Jurnal Linguistik Komputasional
Indonesian Association for Computational Linguistics
Vol 1, No 2
September
2018
e-ISSN :
2621-9336

# Sistem Pencarian Ayat Al-Quran Berdasarkan KemiripanUcapan Menggunakan Algoritma *Soundex* dan *Damerau-Levenshtein Distance*

Puruhita Ananda Arsaningtyas[#1], Moch. Arif Bijaksana[#2], Said Al Faraby[#3]

[#]*Fakultas Informatika, Universitas Telkom, Bandung*

[1]puruhitaananda@students.telkomuniversity.ac.id

[2]arifbijaksana@telkomuniversity.ac.id

[3]saidalfaraby@telkomuniversity.ac.id

*Abstrak*—**Penelitian ini mengimplementasikan dan analisis pencarian ayat Al-Quran dari kemiripan ucapan menggunakan algoritma *Soundex* dan *Damerau-Levenshtein Distance*. Pada pengucapan kata, sering kali ditemukan pengucapan kata yang sama atau mirip tapi memiliki penulisan yang berbeda, hal itu menjadi masalah ketika kita melakukan pencarian ayat pada Al-Quran. Dengan menggunakan algoritma *Soundex query* dan ayat dikodekan sesuai kemiripan fonetis tiap hurufnya, sehingga kata-kata yang memiliki pengucapan sama atau mirip dapat dianggap sama. Kemudian nilai kesamaan *query* dan *dataset* dihitung menggunakan algoritma *Damerau-Levenshtein Distance* kemudian diurutkan dari skor terendah. Berdasarkan pengujian yang dilakukan, didapatkan nilai MAP 0.78, *recall* 0.91, dan korelasi 0.82.**

***Kata kunci*— Inexact string matching, Al-Quran, *Soundex*, Damerau-Levenshtein Distance, kemiripan fonetis.**

***Abstract*—This research is to implement and analyze the search of Al-Quran verse from phonetic similarity using Soundex and *Damerau Levenshtein* Distance algorithm. It is often found the same or similar sound of a word but has a different spelling, it becomes a problem when we do a verse search on Al-Qur'an. By using the Soundex algorithm *query* and verses are encoded according to the phonetic similarity of eachletter, therefore words that have the same or similar pronunciation can be considered the same. Then the similarity value of *query* and dataset calculated using Damerau-Levenshtein distance algorithm and sorted from the lowest score afterward. Based on the tests performed, obtained the value of MAP 0.78, recall 0.91, and correlation 0.82.**

**Keywords—Inexact string matching, Al-Quran, phonetic similarity, Soundex, Damerau-Levenshtein Distance**

## I. PENDAHULUAN

Al-Quran adalah sebuah kitab suci bagi umat Islam dan juga pedoman hidup umat Islam yang diturunkan Allah S.W.T kepada nabi Muhammad S.A.W untuk umatnya. Al-Quran memiliki jumlah kata yang sangat banyak sehingga melakukan pencarian suatu kata pada Al-Quran secara manual sulit dilakukan. Telah banyak tersedia aplikasi pencarian ayat Al-Quran digital berbasis desktop, mobile, dan web. Tetapi masalah yang muncul yaitu pada pengguna yang ingin mencari potongan ayat Al-Quran tetapi tidak mengetahui penulisan asli potongan ayat tersebut, hanya mengetahui bunyinya saja. Karena pada pengucapan kata, seringkali ditemukan pengucapan kata yang sama atau mirip tapi memiliki penulisan yang berbeda. Sehingga, diperlukan solusi untuk mengatasi permasalahan tersebut. Pengembangan yang telah dilakukan yaitu phonetic string matching. Phonetic string matchingmerupakan algoritma pencocokan string berdasarkan kemiripan bunyinya. Terdapat beberapa algoritma *phonetic string matching*, salah satunya algoritma *Soundex*. Algoritma *Soundex* mengelompokan tiap huruf yang memiliki satu kesamaan atau kemiripan pengucapan dalam satu kode. Pada tahun 2006, terdapat sebuah pengembangan algoritma *Soundex*, yaitu Arabic-*Soundex*. Algoritma ini bertujuan untuk mencari nama yang tertulis dengan bahasa dan aksara Arab [1]. Kemudian, pada tahun 2012 dilakukan pengembangan algoritma Arabic-*Soundex*. Pengembangan dilakukan dengan tujuan untuk menciptakan lingkungan yang sesuai untuk pengambilan informasi yang relevan khusus untuk pengucapan lokal bahasa Arab [2]. Pada penelitian ini, diterapkan phonetic string matching untuk pencarian ayat transliterasi Al-Quran menggunakan algoritma *Soundex* yang berdasarkan penelitian sebelumnya.

Hal yang menjadi tantangan untuk penelitian ini adalah, bagaimana membuat sistem yang baik untuk menangani transliterasi Al-Quran dengan pengucapan lokal bahasa Indonesia. Pencocokan string yang digunakan adalah algoritma *Damerau-Levenshtein Distance*. Konsep algoritma ini sama dengan algoritma Levenshtein Distance, perbedaanya adalah adanya penghitungan transposisi terhadap satu karakter [3]. Algoritma *Damerau-Levenshtein Distance* dilipih karena dapat

Submitted 21-08-2018; Revised 17-09-2018; Accepted 21-09-2018

meningkatkan efisiensi ruang dan waktu [4], dapat mengatasi pemenggalan kata yang berbeda, dan memiliki hasil yang lebih baik dibandingkan Levenshtein Distance [4]. Meskipun algoritma *Soundex* dan *Damerau-Levenshtein Distance*a dalah fokus dari penelitian ini, beberapa metode lain akan dilibatkan secara tidak langsung untuk membandingkan kinerja algoritma yang paling baik dari hasil pengujian berdasarkan metrik evaluasi yang ditentukan. Aplikasi Lafzi [5] dan Islamicity [6] akan diukur MAP dan *recall*nya, sebagai *baseline*. Untuk melihat apakah sistem yang dibangun sudah cukup baik atau belum. Karena Lafzi dan Islamicity adalah penelitian yang sudah dipublikasi dan digunakan oleh umum.

Topik pada penelitian ini adalah sistem pencarian ayat Al-Quran berdasarkan kemiripan ucapan. Algoritma pencocokan fonetis yang digunakan yaitu *Soundex* yang dimodifikasi mengikuti aksara Arab dan hukum bacaan Al-Quran.String ayat dan *query* akan diubah menjadi kode fonetis menggunakan algoritma *Soundex*. Kemudian, penghitungan jarak string menggunakan *Damerau-Levenshtein Distance*. Input berupa *query* potongan ayat Al-Quran transliterasi latin. Pedoman transliterasi yang digunakan yaitu SKB Menteri Agama dan Menteri PK RI No. 158 Tahun 1987 dan No. 0543b/U/1987. Output berupa ayat Al-Quran yang sudah terurut dari nilai similarity tertinggi. Similarity menggunakan FOM (Figure of Merit). Sistem yang dibuat berupa console (tidak menggunakan GUI). Dataset yang digunakan untuk pengujian terdiri dari 6 juz Al-Quran yaitu juz 1-5 dan juz 30. Setelah implementasi, dilakukan pengujian dengan metrik evaluasi yaitu MAP (Mean Average Precision), recall, dan korelasi.

Tujuan dari penelitian tugas akhir ini adalah untuk mengimplementasikan metode *Soundex* dan *Damerau-Levenshtein Distance* pada transliterasi ayat Al-Quran, dan melakukan pengujian serta analisis dari hasil pengujian. Evaluasi hasil pengujian menggunakan metrik berupa MAP (*Mean Average Precision*), *recall*, dan korelasi.

## II. STUDI TERKAIT

### A. *Damerau-Levenshtein Distance*

*Damerau-Levenshtein Distance* adalah algoritma untuk mengukur jarak *edit* dua buah *string*. Konsep algoritma ini sama dengan algoritma *Levenshtein Distance*, yaitu terdapat operasi penambahan, penghapusan, penggantian. Tetapi, *Damerau-Levenshtein Distance* memiliki perbedaan dengan *Levenshtein Distance* yaitu adanya operasi transposisi. *Damerau-Levenshtein Distance* optimal digunakan untuk pembuktian keaslian dan menangani kesalahan eja. Berikut adalah pseudocode Algoritma *Damerau-Levenshtein Distance* pada Python [7]:

```
def demLev (substr, query):
    cost = 0
    sizeX = len(query)+1
    sizeY = len(substr)+1
    matrix = np.zeros ((sizeX,sizeY))
    for x in xrange(sizeX):
        matrix [x, 0] = x
    for y in xrange(sizeY):
        matrix [0, y] = y
    for x in xrange (1, sizeX):
        for y in xrange (1, sizeY):
            if query[x-1] is substr[y-1]:
                cost = 0
            elif query[x-1] is not substr[y-1]:
                cost = 1
            matrix[x,y] = min(matrix[x-1,y]+1 #deletion
                        , matrix [x, y-1] + 1 #insertion
                        , matrix [x-1, y-1] + cost #substitution
                        )
            if x>=1 and y>=1 and (query[x-1] is substr[y-2]) and
            (query[x-2] is substr[y-1]):
            #transposition
                Matrix [x, y] = min (matrix[x, y],matrix[x-2,y-
2]+cost)
        return matrix.item (sizeX-1,sizeY-1)
```

### B. *Algoritma Soundex*

*Phonetic string matching* termasuk dalam *inexact string matching*. Algoritma yang mengidentifikasi serangkaian *string* mana saja yang pengucapannya mirip dengan serangkaian *query* yang diberikan [8] [9]. Algoritma *Soundex* merupakan salah satu algoritma *phonetic* yang dikembangkan oleh Odell dan Rusell, dan dipatenkan padatahun 1980. *Soundex* menggunakan kode berdasarkan kemiripan bunyi tiap hurufnya [8] [10]. *Soundex* menghasilkan kode fonetik dengan panjang maksimal empat karakter untuk setiap kata [9]. Dimana huruf - huruf yang mirip pengucapannya dikelompokkan menjadi satu kelompok dengan ditandai kode fonetis yang sama. Kode fonetis *Soundex* terdiri dari satu buah huruf yang merupakan huruf depan dari sebuah *string* dan diikuti maksimal tiga kode fonetis [8].

### C. *Soundex Modifikasi*

Pada tahun 2012, dilakukan pengembangan dari adaptasi algoritma *Soundex* bahasa Inggris untuk bahasa Arab, dikhususkan untuk pelafalan lokal arab. Algoritma *Arabic-Soundex* mengklasifikasikan bunyi menjadi lima bagian utama berdasarkan bagaimana cara pengucapannya, yaitu [10]:

1. *Al-Jauf (Abdominal)*
   Huruf vokal panjang Bahasa Arab:
   الألف المدية , الواو المدية , الياء المدية
2. *Lisani (Dental)*
   ق, ج, ش, ل, ن, ر, ت, د, ز س, ص , ظ, ذ, ث, ك
   ط, ض ,ي

3. *Halqi (Velar)*
   أ, ه ,ح ,ع , خ , غ
4. *Syafawi (Labial)*
   ب ,م,و,ف
5. *Khoisyum (Nasal)*
   Hukum bacaan iqlab, ikhfa, idhgom.

Tujuan dari sistem ini adalah pencarian transliterasi ayat Al-Quran untuk ejaan orang Indonesia, sehingga perlu dibuat modifikasi dalam klasifikasi bunyi tiap hurufnya. Berikut adalah modifikasi pembagian klasifikasi bunyi tiap huruf [1] [9]:

TABEL I
KODE FONETIS *SOUNDEX* VERSI INDONESIA

| No. | Huruf | Kode Fonetis |
|---|---|---|
| 1. | E, I, U, Y | / |
| 2. | A, O | $ |
| 3. | B, P | 0 |
| 4. | C, J, S, X, Z | 1 |
| 5. | D | 2 |
| 6. | F(V) | 3 |
| 7. | G, K(Q) | 4 |
| 8. | L | 5 |
| 9. | M | 6 |
| 10. | N | 7 |
| 11. | R | 8 |
| 12. | T | 9 |
| 13. | W | @ |
| 14 | H | # |

Huruf E, I, U, dan Y dikategorikan pada kode fonetis yang sama, karena merupakan huruf vokal. Huruf Y dalam aksara latin memiliki kemiripan pengucapan dengan huruf I, sehingga dikategorikan pada koe fonetis yang sama. Huruf A dan O merupakan huruf vokal namun pengucapannya berbeda, ddan huruf O tidak terdapat pada aksara Arab, tetapi jika ada kata yang dibaca dengan O pada ayat Alquran dituliskan dengan A, contohnnya Allah, dibaca Alloh.

D. *Metrik Evaluasi*

***Mean Average Precision* (MAP)**

MAP adalah skor yang didapat dari pengukuran kinerja sistem pada serangkaian kueri. Kategori MAP yang baik terlihat dari hasil AP. Jika keseluruhan AP memiliki nilai yang tinggi artinya presisi sistem baik, karena informasi relevan yang dioutputkan oleh sistem memiliki urutan yang berdekatan. Jika nilai AP rendah, artinya presisi sistem buruk, karena informasi relevan yang dioutputkan tidak pada urutan yang berdekatan, dan memiliki jumlah output *false positif* yang banyak. Cara mendapatkan skor MAP adalah dengan menghitung nilai rata - rata dari *average precision* pada tiap kueri [3].

$$MAP = \frac{1}{|Q|} \cdot \sum_{Q_i \epsilon Q} AP(Q_i) \quad (1)$$

|Q| adalah jumlah set *query* yang dihitung. AP(Qi) adalah nilai *average precision* untuk *query* ke i. Sebelum menghitung MAP, terlebih dahulu harus dihitung nilai *average precision* tiap *query*. *Average precision* merupakan pengukuran presisi dengan cara menghitung urutan dokumen yang relevan pada output sistem hingga nilai *recall* dari dokuemn sama dengan 1.

$$AP = \sum_{i=1;i\epsilon N}^{N} \frac{(TP\ ke{-}i)}{peringkat\ TP\ ke{-}i} \quad (2)$$

N adalah jumlah output dari sistem hingga recall= 1 atau keseluruhan output. TP adalah True Positif atau output yang tepat dari sistem dandata asli. TP ke-i artinya peringkat output yang sesuai. Peringkat TP ke-i artinya ada diurutan ke berapa output yang sesuai itu.

***Recall***

Merupakan probalitas informasi yang relevan yang di *retrieve* dibandingkan dengan jumlah informasi keseluruhan yang relevan. Skor dari *recall* menentukan tingkat keberhasilan sistem dalam melakukan pencarian ayat. Recallbernilai 1 jika informasi relevan seluruhnya dioutputkan oleh sistem. Jika nilai recall sistem adalah 1 artinya sistem berhasil melakukan pencarian informasi berdasar *query* yang ada. Berikut adalah notasi penghitungan *recall* [7]:

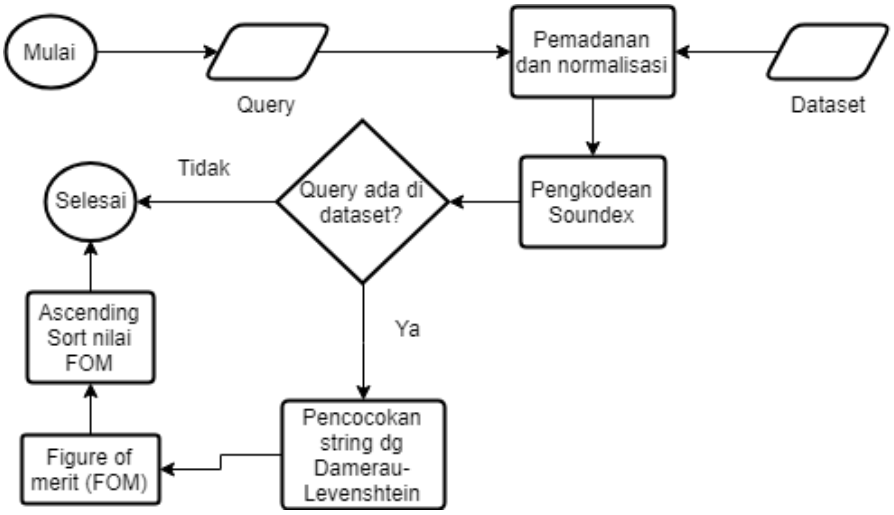
$$Recall = \frac{|\{Informasi\ relevan\} \cap \{Informasi\ ter{-}retrieve\}|}{|Informasi\ relevan|} \quad (3)$$

**Korelasi**

Nilai *similarity* sangatlah relatif, sehingga tidak bisa ditentukan mana saja informasi dari dataset yang cocok dengan *query* [4]. Maka dari itu, diperlukan sebuah metrik untuk mengukur keterkaitan antara dua variabel. Kuatnya hubungan antara variabel dinyatakan dalam koefisien relasi. Koefisien relasi terbesar bernilai 1, dan terkecil bernilai -1. Semakin besar nilai koefisien relasi, artinya hubungan antara variabel semakin kuat [6].

## III. SISTEM YANG DIBANGUN

Pada penelitian tugas akhir ini, terdapat dua proses utama, yaitu pengkodean *Soundex*, dan penghitungan *Damerau-Levenshtein Distance*. Gambar 1 adalah bagan alur sistem:




**Gambar 1** Alur Pengkodean Sistem

**Dataset**

*Dataset* pengujian sistem berupa Al-Quran transliterasli latin. Dengan jumlah 6 juz, yaitu Al-Quran juz 1 – 5 dan juz amma (juz 30). File Al-Quran transliterasi yang digunakan berekstensi .txt. Dataset didapat dari penelitian sebelumnya [9] dan ditambahkan secara manual sebanyak dua juz dengan berpedoman pada alih aksara Arab-Latin Indonesia [8]. Atribut pada dataset berupa nomor surat, nama surat, nomor ayat, dan isi ayat. Tiap baris pada dataset menunjukan satu ayat, sehingga pemisah antar ayat yaitu baris baru. Pada saat sistem dijalankan, dataset akan diload dari direktori lokal. Contoh isi dataset Al-Quran transliterasi:

> 3,Ali-Imran,171,Yastabsyiruuna bini'matin minallahi waafadhlin wa-anallaha laa yudhii'u ajral mu'miniin(a)

Dataset akan digunakan untuk pencarian kesamaan string degan satu set *query*. *Query* didapatkan dari responden muslim yang memiliki kemampuan membaca huruf hijaiyah pada Al-Quran setidaknya cukup. Responden mengisi quisioner dengan cara menuliskan transliterasi pada sejumlah potongan ayat Al-Quran dengan aksara Arab. Sehingga satu set *query* memiliki beberapa variasi untuk setiap *query*nya. *Query* terkumpul sebanyak 19 buah, dengan jumlah variasi tiap *query* yang berbeda - beda. Pemilihan *query* berdasarkan lima kategori bunyi pada bahasa Arab.

**Preprosesing**

Sebelum dataset dan *query* di kodekan, dilakukan preprosesing terlebih dahulu. Preprosesing dilakukan pada program, sehingga tidak perlu dilakukan secara manual. Tujuannya agar teks pada dataset dan *query* seragam dan memudahkan dalam proses selanjutnya. Preprosesing yang dilakukan yaitu:

1. Case Folding: mengubah semua huruf menjadi huruf kecil
2. Menghapus akhiran kata, mengubah huruf sesuai dengan penyerapan kata, dan pemadanan aksara latin sesuai tajwid berdasarkan pedoman [8]. Seluruh aturannya dituliskan pada program di satu fungsi yaitu *normalize*.

Sehingga, output string setelah preprosesing seluruhnya sepadan dan sesuai dengan pedoman alih aksara. Sebagai contoh, string fa'innamaa 'alayka, pemadanan katanya menjadi: fainnamaa alayka.

**Pengkodean *Soundex***

Kemudian, pengkodean *Soundex* dilakukan sesuai dengan tabel di bab 2. Kode pada pengkodean *Soundex* telah disesuaikan sesuai kemiripan bunyi masing - masing karakter. Berikuta dalah urutan pengkodean *Soundex*:

1. Ayat dan *query* dipisahkan per kata (dengan asumsi pemisah adalah spasi' '). Ubah setiap karakter pada kata menjadi kode sesuai dengan tabel di bab 2.
2. Hapus karakter kode yang berulang (beruntun sama lebih dari satu karakter). Juga hapus kode "/" pada string perkata.
3. Hanya tampilkan empat string pertama pada string kode, agar sesuai aturan algoritma *Soundex*.

Contoh: string fainnama alaika menjadi 3$/77$6$$ $5$/4$. Lalu, munculkan awal huruf dari tiap kata. Sehingga string berubah menjadi f$/77$6$$ a5$/4$. Setelah itu, buang karakter yang duplikat, string berubah menjadi f$/7$6$ a5$/4$. Terakhir, hilangkan karakter "/" danoutputkan maksimal 4 karakter pertama yang muncul. Hasil pengkodean dari string fainnamaa alayka adalah f$7$ a5$4.

**Pencocokan String**

ahap ketiga yaitu *Damerau-Levenshtein Distance* untuk mengukur skor kesamaan antara dua buahstring. Pada tahap ini setiap kata pada *query* dan dataset dilakukan pencocokan sehingga didapatkan jarak edit distance. Pencocokan yang dilakukan yaitu: 1) string kata yang sebenarnya dan 2) *string* kode fonetis. Dilakukan dua kali pencocokan untuk menghitung nilai FOM (Figure of Merit), yaitu perangkingan hasil pencarian. Terdapat empat operasi pada algoritma *Damerau-Levenshtein Distance*, pada dasarnya mencari nilai minimum dari operasi yang dilakukan. Penghitungan dilakukan pada matriks diagonal berukuran mxn (m dan n adalah panjang masing-masing string yang dibandingkan). Hasil dari penghitungan ini adalah pada matriks [m, n]. Skor yang didapat merepresentasikan jumlah karakter yang berbeda dari string pembanding.

Sebagai contoh, string fa‘innamaa ’alayka dan wa‘innamaa anaa, dikodekan *Soundex* menjadi f$7$ a5$4 danw$7$ a7$. Jika setiap string dihitung jaraknya, pada dataset dan *query* string dapat berupa satu kata atau lebih. Pada penelitian ini, satu inputan *query* dianggap satu variabel string, dan satu ayat pada dataset Al-Quran dianggap satu *string*. Tabel awal edit distance string f$7$ a5$4 dan w$7$ a7$.

| | | | | |
|---|---|---|---|---|
| 0 | f | $ | 7 | $ |
| w | | | | |
| $ | | | | |
| 7 | | | | |
| $ | | | | |

Matriks awal jarak edit

| | | | | |
|---|---|---|---|---|
| 0 | f | $ | 7 | $ |
| w | 1 | 2 | 3 | 4 |
| $ | 2 | 1 | 2 | 3 |
| 7 | 3 | 2 | 1 | 2 |
| $ | 4 | 3 | 2 | 1 |

Hasil akhir matriks jarak edit

Misalkan m adalah string f$7$ dan n adalah string w$7$. Sehingga panjang string m dan w adalah 4, maka matriksnya 5x5. Terdapat satu pengubahan suatu string yaitu penyisipan, penghapusan, dan substitusi, dengan masing – masing nilai berbeda. Penyisipan: (i,j-1) + 1, penghapusan : (i-1,j) + 1, substitusi : (i-1,j-1) + cost. Dimana, i adalah iterasi panjang string m, dan j adalah iterasi panjang string n. Cost bernilai 0 apabila m[i] ==m[j], jika tidak maka cost = 1 Nilai yang akan menjadi jarak edit adalah matriks [5,5]. Sehingga jarak edit untuk stringf$7$ a5$4 dan w$7$a7$ adalah 1. Artinya, hanya ada satu perbedaan pada string f$7$ dan w$7$, yaitu karakter ’f’ dan ’w’.

**Perangkingan**

Pada pengkodean *Soundex* string yang dikodekan biasanya lebih pendek dari kata aslinya, sehingga dapat mengakibatkan hasil penyamaan yang tidak relevan. Untuk menghindari permasalahan tersebut, dibutuhkan komponen lain untuk memperkecil output yang tidak relevan dan untuk meranking output sesuai dengan relevansinya. Pada tahun 2005, di usulkan sebuah cara untuk mengatasi hal - hal tersebut, yaitu menggunakan *Figure of Merit* (FOM) [11]. Nilai FOM didapatkan dengan menghitung nilai *edit-distance*, dan *phonetic-edit distance* yang didapatkan dari metode *Damerau-Levenshtein Distance*.

$$EDScr = \frac{(MaxLeng(ayat,query) - ED)}{MaxLeng(ayat,query)} \quad (4)$$

$$PEDScr = \frac{(MaxLeng(Sayat,Squery) - PED)}{MaxLeng(Sayat,Squery)} \quad (5)$$

$$FOM = \frac{PEDScr + \frac{EDScr}{10}}{1{,}1} \quad (6)$$

Setelah proses tiga dilakukan, didapatkan nilai perbedaan dari *string* asli, dan jarak *edit* pada *string* kode fonetis. Nilai tersebut digunakan untuk menghitung skor edit distance kata asli (EDScr), dan skor edit distance kode fonetis (PEDScr). Setelah kedua skor didapat, EDScr dan PEDScr digunakan untuk menghitung FOM. Nilai FOM menjadi nilai akhir dari sistem yang digunakan untuk mengurutkan tingkat ”relevansi” dari output sistem. Semakin besar nilai FOM secara kasar dapat dikatakan outputnya semakin relevan.

## IV. EVALUASI

Setelah sistem dibuat, dilakukan tahap pengujian. Pengujian dilakukan dengan menghitung nilai MAP, *recall*, dan korelasi dari *output* program dengan satu set *query* yang telah ditentukan. *Dataset* yang digunakan untuk pengujian adalah Al-Quran transliterasi Latin Juz 1-5, dan Juz 30.

### A. *Hasil Pengujian*

Dengan jumlah dan variasi *query* yang sama, didapat nilai MAP, *recall*, dan korelasi dari sistem dengan metode – metode yang berbeda serta aplikasi Lafzi dan Islamicity. Berikut adalah hasil MAP dan Recall dari setiap sistem:

Tabel II memperlihatkan hasil MAP dan *Recall* dari beberapa algoritma *phonetic* bertujuan untuk melihat perbandingan hasil dari tiap algoritma.

TABEL II
HASIL MAP DAN *RECALL*

| No. | Label | Sistem | MAP | *Recall* |
|---|---|---|---|---|
| 1. | S1 | *Soundex*-Damerau Levenshtein | 0.79 | 0.91 |
| 2. | S2 | *Soundex*-Cosine Similarity | 0.82 | 0.80 |
| 3. | S3 | Metaphone-Levenshtein | 0.76 | 0.81 |
| 4. | S4 | Double Metaphone-Jaro Winkler | 0.84 | 0.89 |
| 5. | S5 | Ngram (Lafzi) | 0.88 | 1 |
| 6. | S6 | Islamicity | 0.72 | 0.87 |

Tabel III adalah nilai *average precision* dari setiap *query* :

TABEL III
HASIL AP TIAP *QUERY*

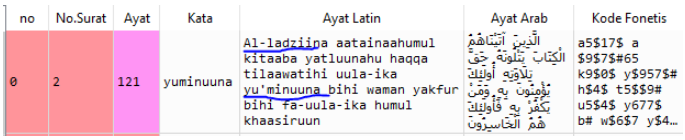
| *Query* Pendek | | *Query* Panjang | |
|---|---|---|---|
| Label | AP | Label | AP |
| Q1 | 1 | Q2 | 0.583 |
| Q6 | 1 | Q3 | 0.33 |
| Q7 | 0.353 | Q4 | 0.83 |
| Q10 | 1 | Q5 | 1 |
| Q12 | 0.925 | Q8 | 1 |
| Q13 | 1 | Q9 | 0.75 |
| Q15 | 1 | Q11 | 0.66 |
| Q17 | 1 | Q14 | 0.13 |
| Q19 | 0.66 | Q16 | 1 |
| | | Q18 | 0.67 |

Pengukuran nilai korelasi sistem yaitu sebesar 0.82.

B. *Analisis Hasil Pengujian*

Dari hasil pengujian yang dilakukan analisis yang dapat diberikan adalah sebagai berikut:

1. Dari hasil AP masing – masing *query*, jika kode fonetis *query* unik maka outputnya akan semakin relevan. Kode fonetis unik artinya satu kode fonetis dimiliki satu kata asli saja. Skor AP Q6, dan Q17 bernilai 1, memiliki kode fonetis yang unik. AP Q1, Q10, dan Q13 bernilai 1 tetapi bukan kode fonetis unik, kode fonetis Q13 (khairun) yaitu h$87 dimiliki oleh kata khairun, khairin, dan haruuna. Skor AP yang tinggi artinya ayat relevan yang dioutputkan oleh sistem memiliki urutan yang berdekatan, meskipun dengan penulisan *query* yang berbeda – beda.
   Hasil AP pada *query*panjang cenderung lebih rendah. Ada beberapa faktor penyebabnya: 1) pengkodean pada sistem menangani *string matching* perkata bukan per kalimat, 2) pengecekan apakah *query* terdapat pada kata di ayat, jika kode fonetis hanya berbeda satu karakter tidak akan dihitung skor kemiripannya, 3) mapping fonetis tidak mengubah huruf pertama dari kata, contohnya kata mimma dan wamimma, secara manual dua kata tersebut kemiripannya tinggi, tetapi karna kode fonetisnya berbeda (mimma:m6$; wamim-ma: w$66$) sistem tidak dapat mendeteksi kemiripan kata tersebut.
   *Query* Q2, Q3, Q14, dan Q18 memiliki nilai AP yang rendah. Pada Q2 terdapat variasi humfiihakholiduun yang tidak dapat terdeteksi *query* fonetisnya pada ayat, karna faktor 1, 2, dan 3. Begitu pula variasi *query* Q3 yaitu ’adzabunmuhiin, variasi *query* Q14 yaitu hoyrunlakum, dan variasi *query* Q18 yaitu wa mimmaa rojaqnaahum. Selain itu, *string matching* yang dilakukan perkata dapat terjadi sebuah kasus dimana dua kata pada satu *query* ditemukan pada dataset tapi tidak bersebelahan. Sebagai contoh, Q4 yaitu Al-ladziina yu’minuuna.



| no | No.Surat | Ayat | Kata | Ayat Latin | Ayat Arab | Kode Fonetis |
|---|---|---|---|---|---|---|
| 0 | 2 | 121 | yuminuuna | Al-ladziina aatainaahumul kitaaba yatluunahu haqqa tilaawatihi uula-ika yu'minuuna bihi waman yakfur bihi fa-uula-ika humul khaasiruun | الَّذِينَ آتَيْنَاهُمُ الْكِتَابَ يَتْلُونَهُ حَقَّ تِلَاوَتِهِ أُولَئِكَ يُؤْمِنُونَ بِهِ وَمَنْ يَكْفُرْ بِهِ فَأُولَئِكَ هُمُ الْخَاسِرُونَ | a5$17$ a $9$7$#65 k9$0$ y$957$# h$4$ t5$$9# u5$4$ y677$ b# w$6$7 y$4… |

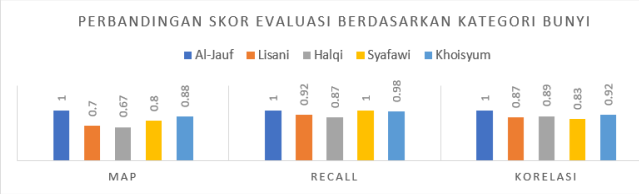
MAP sistem cukup baik, karena keseluruhan AP memiliki nilai yang bervariasi tetapi cenderung tinggi. Nilai MAP dari sistem ini yaitu 0.779, nilai AP yang tinggi banyak terdapat pada *query* – *query* pendek, AP pada *query* – *query* panjang nilainya bervariasi.

2. Rata-rata nilai recall dari sistem adalah 0.91. Nilai recall yang dihitung yaitu variasi *query* yang telah ditentukan. Recall bernilai 1 karena ayat relevan seluruhnya dioutputkan oleh sistem, pada seluruh variasi *query* tersebut. Recall pada variasi *query* tertentu bernilai 0, karena sistem tidak mengoutputkan apapun. Contohnya variasi *query* ’adzabunmuhiin, Qoolaatastabdilun, dan hoyrunlakum. Recall pada *query* mimmaa (Q12) bernilai 0.95 karena ada satu ayat relevan yang tidak dioutputkan. Kata pada ayat tersebut adalah wamimma, karena mimmaa dan wamimmaa memiliki kode fonetis yang berbeda (mimma: m6$; wamimma: w$66$). Secara keseluruhan sistem berhasil melakukan pencarian ayat yang relevan.
3. Pada hasil pengujian, didapatkan rata – rata korelasi sebesar 0.3. Pengujian untuk korelasi dilakukan dengancara mengurutkan kembali output sistem. Output sistem disusun secara acak kemudian user mengurutkan output tersebut secara manual. Jika urutan dari sistem dan pengurutan user sama, maka korelasi bernilai 1. Nilai maksimal dari korelasi adalah 1 dan nilai minimumnya -1.

| Variasi Input | Output | *Sorting* Sistem | *Sorting* Manual | Nilai Korelasi |
|---|---|---|---|---|
| Hum fiihaa khoolidun | Hum fiihaa khaaliduuna | 1 | 1 | 0.40 |
| | Khalidina fiihaa | 2 | 3 | |
| | Khalidan fiihaa | 3 | 4 | |
| | Wa-hum fiihaa khaaliduuna | 4 | 2 | |
| mimma | mimma | 1 | 2 | 0.5 |
| | mimmaa | 2 | 1 | |
| | mimman | 3 | 3 | |

Pada *query* hum fiiha khoolidun recall bernilai 0.4 karena terdapat perbedaan pengurutan pada string wa-humfiiha khaalidunna, sehingga menyebabkan ketidak sesuaian pada urutan dibawahnya. Faktor penyebabnya adalah kode fonetis yang berbeda antara hum fiiha khaaliduuna dan wa-hum fiiha khaaliduuna. Membuat nilai similarity (FOM) yang dihasilkan rendah, dan peringkat yang rendah. Pada *query* mimma recall bernilai 0.5 karena keterkaitan *sorting* manual dan sistem berbeda pada urutan ke-1 dan ke-2. Sistem mengurutkan kata mimma di urutan pertama karena *query* inputan adalah mimma, meskipun yang dimaksud adalah kata mimmaa. Pada beberapa *query* lain nilai korelasi 0 karena tidak ada output dari sistem.

4. Pada pengujian ini jumlah *query* tiap kategori bunyi disamakan, yaitu tiga *query* masing – masing kategori bunyi. Tujuannya adalah agar tidak ada ketimpangan skor metrik pengujian.




Kategori bunyi Al-Jauf (Abdominal) memiliki nilai MAP 1, recall 1, dan korelasi 1. Hasil AP semua *query*nya adalah 1, karena kode fonetis setiap *query* unik hanya dimiliki oleh kata-kata tersebut. Kategori Lisani (Dental) memiliki nilai MAP 0.70, recall 0.92, dan korelasi 0.87. Pada *query* qaala nilaiAP yang didapat dari variasi *query* nya 0.27-0.44. Hal ini disebabkan karena kode fonetis *query* qaala tidak unik. Kode fonetis qaala yaitu k5, sedangkan untuk kata kala, qaalat dan qaalati memiliki kode fonetisyang sama. Rule pada pemadanan kata mengubah q menjadi k. Rule pemadanan bergantung pada mapping fonetis. Huruf-huruf yang masuk kategori Lisani banyak yang dijadikan satu kelompok karena kemiripan pengucapannya, contohnya yaitu huruf kof dan kaf; shin, syin, shod, dan tsa; tho, dan ta.
Kategori Khoisyum (Nasal) memiliki nilai MAP 0.88, recall 0.98, dan korelasi 0.92. Terdapat variasi *query* waintawallaw fa'innama dengan AP 0.31. Pada variasi *query* mimma nilai AP yaitu 0.86, karena ada kata wamimmaa pada dataset yang memiliki kode fonetis berbeda, sehingga meskipun ditemukan kata wamimmaa, ranking ayat tersebut rendah. Tidak ada kesalahan pada pemadanan kata dan mapping fonetis kategori khoisyum. Kategori bunyi Halqi (Velar) memiliki nilai MAP 0.67, recall 0.875, dan korelasi 0.89. Nilai AP pada *query* khairun lakum yaitu 0.133, disebabkan karena adanya variasi *query* hoyrunlakum. *Query* 'ilmi dan minal 'ilmi memiliki AP masing – masing sebesar 1. Hal ini dapat dikarenakan pada kategori bunyi halqi tidakada karakter yang diulang, karna hurufnya dibaca jelas dimulut. Kategori bunyi Syafawi (Labial) memiliki nilai MAP 0.80, recall 1, dan korelasi 0.83. Pada kategori syafawi tedapat huruf hijaiyah fa dan ba, jika dipadankan maka akan menjadi f dan b. Kemudian jika ada huruf pakan dipadankan menjadi huruf f, sedangkan pada mapping *Soundex*, huruf p satu kategori dengan huruf b, akan terjadi ambiguitas.

## V. KESIMPULAN

Berdasarkan pengujian yang dilakukan sistem pencarian ayat Al-Quran berdasarkan kemiripan suara menggunakan algoritma *Soundex* dan *Damerau-Levenshtein Distance* menghasilkan nilai MAP sebesar 0.79, recall sebesar 0.91, dan korelasi sebesar 0.82. Algoritma *Soundex* modifikasi Arab untuk pengucapan Bahasa Indonesia baik digunakan pada *query* pendek dan sedikit kurang baik untuk *query* panjang. Besar kecilnya nilai AP tiap *query* dipengaruhi oleh keunikan kode fonetis kata, string matching perkata, pengecekan apakah *query* terdapatpada kata di ayat, mapping fonetis tidak mengubah huruf pertama dari kata. Dari skor rata – rata recall dapat dikatakan sistem berhasil melakukan pencarian ayat yang relevan. Dari hasil penghitungan korelasi menunjukan keterkaitan antara pengurutan oleh sistem dan pengurutan manual cukup tinggi. Pada kategori lisani terdapat huruf– huruf hijaiyah yang memiliki kesamaan pengucapan sehingga membuat ambigu pada pencocokan kata. Perludilakukan pengubahan pemadanan kata untuk menghilangkan ambiguitas. Kategori bunyi dengan nilai evaluasi terbaik adalah Al-Jauf, karena tidak terdapat kendala pemadanan kata atau pengkodean fonetis. Secara keseluruhan algoritma *Soundex* dan *Damerau-Levenshtein Distance* sudah cukup baik dalam menangani pencarian ayat Al-Quran transliterasi dengan pengucapan lokal bahasa Indonesia.

Adapun beberapa saran yang dapat digunakan untuk penelitian selanjutnya, yaitu melakukan penelitian terhadap algoritma *Soundex* untuk aksara Arab dengan transliterasi Bahasa Indonesia. Kemudian, menambahkan

jumlah ayat pada *dataset* Al-Quran untuk pengujian. Serta, membuat variasi *query* yang lebih banyak dan beragamuntuk pengujian.

## REFERENSI


[1] M. E. Yahia, M. E. Saeed and A. M. Salih, "An Intelligent Algorithm For Arabic Soundex Function Using Intuitionistic Fuzzy Logic," vol. 1, 2006.

[2] N. D. Ousidhoum, A. Bensalah and N. Bensaou, "A New Classical Arabic Soundex Algorithm," 2012.

[3] R. D. Erisandi, "Analisis dan Implementasi Algoritma Damerau Levenshtein Distance untuk Content Based Music Retrieval Analysis and Implementation Damerau Levenshtein Distance Algorithm for Content Based Music Retrieval," pp. 9-10, 2008.

[4] Z. Su, B.-R. Ahn, K.-Y. Eom, M.-K. Kang, J.-P. Kim and M.-K. Kim, "Plagiarism Detection Using the Levenshtein Distance and Smith-Waterman Algorithm," *2008 3rd International Conference on Innovative Computing Information and Control*, pp. 569-569, 2008.

[5] A. D. Ramadhan, "Metode Pencocokan String dengan Transliterasi Nama dalam Aksara Arab dan Latin," 2011.

[6] "Islamicity," HADI, 1995. [Online]. Available: https://www.islamicity.org/. [Accessed 15 June 2018].

[7] G. Desrianti, "Akurasi dalam Pencarian pada Search Engines," p. 3, 2011.

[8] RI and M. A. d. M. P&K, Pedoman Alih Aksara Arab ke Latin, 1987.

[9] M. Z. Fajrian, M. A. Bijaksana and E. Darwiyanto, "Analisis Sistem Pencarian Ayat Al-Quran menggunakan Algoritma Soundex berdasarkan Kemiripan Ucapan," 2017.

[10] M. Syahroni and R. Munir, "Pencocokan String Berdasarkan Kemiripan Ucapan (Phonetic String Matching) dalam Bahasa Inggris," pp. 7-13, 2005.

[11] N. UzZaman and M. Khan, "A double metaphone encoding for approximate name searching and matching in Bangla," 2005.