

## Pengembangan Implementasi Pengujian Pencarian Ayat Al-Qur'an Menggunakan Algoritma *Doublemetaphone* Berdasarkan Kemiripan Ucapan

*Development Implementation Testing in Search Al-Qur'an Verse by Using Doublemetaphone Algorithm Based on Similiarty Speech*

Harini Amalia Maturah<sup>1</sup>, Arif Bijaksana<sup>2</sup>, Eko Darwiyanto<sup>3</sup>

<sup>1,2,3</sup>Prodi S1 Teknik Informatika, Fakultas Informatika, Universitas Telkom

<sup>1</sup>hariniamalia@gmail.com, <sup>2</sup>arifbijaksana@gmail.com, <sup>3</sup>ekodarwiyanto@telkomuniversity.ac.id

### Abstrak

Al-Qur'an merupakan pedoman bagi umat muslim di seluruh dunia. Setiap hari kita membaca Al-Quran dan tidak jarang seseorang melakukan pencarian kata yang terdapat didalam A-Qur'an. Didalam Al-Qur'an terdapat 114 surah yang didalamnya terdiri dari 30 juz dan 6236 ayat. Sehingga tidak mudah bagi seseorang yang ingin melakukan pencarian kata didalam Al-Qur'an. Di zaman yang modern ini banyak orang yang memanfaatkan teknologi untuk mendapatkan informasi yang terkandung di Al-Qur'an. Terdapat banyak perangkat lunak yang dapat digunakan untuk pencarian kata di Al-Qur'an. Hanya saja perangkat lunak tersebut menggunakan aksara dan Bahasa Arab. Hal ini membuat kita sebagai orang awam yang kurang dan belum fasih dalam berbahasa Arab akan kesulitan ketika menggunakan perangkat lunak tersebut. Perangkat lunak yang sudah ada, hanya menggunakan *Exact String Matching* dalam melakukan pencariannya. Apabila seseorang melakukan kesalahan dalam penulisan maka sistem tersebut tidak akan menampilkan hasil yang diinginkan. Kesalahan dalam penulisan bisa saja terjadi karena adanya perbedaan pelafalan orang Indonesia dengan orang Arab. Metode yang digunakan dalam penelitian ini adalah algoritma *doublemetaphone*, algoritma ini akan melakukan pencocokan terhadap kemiripan ucapan. Pencocokan yang dilakukan yaitu melakukan pencarian *query* didasarkan dari kode fonetis, kode fonetis ini yang akan menerjemahkan informasi berdasarkan kemiripan ucapan. Hasil dari penelitian tugas akhir ini didapatkan nilai presisi 0.77 atau 77%, nilai korelasi sebesar 0.93 atau 93% dan nilai *Recall* sebesar 0.83 atau 83%. Telah dilakuakn pula pengembangan terhadap aplikasi berupa tampilan gambar dan suara dari potongan ayat yang memiliki kecocokan dengan *query* yang diinputkan oleh *user*.

**Kata Kunci :** Al-Quran, Phonetic String Matching, Kode Fonetis Algoritma *Doublemetaphone*

### Abstract

*The Qur'an is a guide for Muslims in Worldwide. Everyday we read the Qur'an and not infrequently people searching word in The Qur'an. The Qur'an contains 114 letters, 30 juz', and 6236 verses. So it is not easy for people who wants searching words in The Qur'an. In this modern era many people using technology to obtain information that contained in The Qur'an. There is many software that can be use to searching word in The Qur'an. But the software only using an alphabet and Arabic. This reason make us as people who less and not yet fluent in Arabic will have difficult time when using such software. Software that already available, just use Exact String Matching for their searching method. If someone made a mistake in writing, then the system willnot show the result that we want. Errors in writing can happen because of differences in the pronunciation between Indonesian and Arabian. The method that I used in this research is doublemetaphone algorithm, this algorithm will perform matching based on similarity speech. Matching that I used is perform a search query based on the phonetic code, phonetic code will translate the information based on similarity speech. Result from this research obtained a precision value of 0.77 or 77%, the correlation value of 0.93 or 93% and Recall value of 0.83 or 83%. The development for this application are this application will show the image and the sound that has a match or similar with the query that who input by the user.*

**Keyword :** Al-Quran, Phonetic String Matching, Fonetis Code, *Doublemetaphone* Algorithm

### 1. Pendahuluan

Al-Qur'an merupakan pedoman hidup bagi seluruh umat muslim yang ada di dunia. Indonesia merupakan salah satu negara yang mayoritas masyarakatnya beragama muslim. Al-Qur'an memiliki 114 surat yang di dalamnya terdapat 6236 ayat dan 77.845 kata. Jika kita lihat dari banyaknya jumlah kata yang terkandung didalam Al-Qur'an maka cukup sulit bagi kita jika ingin melakukan pencarian kata didalam Al-Qur'an. Namun di zaman yang modern ini dimana orang-orang sudah banyak menggunakan kecanggihan teknologi untuk mendapatkan informasi yang terkandung didalam Al-Qur'an, teknologi yang tersedia dapat berupa aplikasi yang dapat kita dapat kita unduh kedalam *smartphone* ataupun berupa *website* yang dapat dengan mudah kita kunjungi setiap saat.

Terdapat aplikasi yang sudah tersedia untuk memudahkan dalam melakukan pencarian tersebut. Aplikasi tersebut biasanya mengharuskan penggunaanya untuk melakukan proses pencarian dengan memasukkan kata kunci dalam bahasa dan Aksara Arab. Hal tersebut dapat menyulitkan *user* sebagai orang Indonesia yang tidak terlalu mengerti penulisan dalam bahasa Arab. Terlebih lagi jika ingin mengetikan dengan bahasa Arab diperlukan

*keyboard* atau perangkat lunak tambahan. Adapun masalah lainnya jika kita sudah bisa menginputkan kata tetapi melakukan kesalahan dalam pengetikannya, maka aplikasi tersebut tidak akan menampilkan hasil dari *keyword* yang diinputkan tersebut.

Dalam penelitian sebelumnya mengenai "Analisa dan Implementasi Pencarian Ayat Al-Quran Menggunakan Algoritma *Doublemetaphone* Berdasarkan Kemiripan Ucapan" telah dibuat sebuah aplikasi untuk mengatasi masalah tersebut [1]. Hanya saja terdapat kekurangan dalam pengimplementasian perangkat lunak. Dalam penelitian yang akan penulis kerjakan ini akan dilakukan beberapa pengembangan dari kekurangan yang ada dari penelitian sebelumnya. Kekurangan tersebut diambil berdasarkan saran yang diberikan dari penelitian sebelumnya. Saran yang diberikan oleh penelitian sebelumnya yaitu diperlukan sebuah tahap normalisasi dari pelafalan Al-Quran dan diperlukan perbaikan aplikasi didalam pengurutan nomor ayat pada Al-Qur'an berdasarkan nilai kemiripannya. Selain itu didalam tugas akhir ini saya bertujuan untuk melakukan pengembangan fitur-fitur yang belum tersedia dan fitur tersebut dapat berguna sebagai pembelajaran mengenai pengucapan lafal Al-Qur'an yang baik dan benar menurut tajwid. Algoritma yang digunakan dalam penelitian ini adalah algoritma *doublemetaphone* seperti algoritma yang digunakan dalam penelitian sebelumnya.

Pemilihan algoritma ini dikarenakan algoritma ini sudah terbukti dapat digunakan untuk melakukan pencarian dalam ayat Al-Qur'an. Dalam penelitian ini penulis ingin memberikan hasil dari pencocokan dalam pencarian menjadi akurat dan mengembangkan aplikasi yang mudah digunakan untuk *user*.

## 2. Dasar Teori

### 2.1 Al-Quran

Al-Quran adalah kitab suci yang Allah SWT wahyukan kepada Nabi Muhammad SAW melalui perantara Malaikat Jibril. Umat Islam percaya bahwa Al-Qur'an merupakan puncak dan penutup wahyu dari Allah SWT untuk seluruh umat manusia di dunia. Al-Qur'an merupakan kata dari bahasa Arab yang memiliki arti "bacaan" atau "sesuatu yang dibaca berulang-ulang". Al-Qur'an terdiri dari 114 surah, 30 juz dan 6236 ayat. Al-Qur'an terbagi menjadi 30 bagian yang dikenal dengan nama juz. Pembagian juz ini memudahkan mereka yang ingin menuntaskan pembacaan Al-Qur'an dalam kurun waktu 30 hari. Surah didalam Al-Qur'an memiliki sejumlah ayat, dimulai dari surah yang terdiri dari 3 ayat yaitu Al-Kautsar, An-Nasr dan Al-Asr hingga surah yang mencapai 286 ayat yaitu Al-Baqarah. [2]

### 2.2 Pedoman Alih Aksara Arab ke Latin

Panduan alih aksara dari 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/u/1987. Pedoman ini disusun untuk menunjukkan perbedaan agar perbedaan tersebut dapat dipahami. Banyaknya variasi dalam penulisan kata dari bahasa Arab, hendaknya kita mengutamakan kata yang populer dalam penggunaannya. [2]. Dalam melakukan alih aksara, beberapa penulis menggunakan sumber dari bahasa Inggris beserta alih aksarannya. Alih aksara yang biasa digunakan merupakan alih aksara Qalam. Hal tersebut terkadang membuat kesalahpahaman dan kekeliruan dengan alih aksara Indonesia dalam pembacaannya. Dibawah ini terdapat tabel yang menjelaskan alih aksara Qalam dengan alih aksara Indonesia. [3]

### 2.3 String

Dalam bahasa pemrograman *String* adalah sebuah deretan simbol. *String* menurut *Dictionary of Algorithms and Data Structures* NIST adalah susunan dari karakter-karakter (angka, alphabet atau karakter yang lain) dan direpresentasikan sebagai struktur dan array. *String* biasanya disebut sebagai sebuah bahasa yang abstrak yang menyimpan sebuah sekuens nilai data (berupa *bytes*) yang merupakan pembentuk dari karakter sesuai dengan *encoding* karakter yang disepakati seperti, ASCII. *String* dapat berupa jata, frase atau kalimat [4].

### 2.4 Pencocokan String (String Matching)

Pencocokan *String* (*String Matching*) adalah sebuah permasalahan untuk menemukan sebuah pola susunan karakter *string* didalam *string* lain atau bagian dari sebuah teks. Saat melakukan pencarian *string* dengan menggunakan teknik pencocokan *string*, teknik yang biasa digunakan adalah *Exact String Matching* yaitu teknik yang menghasilkan keluaran *string* yang sama persis dengan *keyword* yang dimasukkan, baik itu dari jumlah maupun urutan dari karakternya. Pencocokan *string* (*string matching*) 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 [5].

### 2.5 Klasifikasi Pencocokan String

Secara garis besar pencocokan *string* (*string matching*) dibedakan menjadi dua bagian, yaitu [7]:

1. *Exact string matching* adalah pencocokan *string* secara tepat. Dilihat dari susunan karakter dalam *string* yang dicocokkan memiliki jumlah maupun urutan karakter yang sama. Contoh: kata *step* akan cocok hanya dengan kata *step*.
2. *Inexact string matching* atau *Fuzzy string* adalah pencocokan *string* secara samar, maksudnya pencocokan *string* dimana *string* yang dicocokkan memiliki kemiripan disusunan karakter yang berbeda

(jumlah atau urutan) tetapi *string* memiliki kemiripan dari penulisan atau kemiripan ucapan (*phonetic string matching*).

Penggunaan metode *inexact string matching* bermanfaat terhadap pencocokan *string* yang memiliki tingkat toleransi terhadap susunan karakter. Toleransi terhadap penulisan tersebut dapat diatasi dengan penggunaan pencocokan *string* berdasarkan kemiripan ucapan (*phonetic string matching*).

## 2.6 Pengodean Fonetis

Pengodean fonetis untuk huruf *Arab* pada teks Al-Qura'an menggunakan *doublemetaphone* yang merupakan bagian dari *inexact string matching* yang berdasarkan pada pencocokan dilihat dari kemiripan pengucapan (*phonetic string matching*). Dalam algoritma ini pelafalan akan diubah kedalam bentuk sebuah kode, yaitu kode fonetis yang memiliki pengaturan dalam pengkodeannya. Tujuannya adalah agar sistem dapat melakukan penanganan terhadap aksara *Arab* dan penulisan lafal dalam aksara Latin yang memiliki kemiripan dalam pengucapannya. Tujuan lain dari adanya pengkodean fonetis ini untuk mengatasi pelafalan bacaan ayat yang memiliki aturan tajwid dalam pembacaannya. Pengkodean ini dilakuakn terhadap teks Al-Qur'an transliterasi Latin dan *query* masukkan dari *user*.

## 2.7 Algoritma Metaphone

Algoritma metaphone dipublikasikan oleh Lawrence Philips dalam sebuah artikel berjudul "Hanging on the Metaphone" dalam jurnal "Computer Language" vol. 7 n. 12, Desember 1990, pp. 39-43. Algoritma metaphone menghasilkan kode fonetis yang panjang karakternya berbeda sesuai dengan panjang *string* masukan [7]. Langkah-langkah dalam algoritma dapat dilihat pada buku "Practical Algorithms for Programmers" [7]. 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. Suara 'sh' direpresentasikan dengan 'X' dan nol ('0') merepresentasikan suara 'th'. Pada algoritma metaphone, pemberian kode fonetis memperhatikan juga interaksi antara konsonan dan vokal dalam kata serta kelompok konsonan bukan hanya sebuah konsonan seperti pada algoritma soundex. Walaupun pada algoritma soundex sudah diberikan aturan baru mengenai perlakuan terhadap kelompok konsonan sebelum langkah pemberian kode fonetis, tetapi masih sangat terbatas.

## 2.8 Algoritma Doublemetaphone

Doublemetaphone adalah salah satu algoritma dari *inexact string matching* dimana pencocokan dilihat dari kemiripan *string* berdasarkan pengucapannya (*phonetic string matching*). Algoritma *doublemetaphone* akan menghasilkan dua buah kode fonetis, yaitu kode fonetis primer dan sekunder [8]. Kedua kode fonetis ini memiliki tujuan untuk mengatasi masalah ambiguitas dan keanekaragaman dalam variasi pelafalan *string* yang tidak diatasi di algoritma metaphone. Dengan adanya dua pengkodean fonetis maka sebuah *string* dapat memiliki dua kemungkinan kemiripan, yaitu memungkinkan kemiripan terdapat pada kode primer atau kode sekundernya. Didalam algoritma *doublemetaphone*, kode fonetis yang dihasilkan memiliki panjang maksimal empat digit karakter. Hal ini mengartikan bahwa apabila terdapat sebuah *string* yang memiliki lebih dari empat digit kode fonetis maka empat kode awal yang dihasilkan terlebih dahulu yang akan diambil, dan jika *string* yang ada hanya menghasilkan kurang dari empat digit maka kode akan ditampilkan seadanya, tanpa adanya penambahan kode lainnya. Kode fonetis berupa karakter huruf, dimana pengeksesian dari kode fonetisnya dilakukan hingga karakter huruf terakhir pada *string* dieksekusi. Dibawah ini merupakan langkah-langkah secara umum dalam mendapatkan kode fonetis pada sebuah *string* yang menggunakan algoritma *doublemetaphone* [8]:

- Sebuah *string* dimasukan untuk ditemukan kode fonetisnya.
- Dilakukan pengecekan pada tiap karakter *string*, jika *string* memiliki karakter huruf A, I, U, E, O dan Y maka diabaikan, kecuali karakter huruf tersebut muncul diawal pengecekan karakter maka keluarkan kode fonetis yang sesuai dengan karakter huruf tersebut.
- Pemberian kode fonetis pada tiap karakter huruf yang dilakukan pengecekan berdasarkan tabel kode fonetis *doublemetaphone*.
- Jika iterasi pengecekan huruf ke-i sama dengan huruf ke-i+1 maka eksekusi pengecekan selanjutnya adalah huruf ke-i +2, namun jika ke-i tidak sama dengan huruf ke-i+1 maka eksekusi yang dilakukan adalah huruf ke-i+1, dan seterusnya.
- Hasil keluaran kode fonetis berupa 4 digit pertama dari kode fonetis yang didapatkan, jika hasilnya kurang maka keluarkan kode fonetis.

## 2.9 Pencocokan Kode Fonetis

Pencocokan dilakukan berdasarkan dari hasil kesamaan karakter pada kode fonetis *query* dengan kode fonetis dataset. Hasil kode fonetis dari *query* masukan dibandingkan dengan kode fonetis pada dataset teks Al-Quran transliterasi Latin. Kemudian dilakukan pencarian kode pada dataset, dataset yang memiliki kode yang sama dengan *query* akan dikeluarkan sebagai hasil *output* system. Kode fonetis *query* akan dianggap cocok apabila

memenuhi seluruh karakter kode pada kode fonetis dataset/korpus teks Al-Quran. Namun, apabila kode fonetis pada *query* maka tidak terdapat kecocokan pada *query* tersebut.

## 2.10 Metrik Evaluasi

Pengukuran kinerja sistem dapat dilakukan dengan mengetahui nilai presisi, *Recall* dan korelasi. Apabila nilai dari ketiga penilaian ini menunjukkan hasil yang tinggi, menandakan bahwa sistem memiliki kemampuan yang baik. Selain itu, ketiga perhitungan nilai ini umumnya digunakan pada pengujian yang berkaitan dengan *information retrieval*. Oleh karena itu, pentingnya dilakukan pengujian sistem terhadap ketiga nilai tersebut.

- Presisi, merupakan tingkat ketepatan antara informasi yang dimiliki oleh pengguna dengan hasil jawaban dari sistem. Maka dapat dikatakan, bahwa nilai presisi didapatkan berdasarkan probabilitas jumlah dokumen yang relevan dari semua dokumen yang di-*retrieve* dibandingkan dengan keseluruhan jumlah dokumen yang di-*retrieve* **Invalid source specified..** Berdasarkan pada tabel 2.11, maka perhitungan untuk presisi di notasikan pada persamaan dibawah ini.

$$\text{Presisi} = \frac{|TP|}{|TP| + |FP|}$$

Persamaan 1

- *Recall*, merupakan tingkat keberhasilan sistem yang dapat menemukan kembali sebuah informasi. Maka dapat dikatakan, bahwa nilai *Recall* didapatkan berdasarkan probabilitas dokumen relevan yang di-*retrieve* dibandingkan dengan jumlah-jumlah dokumen yang relevan **Invalid source specified..** Berdasarkan pada tabel 2.11, maka perhitungan *Recall* di notasikan pada persamaan dibawah ini.

$$\text{Recall} = \frac{|TP|}{|TP| + |FN|}$$

Persamaan 2

- Korelasi, merupakan analisa dalam mengetahui tingkat hubungan antara dua buah variabel yang bersifat kuantitatif. Hubungan antara dua variabel dinyatakan positif, bila suatu variabel ditingkatkan nilainya maka nilai variabel yang lain akan juga meningkat. Sedangkan, hubungan dua variabel dinyatakan negatif bila nilai suatu nilai variabel ditingkatkan, maka akan menurunkan nilai variabel lainnya. Kuatnya hubungan antara variabel dinyatakan dalam koefisien relasi. Koefisien relasi positif terbesar bernilai 1, dan koefisieif negatif terbesar bernilai -1. Makin kecil nilai koefisien relasi, maka akan semakin besar error untuk membuat suatu prediksi. Perhitungan untuk korelasi di notasikan pada persamaan dibawah ini.

$$r_{xy} = \frac{\sum xy}{\sqrt{(\sum x^2 y^2)}}$$

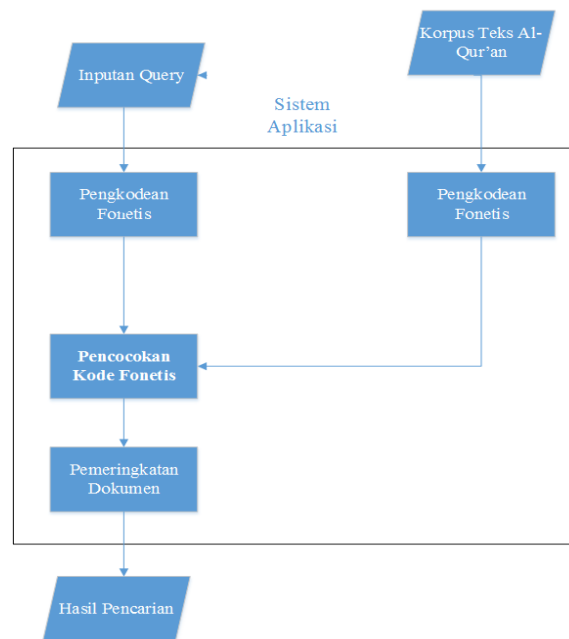
r = Nilai korelasi, x = variabel data 1, y = variabel data 2

Persamaan 3

## 3. Perancangan Sistem

### 3.1. Gambaran Umum Sistem

Dalam penelitian ini akan dibangun sebuah sistem yang dapat memnuhi tujuan dari penelitian tugas akhir ini. Sistem pencarian merupakan sistem pencarian teks ayat di Al-Qur'an yang sudah ditranselasi dari bahasa Arab ke bahasa Latin yang pencariannya dilakukan berdasarkan kemiripan dari kode fonetis. Sistem akan dibangun berdasarkan algoritma *doublemetaphone*, penggunaan algoritma ini berguna untuk melakukan konversi dari sebuah masukkan *string* menjadi *string* kode fonetis. Pengkodean fonetis dilakukan pada dabase korpus Al-Quran dan terhadap *query* masukkan sistem. Lalu akan dilakukan pencocokan *string* (*string matching*) antara *query* masukkan dengan database korpus Al-Qur'an, pencocokan dilihat dari kemiripan kode fonetisnya. Akan dilakukan pemeringkatan terhadap kode fonetis berdasarkan bobot dari kemiripan kata tersebut. Hasil yang diharapkan dari pencarian kata dari teks Al-Qur'an ini berupa teks ayat Al-Qur'an yang berisikan informasi mengenai nama surah dan nomor ayat. Selain itu juga, terdapat sebuah gambar dari hasil surah yang dianggap cocok dengan *query* yang dimasukkan. Tujuan dari penelitian ini juga adalah memberikan pembelajaran cara pembacaan ayat Al-Qur'an yang baik dan benar maka akan diberikan juga hasil berupa suara dari ayat surat yang dimaksudkan. Agar *user* yang kurang bisa membaca ayat Al-Qur'an sesuai dengan hukum bacaan tajwid dapat mengucapkannya dengan benar. Gambaran sistem akan dijelaskan pada gambar dibawah ini.



Gambar 1 Gambaran Umum Sistem

### 3.2. Alur Proses Sistem

Tugas akhir ini menggunakan teks Al-Quran dengan transliterasi Latin yang disesuaikan dengan pelafalan orang Indonesia sebagai dataset/korpusnya. Pemilihan Al-Quran dengan transliterasi Latin bertujuan agar pengguna di Indonesia dapat dengan mudah mempelajarinya. Pengubahan teks Al-Quran dari aksara bahasa arab menjadi aksara Latin dilakukan dalam berbagai tahap.

- *Input query*, merupakan tahapan dimana *user* memasukan *string* berdasarkan dengan *user* yang selanjutnya dituliskan oleh *user* menjadi *query input*.
- Pengkodean fonetis, merupakan tahap *query* yang telah dimasukkan akan diubah menjadi sebuah kode fonetis berdasarkan pada aturan algoritma *doublemetaphone*.
- Pencocokan kode fonetis, merupakan tahap pencocokan kode fonetis dari *query input* dengan kode fonetis pada database korpus Al-Qur'an.
- Tahapan selanjutnya dilakukan pengecekan terhadap proses pencocokan fonetis. Jika kode fonetis ditemukan maka hasil dari pencocokan tersebut akan diproses pada tahap selanjutnya. Tetapi, jika kode fonetis tidak ditemukan maka *user* akan diminta kembali untuk memasukkan *query* pada tampilan awal aplikasi atau *user* dapat memilih untuk tidak melakukan pencarian dan keluar dari aplikasi.
- Pemeringkatan dokumen, merupakan tahap dilakukannya pengurutan dari hasil pencocokan kode fonetis, pengurutan dilakukan berdasarkan dari tingkat dengan bobot kemiripan paling mirip kode fonetisnya. Jika saat pengurutan terdapat beberapa bobot yang memiliki kemiripan kode fonetis yang sama, maka urutan akan ditentukan dengan urutan surah dan ayat di Al-Qur'an.
- Hasil *output* yang diharapkan dari sistem adalah berupa ayat-ayat Al-Qur'an yang memiliki kemiripan kode fonetis dengan *query input*. *Output* yang dihasilkan berupa teks ayat Al-Qur'an, nama surah, nomor ayat, kode fonetis, gambar dari ayat yang dimaksud dan suara ayat Al-Qur'an yang cara pembacaannya baik dan benar sesuai hukum tajwid.

### 3.3. Data/Korpus Al-Qur'an

Dalam melakukan penelitian tugas akhir ini, digunakan teks Al-Qur'an dengan transliterasi Latin yang melafalannya disesuaikan dengan orang Indonesia. Pemilihan Al-Qur'an dengan transliterasi Latin bertujuan agar pengguna di Indonesia dapat dengan mudah mempelajarinya. Terdapat beberapa tahap dalam pengubahan teks Al-Qur'an dari aksara bahasa Arab menjadi aksara Latin, yaitu

1. Tahap pertama yaitu mempelajari berbagai variasi penulisan teks Al-Quran yang didasarkan pada hukum bacaannya atau *tajwid*, baik itu hukum *ikhfa*, *idgham*, dan *iqlab*.
2. Pada tahap kedua dilakukan riset kecil kepada responden beragama Islam untuk mengetahui cara pelafalan dari tiap-tiap responden
3. Tahap ketiga melakukan riset database Al-Qur'an yang telah dilatinkan melalui sumber pustaka.
4. Tahap terakhir yaitu penulis mengumpulkan teks Al-Quran berdasarkan pedoman alih aksara Arab ke aksara Latin

### 3.4. Pengodean Fonetis Algoritma *Doublemetaphone*

Pengkodean kode fonetis dilakukan pada database/korpus Al-Qur'an yang sudah berupa aksara Latin. Data tersebut akan diproses untuk dijadikan kode fonetisnya, harus dilakukan normalisasi terlebih dahulu terhadap teks berupa tanda baca kevali apostrof (') dan karakter yang dimasukan hanya berupa *string query*. Algoritma *doublemetaphone* dibagi menjadi dua dalam pembentukan kode fonetisnya, yaitu kode fonetis primer dan sekunder. Kode fonetis pada *doublemetaphone* diubah sesuai dengan pelafalan dan ejaan bahasa Indonesia dan penulisan dalam aksara *Arab* yang telah ditraslasasikan dalam aksara Latin. Berikut ini merupakan aturan dalam pembentukan kode fonetis untuk algoritma *doublemetaphone* yang telah diubah kedalam aksara Latin sesuai dengan ejaan bahasa Indonesia.

Tabel 1 Aturan Kode Fonetis *Doublemetaphone* Penelitian

No.	Huruf	Kode Fonetis		Contoh	Keterangan
		Primer	Sekunder		
1	A	A	A	Amin	Hanya diawal String
2	B	B	B	Badrullah, Mim'bagdi	
3	C			Diabaikan	
4	D	D	D	Duha	
		Z	Z	Aladzinahum	Pelafalan huruf D melebur, jika huruf selanjutnya Z
5	E				Diabaikan, karena pelafalan ayat Al-Quran tidak mengenal huruf E
6	F	F	F	Faaridhun	
7	G	G	G	Magfirah	
8	H	H	H	Allah	
		D	D	Ba'dhukum	Pelafalan huruf H melebur, jika huruf sebelumnya D
		G	G	Ghaibas	Pelafalan huruf H melebur, jika huruf sebelumnya G
		T	T	Thahir	Pelafalan huruf H melebur, jika huruf sebelumnya T
		Z	Z	Azhlama	Pelafalan huruf H melebur, jika huruf sebelumnya Z
9	I	I	I	Insyaallah	Hanya diawal String
10	J	J	J	Mujahirin	
		Z	J	Dzalikal	
11	K	K	K	Anfusakum	
		H	K	Khatamallah	Pelafalan huruf K melebur, jika huruf selanjutnya H
12	L	L	L	Bighairil	
13	M	M	M	Aamanuu	
14	N	N	N	Muslimin	
		N	M	min ba'dii	Hukum bacaan Iqlab, maka dilafalkan M
15	O				Diabaikan
16	P	F	F	Fimaa	
17	Q	Q	Q	Qiyamulaah	
		K	Q	Walaqad	
18	R	R	R	Ajruhum	
19	S	S	S	Khaasirin	
20	T	T	T	Aataina	

		D	T	Ahad	
		S	S	Tsumma	Pelafalan huruf T melebur, jika huruf selanjutnya S
21	U				Diabaikan
22	V	F	F	Fiima	
23	W	W	W	Wamaa	
24	X				Diabaikan
25	Y	Y	Y	Yauman	
		S	S	Tasykuruun	
		I	I	Iyaakanabudu	
26	Z	Z	Z	Zulhaq	
27	ng (ikhfa)	N	N	Minhum	Aturan Hukum Ikhfa
28	'	X	X	Mu'minunn	Apostrof, tanda petik

Tabel diatas pengkodean fonetis dengan algoritma *doublemetaphone* sudah disesuaikan dengan penulisan aksara Latin dalam bahasa Indonesia. Tahap pengkodean fonetis dalam algoritma *doublemetaphone*, yaitu:

4. Mengabaikan karakter *vocal* pada *query input*, kecuali untuk huruf *vocal* diawal *string*
5. Berikan kode fonetis disetiap karakter pada *string query* masukan.
6. Pemberian kode fonetis hanya maksimal 4 karakter, apabila kode yang dihasilkan kurang dari 4 karakter maka tampilkan seadanya.
7. Karakter *string* akan ditransformasikan menjadi kode fonetis, apabila huruf yang sedang dilakukan pengecekan untuk eksekusi (ke-i) jika tidak sama dengan huruf pada karakter selanjutnya (ke-i+1), namun jika huruf yang akan dieksekusi memiliki huruf yang sama dengan karakter selanjutnya maka yang akan di eksekusi (ke-i+2).

### 3.3 Pencocokan Kode Fonetis

Proses pencocokan kode fonetis dilakukan berdasarkan kemiripan antara kode fonetis *query* dengan kode fonetis pada dataset/korpus. Pencocokan dilakukan dengan cara mencocokkan kode fonetis *query* pada setiap baris pada kolom *database* kode fonetis ayat Al-Quran. Kolom kode fonetis ayat Al- Quran ini berisi mengenai kata-kata pada sebuah ayat Alquran yang telah dikodekan menjadi kode fonetis. Pada setiap baris pada kolom akan dilakukan pengecekan untuk mengetahui apabila terdapat kode fonetis yang sama dengan kode fonetis *query*. Pengecekan dilakukan dengan cara mencocokkan kode fonetis di setiap kata-perkata pada dataset/korpus Al-Quran. Iterasi yang dilakukan untuk mendapatkan kode fonetis yang cocok dengan cara pengecekan antar tiap karakter pada kedua kode fonetis yang sedang dilakukan pencocokan. Jika keduanya memiliki kode yang sama maka data pada dataset tersebut akan dikeluarkan menjadi sebuah *output* sistem. Pengecekan dan pencocokan ini dilakukan secara terus menerus hingga semua data pada database dataset/korpus Al-Quran telah selesai diperiksa.

## 4. Pengujian dan Analisis

### 4.1. Tujuan Pengujian

Pengujian yang dilakukan dalam tugas akhir ini bertujuan untuk mengembangkan implementasi pengujian dari pencarian ayat Al-Qur'an. Program aplikasi tersebut akan mencocokkan hasil *query input* dari *user* terhadap korpus Al-Qur'an yang tersedia didalam database. *Input* maupun *output* sistem berupa teks Al-Qur'an dari transkripsi Latin. Pengembangan ini akan menggunakan algoritma *doublemetaphone*. Diharapkan dengan adanya pengujian ini akan diperoleh hasil dari presisi, *Recall*, dan kolerasi yang lebih akurat.

### 4.2. Hasil Pengujian dan Analisa Skenario Pertama

Skenario pertama ini akan dilakukan sebuah pengujian untuk mengetahui nilai presisi dari sistem berdasarkan *query input* dari *user*. Skenario ini akan membuktikan hasil bahwa sistem dapat menangani beberapa bentuk variasi penulisan. Tahapan dari skenario ini sebagai berikut:

1. *Input query* dengan berbagai macam variasi penulisan berdasarkan dari pelafalan ayat Al-Quran.
2. Aplikasi mengurutkan hasil keluaran berdasarkan kemiripan dengan *query inputan* secara *descending*, dimana posisi paling atas hasil keluaran adalah yang memiliki nilai kemiripan yang baik.
3. Diberikan *threshold*, dimana jika presentase nilai kemiripan kata yang dikeluarkan memiliki nilai >70% maka dianggap memiliki kemiripan dan nilai kemiripan < 70% maka kata dianggap tidak memiliki kemiripan.
4. Didapatkan kata dari hasil keluaran sistem yang berdasarkan *threshold* yang telah ditentukan.
5. Didapatkan nilai presisinya.

Skenario pertama didapatkan hasil pengujian seperti pada tabel dibawah ini.

Tabel 2 Hasil Uji Presisi

No	Query	Jumlah Hasil Output Sistem	Jumlah Output Berdasarkan Nilai Threshold >70%	Presisi
1	Famaa	61	18	0.30
2	Innaloha	37	37	1.00
3	Tsuma	100	26	0.26
4	Rohmatan	4	4	1.00
5	Yuminun	18	18	1.00
6	Sayton	10	9	0.90
7	Badukum	1	1	1.00
8	Saaiin	100	32	0.32
9	Jalika	59	59	1.00
10	Dunillohi	7	7	1.00
11	Musa	100	14	0.14
12	Waidza	100	32	0.39
13	Alladzina	21	21	1.00
14	Wamina	55	38	0,70
15	Robana	31	19	0,60
16	Kanna	100	23	0.23
17	Wamma	100	41	0.41
18	Kafaru	81	81	1.00
19	Ghafur	23	22	0.95
20	Innaka	14	14	1.00
21	Allamin	10	10	1.00
22	Lillahi	11	100	0.11
23	Dzalikal	3	3	1.00
24	Yauman	18	18	1.00
25	Minal	75	75	1.00
26	Tasykurun	4	4	1.00
27	Mutaqiin	6	6	1.00
28	Muslimin	11	11	1.00
29	Allahu	11	10	0.90
30	Walahum	13	13	1.00
Presisi rata-rata				0.77

Dari Skenario pengujian pertama yang terdapat pada tabel, didapat hasil presisi dari setiap *query*-nya. Presisi yang didapatkan setiap *query*-nya menghasilkan sebuah nilai presisi rata-rata yaitu 0.77. Hal ini menunjukkan bahwa sistem dapat menangani bentuk *query* yang diinputkan *user* yang bervariasi.

#### 4.3. Hasil Pengujian dan Analisa Skenario Kedua

Skenario kedua ini dilakukan pengujian untuk mencari nilai kolerasi dari hasil pengurutan *output* sistem. Hal ini dapat memberikan relevansi dari hasil *output* sistem. Tahapan dari skenario ini sebagai berikut:

1. *Query* ditentukan oleh penulis, kemudian dilakukan pengecekan oleh sistem.
2. Hasil pengecekan oleh sistem akan dikeluarkan berdasarkan urutan kemiripannya.
3. Penulis meminta 10 responden secara *random* untuk melakukan pengurutan berdasarkan *inputan* yang telah diproses oleh sistem.



4. Dilakukan perbandingan anantara pengurutan yang dilakukan koresponden dengan sistem.
5. Dilakukan perhitungan nilai korelasi, *range* nilai korelasi berkisar antara 1-(-1). Nilai 1 menunjukkan sistem memiliki korelasi yang baik dengan kebutuhan penggunaanya -1 menandakan sistem memiliki korelasi yang.

Skenario kedua didapatkan hasil pengujian seperti pada tabel dibawah ini.

Tabel 3 Uji Korelasi

No	Input	Output	Sorting sistem	Sorting user	Nilai Korelasi
Q1	Famaa	famaa faamaa faimmaa fiimaa	1 2 3 4	1 2 3 4	1
Q2	Innaloha	innallaha innallahaa	1 2	1 2	1
Q3	Tsuma	tsumma tsamma sami'ahu	1 2 3	1 2 3	1
Q4	Rohmatan	rahmatan rahmatin rahmati	1 2 3	1 2 3	1
Q5	Dunillohi	duunillahi diinillahi	1 2	1 2	1
Q6	Sayton	saiyaitan syaitaan sythan sathan	1 2 3 4	3 1 2 4	0.4
Q7	Saaiin	syaiin syaian siiniin	1 2 3	1 2 3	1
Q8	Jalika	dzalika dzalikal dzalikum	1 2 3	1 2 3	1
Q9	Robana	rabbanna rabbuna raibin	1 2 3	1 2 3	1
Q10	Kanna	kaana kaanu	1 2	1 2	1

No	Input	Output	Sorting sistem	Sorting user	Nilai Korelasi
		kuunuu	3	3	
Q11	Wamma	wamaa wamimmaa	1 2	1 2	1
Q12	Kafaru	kafaruu kafara kufira	1 2 3	1 2 3	1
Q13	Lillahi	lillahi lahu lahaa	1 2 3	1 2 3	1
Q14	Waidza	waidz waidzaa wauudzuu	1 2 3	2 1 3	0.5
Q15	Innaka	innaka innakum	1 2	1 2	1
<b>Rata-rata Nilai Korelasi</b>					<b>0.93</b>

Dalam pengujian skenario kedua ini dilakukan pengisian kuisioner yang dibantu oleh 10 responden yang telah mahir dalam membaca Al-Qur'an. Diberikan kuisioner yang berisi *query* yang telah ditentukan oleh penulis, *query* tersebut *dinputkan* kedalam sistem dan hasil dari *query* tersebut berupa *output* dari sistem yang dijadikan data pada kuisioner tersebut. Tujuan dari kuisioner tersebut untuk mencari tahu bagaimana dan keinginan dari *user* yang akan menggunakan sistem ini. Hasil kuisioner berupa daftar urutan dari setiap *query* yang diberikan. Daftar urutan tersebut berupa angka dari 1-4, angka 1 menunjukkan *query* tersebut berada di urutan paling atas, angka 2 dan 3 merupakan urutan selanjutnya, sedangkan angka 4 merupakan urutan paling rendah hasil *output query* sistem maupun pengurutan oleh responden. Dari hasil perbandingan urutan yang didapat dari *output query* sistem dengan kuisioner dari reponden akan didapatkan hasil nilai korelasi. Hasil tersebut dapat dilihat dari tabel 4.2 dengan nilai korelasi rata-rata sebesar 0.93 atau 93%. Nilai korelasi yang memiliki nilai baik bernilai 1 dan bernilai buruk 0.4. Dapat dilihat dai tabel tersebut bahwa Q6 bernilai 0.4 dimana faktor yang menyebabkan adalah pengurutan yang berbeda dari yang diberikan sistem dan pengurutan yang dilakukan oleh responden.

#### 4.4. Hasil Pengujian dan Analisa Skenario Ketiga

Skenario terakhir ini akan dilakukan pengujian terhadap ketepatan atau kemiripan sistem dalam memberikan hasil *output* yang diinginkan *user*. Skenario ini akan menghasilkan nilai yang didapatkan dari nilai *Recall*. Tahapan dari skenario ini sebagai berikut:

1. Penulis menentukan *query inputan* yang diuji.
2. Peneliti melakukan tahapan pencarian secara manual pada Al-Quran terhadap *query* yang diuji.
3. *Query inputan* diinputkan pada sistem.
4. Dilakukan perbandingan antara hasil *output* yang dilakukan oleh peneliti secara manual dengan hasil *output* sistem.
5. Hasil yang dicari berupa banyaknya jumlah kata beserta informasi yang terkandung.
6. Didapatkan nilai *Recall* dengan cara hasil jumlah *output* pada sistem dibagi dengan hasil pencarian secara manual.

Skenario ketiga didapatkan hasil pengujian seperti pada tabel dibawah ini.

Tabel 4 Pencarian Recall

No.	Query	Potongan Ayat	Jumlah Hasil Output Sistem	Jumlah Hasil Manual	Recall
1.	Famaa	مف	6	6	1.00
2.	Innalloha	إِنَّا اللَّهُ	35	37	0.95
3.	Tsuma	مَث	12	34	0.35
4.	Rohmatan	رَحْمَةً	2	2	1.00
5.	Musa	يَمْ	9	12	0.69
Nilai Recall Rata-Rata					0.80

Hasil pengujian ketiga ini sistem dapat cukup memberikan hasil yang diinginkan oleh *user*. Hasil nilai recal rata-rata adalah 0.80 atau 80%. Beberapa *query* tidak dapat mendapatkan hasil yang baik. Seperti pada *query* Tsuma, dari hasil *output* sistem terdapat 12 ayat yang memiliki kemiripan dengan *query inputan* sedangkan pada sistem manual terdapat 34 ayat. Terdapat perbedaan yang signifikan dari kedua hasil *output* sistem dengan hasil manual.

## 5. Kesimpulan

Berdasarkan hasil pengujian yang didapatkan dari analisis yang telah dilakukan, maka dapat ditarik kesimpulan sebagai berikut:

1. Berdasarkan hasil dari pengujian diatas, kinerja dari sistem pencarian ayat Al Qur'an berdasarkan kemiripan ucapan menggunakan algoritma *doublemetaphone* menghasilkan nilai presisi 0.77 korelasi 0.93 dan nilai *Recall* 0.88.
2. Terdapat pengembangan dalam aplikasi dimana *user* dapat melihat gambar dan suara dari potongan ayat yang hasil dari *query* sistem yang memiliki kemiripan dengan *query* yang diinputkan.

## Daftar Pustaka

- [1] A. B. S. M Denny Ikhsan, "Analisa dan Implementasi Pencarian Ayat Al-Quran Menggunakan Algoritma Doublemetaphone Berdasarkan Kemiripan Ucapan," 2016.
- [2] "Al-Qur'an," [Online]. Available: <https://id.wikipedia.org/wiki/Al-Qur%27an>. [Diakses 3 November 2016].
- [3] "Pedoma Alih Aksara Arab ke Latin," Wikipedia, [Online]. Available: [http://id.wikipedia.org/wiki/Wikipedia:Pedoman\\_alih\\_aksara\\_Arab\\_ke\\_Latin](http://id.wikipedia.org/wiki/Wikipedia:Pedoman_alih_aksara_Arab_ke_Latin). [Diakses 3 November 2016].
- [4] D. A. RI, dalam *Pedoman Transliterasi Arab Latin*, Jakarta, Departemen Agama RI, 2003.
- [5] E. V. Haryanto, Rancangan Bangunan Prototype Mesin Pencarian String Menggunakan Metode Fuzzy String Matching, Bali: Konferensi Nasional Sistem dan Informatika, 2011.
- [6] M. S. a. R. Muni, "Pencocokan String Berdasarkan Kemiripan Ucapan (Phonetic String Matching) dalam Bahasa Inggris," pp. 2-5, 2005.
- [7] B. a. Rex, dalam *Practical Algorithms for Programmer*, Addison Wesley, 1995.
- [8] R. M. Mokhammad Syaroni, "PENCOCOKAN STRING BERDASARKAN KEMIRIPAN UCAPAN (PHONETIC)," 2005.
- [9] N. Fachtiamsi, "Analisis dan Implementasi Algoritma Pencocokan String Berdasarkan Penulisan Dan Pengucapan Untuk Nama Indonesia," Universitas Telkom, Bandung, 2009.
- [10] P. Christen, "A Comparison of Personal Name Matching: Techniques and Practical Issues," Department of Computer Science, The Australian National University, Canberra.
- [11] M. A. Istiadi, "Sistem Pencarian Ayat Al-Quran Berbasis Kemiripan Fonetis," pp. 2-4, 2012.
- [12] B. & Rex, "Practical Algorithms, Addison Wesley., 1995.