

Analisis Pencocokan Nama Arab Terjemahan Bahasa Indonesia Menggunakan *Soundex* dan *Levenshtein Distance*

Fauzan Ramadhan¹, Moch. Arif Bijaksana², Bambang Ari Wahyudi³

^{1,2,3}Fakultas Informatika, Universitas Telkom, Bandung

¹fzn.rmdhn@gmail.com, ²arifbijaksana@telkomuniversity.ac.id, ³bambangari@telkomuniversity.ac.id

Abstrak

Cara seseorang dalam mengeja nama orang lain seringkali berbeda dengan orang yang lain. Padahal nama yang akan dieja adalah nama dari seseorang yang sama. Seperti nama 'Aisyah', kadang ada yang mengejanya dengan kata 'Aisyah', 'Aisha', 'Aisah', dan lain-lain. Pencarian nama periwayat pada saat ini baru sampai tahap '*string matching*', sehingga ketika seseorang menggunakan ejaan nama yang berbeda, sistem tersebut tidak akan menampilkan hadits yang diriwayatkan oleh nama-nama yang dianggap mirip. Harapannya dengan sistem yang dibangun ini akan memperbaiki pengalaman pencarian nama dari periwayat hadits, sehingga sistem yang ada akan menampilkan hadits yang diriwayatkan oleh seseorang dimulai dari yang mempunyai kemiripan nama tertinggi sampai terendah dengan nama yang dicari. Penelitian ini menggunakan metode *Soundex*, lalu dilanjutkan dengan metode *Levenshtein*. Setelah itu, akan dilakukan penghitungan untuk menilai kinerja dari sistem ini menggunakan nilai *precision*, *recall*, *f-measure* dan akurasi. *Soundex* akan menentukan nama yang mempunyai kesamaan pengucapan suatu nama terhadap nama lain. Sedangkan *Levenshtein* akan memberikan nilai kemiripan dari nama yang sudah dipilih sebelumnya dengan nama yang dicari oleh pengguna. Nilai kinerja sistem didapat dari hasil yang diberikan dibandingkan dengan nilai yang ada pada *gold standard*. Dengan beberapa pengujian, sistem yang dibangun menggunakan *Soundex* dan *Levenshtein* ini mendapatkan nilai akurasi sebesar 99.95 persen.

Kata kunci : *Soundex*, *Levenshtein*, *precision*, *recall*, *f-measure*, akurasi

Abstract

The way someone spells someone else's name is often different from other people. Even though the name that will be spelled is the name of the same person. Like the name 'Aisha', sometimes there are who spell it with the words 'Aisyah', 'Aisha', 'Aisah', and others. The search for the name of the narrator at this time is only until the string matching stage, so when someone uses a different spelling of the name, the system will not display the hadith narrated by names that are considered similar. The hope with this system will improve the experience of searching the names of the hadith narrators, so that the existing system will display the hadith narrated by someone starting from the one having the highest to the lowest name with the name sought. This study used the *Soundex* method, then continued with the *Levenshtein* method. After that, it will be calculated to assess the performance of this system using *precision*, *recall*, *f-measure* and accuracy. *Soundex* will determine the name that has the same pronunciation as a name for another name. Whereas *Levenshtein* will give a similar value of the name that has been previously selected with the name that is searched by the user. The value of system performance is obtained from the results given compared to the values that are at the gold standard. From several tests, the system built using *Soundex* and *Levenshtein* has an accuracy value of 99.95 percent.

Keywords: *Soundex*, *Levenshtein*, *precision*, *recall*, *f-measure*, accuracy

1. Pendahuluan

Latar Belakang

Nama dalam kehidupan sehari-hari merupakan sesuatu yang sangat penting bagi siapapun atau apapun. Nama menurut Kamus Besar Bahasa Indonesia adalah kata untuk menyebut atau memanggil orang, tempat, binatang, dan sebagainya. Biasanya nama yang digunakan untuk seseorang merupakan nama-nama yang mempunyai arti bagus, contohnya seperti Muhammad yang artinya 'orang yang terpuji', Abdul Rohim yang artinya 'hamba yang penyayang', dll. Nama-nama yang dipilih juga tidak jarang diambil dari nama-nama sahabat nabi atau orang yang pernah meriwayatkan sebuah hadits, seperti 'Aisyah, Abu Dzar, dll. Selain itu, pencarian nama ini juga dilakukan untuk mencari suatu hadits yang diriwayatkan oleh seorang sahabat Nabi. Sehingga nantinya dapat membantu

mencari suatu hadits menggunakan pencarian terhadap nama periwayatnya. Hadits menurut bahasa artinya baru. Hadits juga berarti “sesuatu yang dibicarakan dan dinukil” dari sumber sebelumnya. Sedangkan menurut istilah ilmu hadits, hadits adalah apa yang disandarkan kepada Nabi Muhammad Shallallahu ‘alaihi wa sallam, baik berupa ucapan, perbuatan, penetapan, sifat atau sirah beliau, baik sebelum kenabian dan sesudahnya [2].

Cara seseorang dalam mengeja nama orang lain sering kali berbeda walaupun nama yang dimaksud adalah nama yang sama, dalam artian cara pengucapan dan nama yang dituju adalah nama dari seseorang. Penelitian ini dibuat untuk menganalisa nama yang dinilai mirip atau merupakan variasi nama dari seseorang bergantung pada kebiasaan seseorang dalam mengeja nama orang lain (dalam hal ini adalah nama dari periwayat hadits). Metode yang akan digunakan pada penelitian ini adalah menggunakan metode *Soundex*, yaitu algoritma pengkodean fonetik yang mengubah semua huruf setelah huruf yang pertama sampai huruf yang ke empat menjadi angka sesuai dengan tabel pengkodean dan *Levenshtein distance*, yaitu metode untuk mencari seberapa banyak proses penghapusan, penambahan dan penyisipan karakter (disebut sebagai pencarian jarak) antara nama satu dengan nama yang lain kemudian dicari persentase kemiripannya untuk dilakukan proses ranking. Nama yang dimaksud dalam hal ini adalah nama dari periwayat hadits. Hadits yang dipakai dalam tugas akhir ini adalah terjemahan dari hadits shahih Bukhori. Pada penelitian mengenai *name matching* yang ada sebelumnya, nama yang jadi *dataset* diujikan dengan beberapa metode dan salah satunya menggunakan metode yang ada pada *phonetic encoding* yaitu metode *Soundex*.

Topik dan Batasannya

Adapun permasalahan pada tugas akhir ini adalah untuk mencari nama yang diperkirakan sebagai nama yang mirip atau merupakan variasi nama dari nama yang dicari. Batasan pada tugas akhir ini yaitu inputan berupa nama yang ingin dicari kemiripannya atau variansinya, dataset yang digunakan adalah terjemahan Indonesia dari hadits shahih bukhori, keluarannya adalah hadits yang sudah diranking, lalu metode yang digunakan hanya menggunakan metode *Soundex* untuk melakukan pengklasifikasikan nama-nama yang mempunyai kode *phonetic* yang sama dan menggunakan metode *Levenshtein* untuk melakukan *ranking* terhadap kata-kata tersebut.

Tujuan

Adapun tujuan yang ingin dicapai dalam penelitian tugas akhir ini yaitu mengetahui hasil analisa bahwa kedua nama tersebut adalah nama yang sama atau serupa atau merupakan variasi nama dari nama yang dicari menggunakan metode *Soundex* dan mencari persentase kemiripannya dengan menggunakan metode *Levenshtein*. Lalu melakukan perankingan hadits berdasarkan nilai kemiripan yang didapat oleh kata nama pada hadits tersebut. Selain itu, nama yang sudah diperkirakan akan dinilai kualitasnya dengan nama yang sebenarnya dengan menggunakan *precision*, *recall* dan akurasi.

2. Studi Terkait

Hadits adalah segala sesuatu yang dinukil atau disandarkan kepada Nabi Muhammad *shallallahu 'alaihi wa sallam* meliputi perkataan, perbuatan dan penetapannya [2]. Hadits merupakan salah satu sumber dasar hukum yang digunakan oleh orang muslim dalam menentukan sebuah hukum setelah Al-Quran. Hal ini disampaikan dalam sabda Nabi Muhammad *shallallahu 'alaihi wa sallam* yang artinya “saya tinggalkan dua perkara yang kamu tidak akan tersesat apabila berpegang pada keduanya, yakni Kitabullah (Al-Quran) dan sunnah Nabi-nya (Hadits)” (H.R. Malik) [1]. Hadits disampaikan oleh Nabi Muhammad *shallallahu 'alaihi wa sallam* kepada para sahabatnya seperti saat beliau mengajarkan Al-Quran kepada para sahabatnya yaitu dengan menggunakan metode hafalan. Dalam perjalanan hadits, ada beberapa sahabat Nabi yang menulis hadits, seperti Abdullah bin 'Amr bin Al-'As, Abdullah bin Abbas dan Jabir bin Abdullah [8].

Name matching adalah salah satu inti dari berbagai macam aplikasi semisal penambangan teks, ekstraksi informasi, dan lain sebagainya. Pada kenyataannya, proses untuk mencocokkan nama itu bukan hal yang sepele, karena banyaknya variasi dari sebuah nama sehingga diharuskan untuk membuat sebuah teknik untuk mencocokkan nama tersebut dengan tepat [5]. *Soundex* adalah algoritma pengkodean fonetik yang mengubah semua huruf setelah huruf yang pertama sampai huruf yang ke empat menjadi angka sesuai dengan tabel pengkodean berdasarkan cara pengucapan dalam bahasa Inggris. Adapun algoritma *Soundex* bisa digambarkan dengan sebagai berikut [7] yaitu 1) hilangkan huruf vokal, meliputi a, i, u, e, o, y, h, dan w, 2) ganti huruf-huruf konsonan dengan aturan seperti pada tabel 6, 3) jika pada kata tersebut terdapat huruf yang ganda, maka huruf tersebut akan diperlakukan sebagai satu huruf, seperti kata Barro diberi kode B600 (B untuk huruf awal, 6 untuk huruf r yang pertama, r yang kedua diabaikan, 00 adalah tambahan untuk kode *phonetic* yang kurang dari 4 karakter) dan 4) jika ada string yang memiliki dua huruf dengan kode *phonetic* yang sama, maka huruf yang kedua diabaikan, seperti pada kata Jackson, kode *phonetic*-nya adalah J250 (J untuk huruf awal, 2 untuk huruf c, huruf k diabaikan, huruf s diabaikan, 5 untuk

huruf n, dan 0 adalah tambahan untuk kode *phonetic* yang kurang dari 4 karakter)[6].

Tabel 1. Tabel kode *phonetic*

Huruf	kode
b, f, p, v	1
c, g, j, k, q, s, x, z	2
d, t	3
l	4
m, n	5
r	1

Levenshtein distance adalah algoritma yang dapat menghitung jumlah terkecil dari operasi *edit*, seperti penambahan huruf, penghapusan huruf, dan penggantian huruf yang diperlukan untuk mengubah suatu *string* ke *string* lain. Pada dasarnya, setiap operasi *edit* yang dilakukan, akan dihitung satu jarak. Semakin banyak perubahan yang diperlukan, maka semakin banyak nilai jarak yang didapat. Setelah didapat jumlah jarak dari operasi *edit* tersebut, maka nilai jarak akan digunakan untuk menentukan nilai kesamaan antara *string* satu ke *string* yang lain dengan rumus sebagai berikut : [5].

$$sim_{ld}(s_1, s_2) = 1.0 - \frac{dist_{ld}(s_1, s_2)}{\max(|s_1|, |s_2|)} \quad (1)$$

Dalam proses pencocokan kata nama, ada sebuah perhitungan untuk mencari sudah sebaik apa sistem yang dibangun. Perhitungan tersebut didasarkan kepada nilai *precision*, *recall*, *f-measure* dan akurasi. Untuk mencari keempat nilai tersebut, diperlukan 4 faktor, yaitu *true positif* (TP), *true negatif* (TN), *false positif* (FP), dan *false negatif* (FN). *True positif* didapat dari jumlah kata yang diprediksi adalah sebuah nama dan pada kenyataannya adalah nama tersebut. *False positif* didapat dari jumlah kata yang diprediksi adalah sebuah nama, namun kenyataannya bukan nama tersebut. *False negatif* didapat dari jumlah kata yang diprediksi bukan kata suatu nama, namun kenyataannya kata suatu nama itu. *True negatif* didapat dari jumlah kata yang diprediksi bukan kata suatu nama dan pada kenyataannya adalah memang bukan kata tersebut. *Precision* itu menunjukkan berapa banyak kata nama yang tertebak dari seluruh kata nama yang ditebak, sehingga untuk menghitung *precision* adalah sebagai berikut [3].

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

Recall itu menunjukkan berapa banyak kata nama yang tertebak dari seluruh kata nama. Untuk menghitung *recall* adalah sebagai berikut.

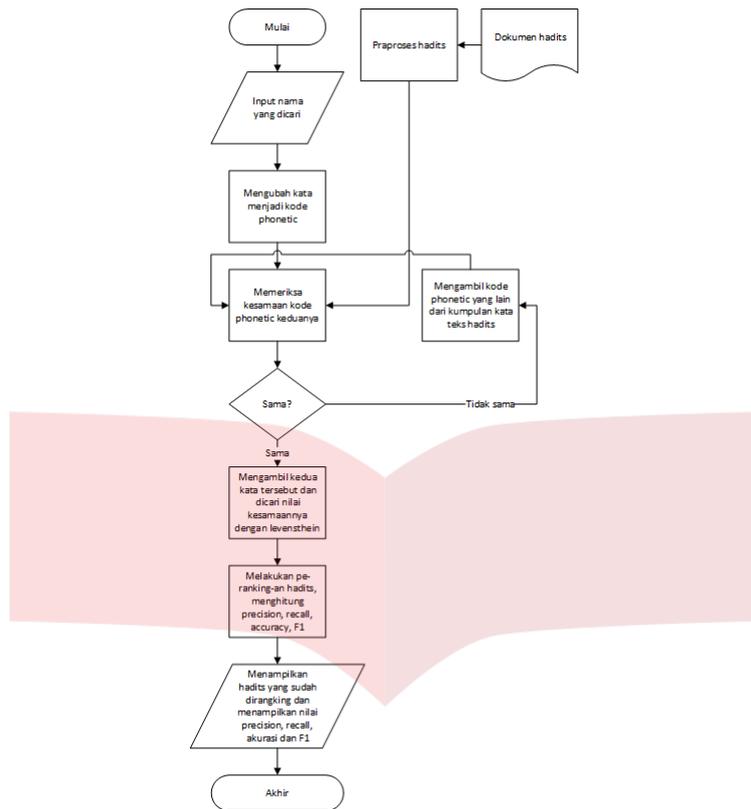
$$Recall = \frac{TP}{TP + FN} \quad (3)$$

F-measure dicari dengan menggunakan nilai dari *precision* dan *recall* yang sudah didapat sehingga didapatkan nilai rata-rata harmoniknya, yaitu sebagai berikut $F\text{-measure} = 2 * \frac{Precision * Recall}{Precision + Recall}$. Nilai akurasi dari sistem dapat dicari dengan menggunakan 4 faktor yang telah disampaikan sebelumnya, yaitu sebagai berikut $Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \cdot [4]$

3. Sistem yang Dibangun

Pada gambar 1, digambarkan mengenai pembangunan sistem yang menggunakan algoritma *Soundex*, lalu dilanjutkan dengan penerapan algoritma *Levenshtein distance* untuk dilakukan perankingan hadits pada nama yang ingin dicari. Data yang akan digunakan sudah disimpan dalam bentuk *file* yang tiap-tiap *file*-nya berisi satu hadits yang diambil dari hadits Bukhori.

Pada awalnya, sistem akan meminta inputan dari *user* yang merupakan nama orang. Setelah itu, sistem akan membuat kode *Soundex*-nya seperti pada gambar 2. Di sisi lain, sistem akan mengambil hadits dari tiap-tiap *file* untuk dilakukan pencarian nama berdasarkan inputan *user*. Ketika sistem sudah mengambil hadits, sistem akan melakukan pemotongan hadits tersebut menjadi kumpulan kata. Lalu, sistem akan menghapus tanda baca dari tiap-tiap kata yang ada, sehingga dihasilkan kata yang hanya terdiri dari kumpulan huruf saja. Setelah itu, sistem akan memilih kata-kata yang mempunyai salah satu ciri dari kata nama, yaitu adanya huruf kapital di awal katanya. Kata-kata yang terpilih akan dibuatkan kode *Soundex*-nya dan disimpan sementara dalam sebuah penyimpanan seperti pada gambar 3. Satu persatu kode *Soundex* dari kata-kata yang terpilih akan dicocokkan dengan kode *Soundex* dari kata inputan. Jika kedua kata tersebut mempunyai kode *Soundex* yang sama, maka kode *Soundex* dari kata yang terpilih dari hadits itu akan disimpan beserta dengan katanya. Setelah semua kode *Soundex* dari kata-kata



Gambar 1. Sistem yang dibuat

yang terpilih pada hadits itu diperiksa kesamaannya dengan kode *Soundex* kata inputan, maka kata-kata terpilih itu akan dicari nilai kesamaannya dengan kata inputan menggunakan algoritma *levesthein distance*. Dalam mencari nilai *similarity*, digunakanlah metode *Levenshtein*. Sebelum menghitung *similarity*, perlu diketahui perbedaan huruf antara kata pertama dan kata kedua. Perbedaan huruf ini biasa disebut *distance* atau jarak. Pencarian jarak bisa didapat dari melihat perbedaan hurufnya atau bisa dengan menggunakan teknik penelusuran dengan matriks. Lihat gambar 4.

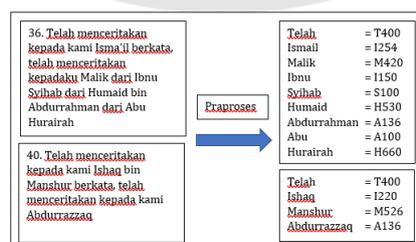
Setelah didapat nilai *distance*, maka masukkan nilai tersebut ke rumus sebagai berikut:

$$Levenshtein(S_1, S_2) = 1.0 - \frac{dist(S_1, S_2)}{\max(|S_1|, |S_2|)} \tag{4}$$

Kata-kata terpilih yang sudah dicari nilai kesamaannya dengan kata inputan, akan dilakukan perankingan ber-

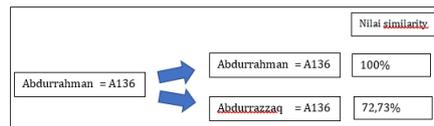


Gambar 2. Proses perubahan kata inputan ke kode *Soundex*



Gambar 3. Proses perubahan 2 hadits menjadi kode *Soundex*

dasarkan nilainya. Semakin besar nilai kesamaannya, maka akan disimpan di peringkat yang lebih tinggi, begitu



Gambar 4. Hasil *similarity* kata yang punya kode *Soundex* sama dengan kata inputan

Tabel 2. Kata yang diberikan kepada responden untuk menjadi kata uji sistem

Nama Arab yang diujikan	Hasil dari Responden	Kata yang Diuji	Kode Soundex	Nama Arab yang diujikan	Hasil dari Responden	Kata yang Diuji	Kode Soundex
بِلَال	Bilaal Bilal	Bilaal	B440	عَائِشَة	Aaisyata Aisyah	Aaisyata	A230
عَبْدُ الرَّحْمَنِ	Abdurrahman Abdurrohmani	Abdurrohmani	A136	عَبْدُ اللَّهِ	Abdulloh Abdullaah	Abdulloh	A134
قَتَادَة	Qotaadah Qotadah	Qotaadah	Q330	عُمَرُ	Umar Umaro	Umar	U560
أَبُو	Abuu Abu	Abuu	A100	مُحَمَّدُ	Muhammadu Muhammad	Muhammadu	M530
عَبَّاس	Abbaasin A'bbaas	Abbaasin	A125	شُعْبَة	Syu'bah Syubah	Syu'bah	S150

pula sebaliknya. Disaat yang bersamaan, sistem melakukan perhitungan *precision*, *recall*, *f-measure* dan akurasi dari proses pencarian nilai kesamaan dan kemunculan kata yang serupa dengan kata yang ada pada *gold standard*. Setelah selesai perangkingan, maka sistem akan menampilkan hadits yang mempunyai kata tersebut. Hasil yang ditampilkan oleh sistem ini adalah perangkingan hadits yang mempunyai kata dengan nilai kesamaan lebih besar dengan nilai kesamaan kata inputan dari *user*.

4. Evaluasi

4.1 Hasil Pengujian

Pengujian dilakukan berdasarkan *query* yang didapat dari cara baca responden terhadap nama bahasa arab yang disajikan dengan tulisan arab. Responden menginputkan caranya ketika membaca nama tersebut, seperti pada tabel 2. Setelah didapat beberapa hasilnya, maka dicarilah nilai kesamaannya menggunakan metode *Soundex-Levenshtein*. Setelah didapat nilai kesamaannya, program mencari nama dengan tingkat kemiripan yang paling besar untuk secara otomatis dicarikan jumlah kemunculan kata tersebut dari 100 hadits yang disediakan. Pencarian jumlah kemunculan itu didapat dari *gold standard* yang telah dibuat dan sudah diverifikasi oleh salah seorang dosen Prodi Ilmu Hadits, Fakultas Ushuluddin, UIN Sunan Kalijaga yang bernama Dr. Agus Suyadi, M.Ag. Pengujian dilanjutkan untuk mendapatkan nilai *precision*, *recall*, *f-measure* dan akurasi dari sistem yang sudah dibangun. Tabel 3 dan tabel 4 adalah beberapa hasil pengujian yang telah dilakukan. Pada tabel 5, diperlihatkan hasil sistem terhadap perangkingan dari 100 hadits yang telah dicari kata namanya, dengan kode *Soundex* yang sama dengan kode *Soundex* dari kata inputan.

Hasil yang dicapai oleh sistem ini beragam karena *query* yang dimaksud disini adalah hasil dari responden. Terdapat salah satu *query* yang ketika diujikan ke sistem, tidak ditemukan hasilnya, yaitu kata Abbaasin. Kata ini ketika dijadikan kode *Soundex* dan dilakukan pencarian kesamaan kode *Soundex* dari setiap kata pada hadits, tidak ditemukan kata dengan kode *Soundex* yang sama. Sedangkan jika dilihat dari kuisioner yang dipakai, kata Abbaasin ini merupakan cara baca seorang responden terhadap kata Abbas. Selain itu, terdapat salah satu hasil dari kata yang mempunyai kode *Soundex* yang mirip dengan *query* namun secara artian kata merupakan kata yang berbeda yaitu kata Apa. Kata Apa menjadi salah satu dari hasil kata yang memiliki kode *Soundex* yang sama dengan kata Abu. Abu dan Apa memiliki kode *Soundex* yang sama, yaitu A100.

Tabel 3. Hasil pengujian terhadap beberapa *query* dari responden

No.	Query	Kode Soundex	Gold Standard (GS)	Jumlah Gold Standard (GS)	Kata yang Mirip Soundex-nya	Kode Soundex	Kemiripan
1	Bilaal	B440	Bilal	4	Bilal	B440	83.34%
2	Abdurrohmani	A136	Abdurrahman	11	Abdurrahman	A136	83.34%
					Abdurrazzaq	A136	58.33%
3	Qotaadah	Q330	Qotadah	9	Qotadah	Q330	87.50%
4	Abuu	A100	Abu	145	Abu	A100	75.00%
					Auf	A100	50.00%
					Ayyub	A100	40.00%
					Apa	A100	25.00%
5	Abbaasin	A125	-	-	-	-	-
6	Aaisyata	A230	Asad	1	Asad	A230	62.50%
					Aqadi	A230	50.00%
7	Abdulloh	A134	Abdullah	81	Abdullah	A134	87.50%
					Abdul	A134	62.50%
8	Umar	U560	Umar	21	Umar	U560	100.00%
					Umarah	U560	66.67%
9	Muhammadu	M530	Muhammad	52	Muhammad	M530	88.89%
					Mahmud	M530	55.56%
10	Syu'bah	S150	Syubah	23	Syubah	S150	85.71%
					Syaibah	S151	71.42%
					Shabbah	S152	57.55%
					Syihab	S153	42.84%
					Syuaib	S154	42.85%
					Siapa	S155	42.85%
Rata-rata							62.95%

4.2 Analisis Hasil Pengujian

Hasil pengujian ini menunjukkan adanya keberagaman hasil dengan nilai kesamaan yang beragam pula. Pada sistem ini tidak ada penggunaan *threshold*, karena selain ingin mencari nama yang secara penulisan sama dengan apa yang diinputkan oleh *user*, sistem ini juga menampilkan kata yang sekiranya mempunyai kode *Soundex* yang sama dengan kode *Soundex* dari inputan *user*. Sebagai contoh, kata Aaisyata ini ternyata membuat sistem mencari kata yang mempunyai kode *Soundex* sama dengan Aaisyata (yaitu A230), kata tersebut adalah kata Asad. Kedua kata ini mempunyai kode *Soundex* yang sama, sehingga sistem menganggap bahwa kata Aaisyata dan Asad adalah kata nama yang sama. Contoh lain adalah kata Syu'bah ternyata memberikan hasil pencarian terhadap kata Syubah, Syaibah, Shabbah, Syihab, Syuaib, Siapa, Shuhaib. Padahal jika kita telusuri dari kata-kata pada *gold standard*, maka ditemukan kata Syihab, Syuaib, Syubah, Shabbah, dan Shuhaib itu adalah nama dari orang yang berbeda. Bahkan kata Siapa yang bukan merupakan kata nama, dianggap sebagai kata nama karena mempunyai kode *Soundex* yang sama dengan kata inputan (Syu'bah) dan diawali oleh huruf kapital.

Perangkingan yang dilakukan sistem mengacu pada besaran nilai kesamaan antara kata inputan (*query*) dengan kata nama yang ada pada tiap hadits. Perangkingan ini dimaksudkan sebagai saran atas nama yang dicari oleh *user*, sehingga *user* tidak harus mencari yang nama yang paling mirip secara manual. Namun, pada prakteknya bisa saja nama yang diinginkan oleh *user* adalah nama yang berada pada peringkat terakhir dari hasil yang disajikan oleh sistem. Dari 10 *query* yang didapat melalui responden yang berbeda, sistem ini menghasilkan nilai rata-rata akurasi sebesar 92.22 persen, dengan nilai *precision* rata-rata adalah 75.56 persen, *recall* rata-rata adalah 100 persen dan *F-measure* rata-rata sebesar 80 persen. Nilai akurasi rata-rata sistem ini yang mencapai angka 92.22 persen menunjukkan bahwa sistem ini sudah cukup baik, hanya saja mungkin butuh perbaikan lagi sehingga kasus Abbasin tidak akan dianggap tidak ditemukan.

Tabel 4. Hasil *precision*(P), *recall*(R), *f-measure*(F) dan akurasi(A) terhadap *query* dari responden

No.	Query	Kode Soundex	TP	FP	FN	TN	P	R	F	A
1	Bilaal	B440	4	0	0	12680	100.00%	100.00%	100.00%	100.00%
2	Abdurrohmani	A136	11	1	0	12672	91.67%	100.00%	95.65%	99.99%
3	Qotaadah	Q330	9	0	0	12675	100.00%	100.00%	100.00%	100.00%
4	Abuu	A100	145	12	0	12527	92.36%	100.00%	96.03%	99.91%
5	Abbaasin	A125	0	0	0	12684	0.00%	0.00%	0.00%	100.00%
6	Aaisyata	A230	1	2	0	12681	33.33%	100.00%	50.00%	99.98%
7	Abdulloh	A134	81	14	0	12589	85.26%	100.00%	92.05%	99.89%
8	Umar	U560	21	2	0	12661	91.30%	100.00%	95.45%	99.98%
9	Muhammadu	M530	52	1	0	12631	98.11%	100.00%	99.05%	99.99%
10	Syu'bah	S150	22	1	33	12628	40.00%	95.65%	56.41%	99.73%
Rata-rata							73.20%	89.57%	78.46%	99.95%

Tabel 5. Hasil yang ditampilkan sistem dalam bentuk *rangking*

No.	Query	Kode Soundex	Gold Standard (GS)	Jumlah Kata GS	Kata yang Mirip Soundex-nya	Kode Soundex	Kemiripan	Nomor Hadits	Rank
1	Bilaal	B440	Bilal	4	Bilal	B440	83.34%	8	1
								89	2
								96	3
								96	4
2	Abdurrohmani	A136	Abdurrahman	11	Abdurrahman	A136	83.34%	3	1
								18	2
								18	3
								30	4
								36	5
								65	6
								68	7
								69	8
								89	9
								99	10
					99	11			
					Abdurrazzaq	A136	58.33%	40	12
3	Qotaadah	Q330	Qotadah	9	Qotadah	Q330	87.50%	12	1
								12	2
								14	3
								20	4
								42	5
								42	6
								63	7
								63	8
					79	9			

5. Kesimpulan

Kesimpulan yang bisa diambil dari hasil analisis adalah program ini bisa dijadikan *tools* dalam mencari sebuah nama dengan menggunakan inputan nama yang dianggap variasi dari suatu nama. Nilai akurasi menunjukkan 92 persen hasil yang ditampilkan adalah nama yang benar sesuai dengan nama yang diinputkan. Namun, tidak berarti nilai ini menjadikan keseluruhan sistem menjadi bagus, karena data uji yang disajikan hanya beberapa saja. Saran untuk pengembangan selanjutnya adalah perlu adanya pembuatan korpus nama periwayat hadits untuk meningkatkan akurasi dari program ini. Selain itu, perlu adanya penambahan data sehingga dapat mencari nama lebih banyak lagi. Dalam program ini, data yang disajikan hanya 100 hadits dari banyaknya hadits Shohih Bukhori, dan 6 kitab hadits yang lain.

Daftar Pustaka

- [1] R. Al Hana. Studi hadits, 2011.
- [2] S. M. Al-Qaththan. *Pengantar Studi Ilmu Hadits*. Pustaka Al Kautsar, 2012.
- [3] M. Alifikri. *Indonesian Name Matching Using Machine Learning Supervised Approach*. Telkom University, 2017.
- [4] S. P. Benny, S. Vasavi, and P. Anupriya. Hadoop framework for entity resolution within high velocity streams. *Procedia Computer Science*, 85:550–557, 2016.
- [5] P. Christen. A comparison of personal name matching: Techniques and practical issues. In *Data Mining Workshops, 2006. ICDM Workshops 2006. Sixth IEEE International Conference on*, pages 290–294. IEEE, 2006.
- [6] Z. Fan. Matching character variables by sound: a closer look at soundex function and sounds-like operator. *SAS® Users Group Institute, Paper*, pages 072–29, 2004.
- [7] D. Pinto, D. Vilarino, Y. Alemán, H. Gómez, and N. Loya. The soundex phonetic algorithm revisited for sms-based information retrieval. In *II Spanish Conference on Information Retrieval CERI*, 2012.
- [8] H. A. Sanaky. Hadits pada masa nabi kajian hadits dan perbedaannya dengan as-sunnah, al-khabar, atsar. 1999.