoyrun dan khayrun menghasilkan kode fonetis hyrn, sedangkan kata khairun menghasilkan kode fonetis hrn. Sehingga variasi penulisan hoyrun dan khayrun yang diinputkan pada sistem tidak mengeluarkan hasil yang dicari. Dan itu yang menyebabkan kecilnya MAP pada query khairun.

Begitu juga pada pengujian query yang lebih dari satu kata yaitu query khairun lakum memiliki MAP terkecil sebesar 12.04%. Hal ini disebabkan adanya perbedaan kode fonetis antara query masukkan pengguna dengan dataset. Karena pada kata khairun lakum menerapkan hukum tajwid idgham, dimana jika terdapat huruf n bertemu dengan huruf l. Maka huruf n akan dihapus atau diabaikan.

Selain itu telah dilakukan juga pengujian sistem berdasarkan panjang kode fonetisnya seperti pada tabel 5. Dimana panjang kode fonetis 3, 5 dan 10 memiliki nilai MAP terbesar yaitu 100%. Sedangkan pada panjang kode fonetis 9 memiliki nilai MAP terkecil yaitu 53.61%.

4.2.2 Hasil Pengujian dan Analisis Skenario Kedua

Pada skenario kedua ini dilakukan pengujian untuk mengetahui nilai recall dari sistem berdasarkan variasi dari query masukan pengguna. Adapun tahapan yang dilakukan adalah sebagai berikut:

1. Penulis menentukan query inputan yang ingin diuji. Dan didapatkan 55 macam variasi penulisan seperti yang tertera pada lampiran 3 berdasarkan hasil kuesioner yang harus dilakukan pengujian.
2. Penulis melakukan pencarian secara manual pada Al-Qur'an terhadap query yang diuji.
3. Masukkan query pada sistem.
4. Dilakukan perbandingan antara hasil output sistem dengan hasil pencarian manual. Dan koreksi surat mana saja yang tidak dioutputkan sistem.
5. Perhitungan recall didapatkan dengan cara jumlah output yang dihasilkan sistem dibagi dengan total jumlah yang benar menurut manual.
6. Didapatkan nilai recall seperti pada tabel 4.

Dari hasil pengujian ini dapat disimpulkan bahwa sistem cukup memberikan hasil yang diinginkan oleh pengguna. Hasil nilai recall dari pengujian ini adalah 81.99%. Beberapa query tidak bisa mendapatkan hasil recall yang baik, dikarenakan adanya hasil yang tidak dioutputkan pada sistem. Seperti query khairun, dari hasil output sistem terdapat 2 hasil yang kode fonetisnya memiliki kemiripan dengan query masukkan. Sedangkan pada pencarian manual terdapat 22 hasil. Darisini dapat kita lihat bahwa terdapat 20 hasil yang tidak dioutputkan oleh sistem. Semua yang tidak dioutputkan oleh sistem dikarenakan adanya penerapan hukum tajwid idgham. Dimana jika terdapat huruf latin n bertemu dengan salah satu huruf idgham yaitu y, n, m, w, l, r. Maka huruf 'n' dari kata tersebut dihapus. Semisalnya pengguna menginputkan query khairun, kode fonetis dari query tersebut adalah hrn. Sedangkan pada dataset terdapat kata khairun lakum, dan terdapat huruf n bertemu dengan huruf l. Maka huruf 'n'nya dihapus, sehingga kode fonetis dari query tersebut menjadi hr lkm. Karena dari kode fonetis antara query masukkan dan dataset berbeda, maka sistem tidak mengeluarkan hasil yang ingin dicari. Itulah yang menyebabkan nilai recall rendah.

Selain itu telah dilakukan juga pengujian sistem berdasarkan panjang kode fonetisnya seperti pada tabel 5. Dimana panjang kode fonetis 5 dan 10 memiliki nilai recall terbesar yaitu 100%. Sedangkan pada panjang kode fonetis 4 memiliki nilai MAP terkecil yaitu 60%.

4.2.3 Hasil Pengujian dan Analisis Skenario Ketiga

Pada skenario ketiga ini dilakukan pengujian untuk mengetahui nilai korelasi. Dimana penulis mengurutkan setiap variasi query pada query yang telah ditentukan secara manual. Daftar urutan tersebut berupa angka 1-5, angka 1 menunjukkan query tersebut berada pada posisi paling atas. Angka 2,3,4 merupakan urutan selanjutnya, dan angka 5 menunjukkan query tersebut berada pada posisi paling bawah. Dari hasil perbandingan urutan yang didapat antara pengurutan manual yang dilakukan oleh penulis dengan pengurutan dari output sistem maka akan didapatkan hasil nilai korelasi. Hasil nilai korelasi pada pengujian ini adalah 89.47%. Dimana nilai korelasi yang bernilai 1 merupakan nilai yang baik dan yang bernilai -1 merupakan nilai yang kurang baik. Faktor yang menyebabkan pengurutan yang berbeda antara pengurutan manual yang dilakukan oleh penulis dengan pengurutan dari output sistem adalah hasil similaritynya. Pada query wain tawallaw fa'innama menghasilkan similarity yang lebih rendah dikarenakan query wain tawallaw fa'innama memiliki panjang karakter yang lebih sedikit dibandingkan dengan query waintawallaw fa'innama.

Selain itu pada hasil pengujian sistem berdasarkan panjang kode fonetis seperti pada tabel 5, nilai korelasi tidak dimasukkan. Karena setiap query memiliki variasi query. Dan setiap variasi query pada query tersebut terdapat panjang kode fonetis yang berbeda. Sedangkan nilai pada korelasi yang ada pada lampiran 4 tidak dihitung satu persatu variasi querynya seperti AP dan recall, namun langsung keseluruhan dari variasi query tersebut. Sehingga untuk nilai korelasi tidak bisa dihitung rata-ratanya pada hasil pengujian sistem berdasarkan panjang kode fonetisnya.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Berdasarkan hasil pengujian yang diperoleh dan analisis yang telah dilakukan maka dapat ditarik kesimpulan sebagai berikut:

1. Pembangunan sistem pencarian ayat Al-Qur'an berdasarkan kemiripan pengucapan untuk Al-Qur'an transliterasi latin yang disesuaikan dengan pelafalan bahasa Indonesia dapat dilakukan dengan menggunakan algoritma metaphone.
2. Berdasarkan dari hasil pengujian yang telah dilakukan, kinerja sistem pencarian ayat Al-Qur'an berdasarkan kemiripan pengucapan dengan menggunakan algoritma metaphone mendapatkan nilai MAP (Mean Average Precision) 76.34%, nilai recall 81.99%.
3. Algoritma Levenshtein Distance cukup baik digunakan dalam penghitungan tingkat similarity antara dua buah string, khususnya pada penelitian ini. Terlihat dari nilai korelasi sebesar 89.47%, dimana nilai 100% bernilai sangat baik dan -100% bernilai sangat buruk.

5.2 Saran

Adapun saran yang didapatkan dari penelitian tugas akhir ini sebagai pertimbangan untuk penelitian selanjutnya adalah sebagai berikut:

1. Menambah jumlah dataset yang digunakan untuk meningkatkan performansi dari sistem.
2. Mengembangkan sistem dengan menggunakan algoritma fonetis maupun similarity yang berbeda. Seperti N-Grams, Caverphone dan lain-lain.
3. Mengembangkan sistem dengan menggunakan aksara arab.
4. Mengembangkan sistem yang dapat menangani query lebih dari satu kata yang antar katanya tersambung.

Daftar Pustaka

- [1]N. Baidan. *Perkembangan tafsir al-Qur'an di Indonesia*. Tiga Serangkai, 2003.
- [2]A. Binstock and J. Rex. *Practical algorithms for programmers*. Addison-Wesley Longman Publishing Co., Inc., 1995.
- [3]V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710, 1966.
- [4]S. H. Peggy. Optimasi pencarian kata pada aplikasi penerjemah bahasa mandarin–indonesia berbasis android dengan algoritma levenshtein distance.
- [5]A. Pirkola, J. Toivonen, H. Keskustalo, K. Visala, and K. Järvelin. Fuzzy translation of cross-lingual spelling variants. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 345–352. ACM, 2003.
- [6]B. Pratama and S. Pamungkas. Analisis kinerja algoritma levenshtein distance dalam mendeteksi kemiripan dokumen teks. *LOGIK@*, 6(2):131–143, 2016.
- [7]M. A. d. M. P. RI. *Pedoman Alih Aksara Arab ke Latin*. RI, Menteri Agama dan Menteri P&K, 1987.
- [8]N. Safaat. *Android: Pemrograman aplikasi mobile smartphone dan tablet pc berbasis android*, revisi kedua. 2015.
- [9]M. Syaroni and R. Munir. Pencocokan string berdasarkan kemiripan ucapan (phonetic string matching) dalam bahasa inggris. In *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*, 2005.
- [10]A. Ta'a, Q. Abed, and M. Ahmad. Al-quran ontology based on knowledge themes. *Journal of Fundamental and Applied Sciences*, 9(5S):800–817, 2017.