Jurnal Teknovasi
Volume 04, Nomor 02, 2017, 69 – 78
ISSN : 2540-8389

PERBAIKAN METODE JARO–WINKLER DISTANCE UNTUK APPROXIMATE STRING SEARCH MENGGUNAKAN DATA TERINDEKS APLIKASI MULTI USER

Friendly[1*]
[1]Politeknik Negeri Medan
*Email: friendly@polmed.ac.id

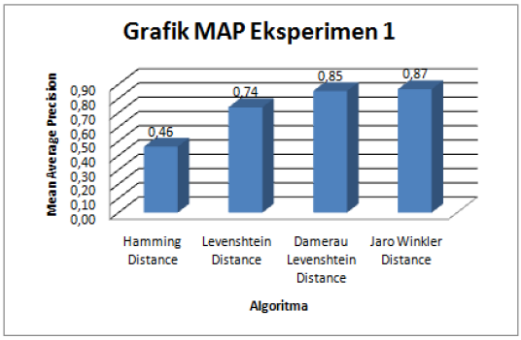
ABSTRAK

Metode jaro winkler digunakan dalam pembanding lexicographic untuk mencari kata yang sesuai atau mendekati kata yang dicari. Dari waktu yang diperlukan untuk menghasilkan kedekatan kata, adalah suatu *overhead* bila dilakukan secara berulang oleh pengguna yang berbeda. Apabila metode ini diterapkan pada aplikasi berbasis *multi user*, maka pengguna akan merasakan waktu akses yang bertambah lama dibandingkan dengan pencarian tepat suatu kata. Untuk mengatasi terjadinya pencarian dan pemrosesan data secara berulang, pada metode Jaro-winkler diterapkan pengindeksan data hasil pencarian yang telah dihasilkan. Dari pengujian yang dilaksanakan, tampak bahwa proses pencarian kedekatan string untuk pencarian dengan kata kunci yang sama pada proses pencarian berikutnya lebih cepat antara 90-92% dibandingkan dengan pencarian dengan menggunakan metode Jaro Winkler saja. Dengan semangkin berkembangnya indeks data pencarian, akan sangat mempercepat proses pencarian terhadap data-data string yang memiliki kedekatan.

Kata-kata kunci: *jaro winkler*, kedekatan kata, pencarian kata, data indek

PENDAHULUAN

Metode pencarian terhadap kata telah lama ditemukan dan dikembangkan dengan berbagai metode yang diadaptasi dari beberapa pendekatan statistik dan metode fuzzy. Metode-metode tersebut diataranya adalah metode: Hamming Distance, Jaccard Distance, Jaro Distance, Jaro-Winkler Distance, Levenshtein Distance, Overlap Coefficient, Ratcliff-Obershelp Similarity, Sorensen-Dice Distance, Tanimoto Coefficient dan lain sebagainya. Metode tersebut digunakan dalam pembanding *lexicographic* untuk mencari kata yang sesuai atau mendekati kata yang dicari . (Navarro, 2001)
Berdasarkan penelitian yang dilakukan Y. Rochmawati and R. Kusumaningrum (2016), metode Jaro-winkler memberikan hasil yang sangat memuaskan dengan nilai *Mean Average Precision* (MAP) 0.87. Pengujian dilakukan untuk menentukan kesalahan dalam pengetikan. Hasil pengujian atas beberapa metode pendekatan string ditunjukkan pada Gambar 1.



Gambar 1. Perbandingan Penggunaan Penggunaan Metode Pendekatan String (Y. Rochmawati and R. Kusumaningrum 2016)

Jurnal Teknovasi
Volume 04, Nomor 02, 2017, 69 – 78
ISSN : 2540-8389

Metode Jaro-winkler umumnya digunakan untuk mencari kedekatan sejumlah data yang homogen K. Dressler and A.-C. N. Ngomo (2017). Dengan tingkat kesuksesan dalam mendeteksi kesalahan kata serta dalam mengetahui kedekatan data pada pengujian yang dilakukan sebelumnya, tentunya. efektifitas metode ini cukup baik. Pada penelitian K. Dressler and A.-C. N. Ngomo, diperoleh pengukuran tingkat efektifitas dari metode Jaro-winkler seperti ditunjukkan pada Tabel 2.

Tabel 2. Hasil Pengujian Metode Jaro-winkler [3]

DBpedia Class	Size	OA(0.8)	OA(0.9)	OA(0.95)
Actors	9509	15.07	10.13	6.38
Architect	3544	5.58	5.48	2.32
Criminal	5291	11.54	7.77	4.52

Hasil yang diperoleh pada Tabel 2 menggunakan beberapa jenis data dengan jumlah yang bervariasi. Waktu yang diperlukan dalam tiap pengujian cukup besar yakni antara 2,32-15,07 detik. Pengujian yang dilakukan umumnya dilakukan terhadap data-data yang hanya diakses oleh satu pengguna dan tidak digunakan oleh beberapa pengguna sekaligus. Bila pencarian string dengan metode Jaro-winkler digunakan untuk melakukan pencarian data pada aplikasi berbasis web, maka akan muncul beban terhadap penggunaan sumber daya *server*, terutama pada pencarian kata yang sama secara berulang. Hal ini dapat meningkatkan waktu yang diperlukan oleh server untuk dapat menampilkan hasil penelusuran.

METODE PENELITIAN

Metode Approximate String Search

Approximate string search atau pencarian kedekatan kata merupakan suatu metode pencarian terhadap beberapa kata yang mengijinkan munculnya error. Tujuan utama dari pencarian ini adalah untuk melakukan pencarian pada kata dimana terdapat kesalahan penulisan yang tidak disengaja maupun penulisan kata-kata yang disesuaikan dengan cara pelafalan pengguna dan tidak tepat dengan kata yang dimaksud. Pada pencarian kedekatan kata ini, pencarian string dilakukan secara samar yaitu kata yang dipasangkan memiliki kemiripan namun keduanya memiliki susunan karakter yang berbeda (mungkin jumlah atau urutannya) tetapi kata tersebut memiliki kemiripan baik kemiripan tekstual/penulisan atau kemiripan ucapan. Kesesuaian kata berdasarkan kemiripan tekstual/penulisan meliputi jumlah karakter, susunan karakter dalam dokumen ini disebut sebagai *approximate string matching*. Metode pencarian kesesuain kata berdasarkan kemiripan ucapan dari segi pengucapan disebut sebagai *phonetic string matching*.
Berikut ini contoh dari *approximate string matching* dan *phonetic string matching*: (a) kemarin dengan kemarau, memiliki jumlah karakter yang sama tetapi ada karakter yang berbeda. Jika perbedaan karakter ini dapat ditoleransi sebagai sebuah kesalahan penulisan maka dua string tersebut dikatakan cocok. (b) basa dengan basah dari tulisan berbeda tetapi dalam pengucapannya mirip sehingga dua string tersebut dianggap cocok. Contoh yang lain adalah basa, dengan bassa, bassah, baasa, bahsah.
Selain pada penelusuran kata seperti diatas, *approximate string search* digunakan juga dalam mencari urutan nucleotide pada sejumlah data DNA, pengecekan ejaan dan filter *spam* pada surat elektronik.
Kedekatan terhadap data pencarian dan data tempat pencarian diukur dengan melakukan perhitungan jumlah perubahan yang dilakukan terhadap kata dengan memecah dan menyusun kembali kata tersebut sehingga mendapatkan kata yang tepat. Nilai ini disebut dengan jarak perubahan.
Operasi sederhana yang dilakukan adalah sebagai berikut:

- Penyisipan : buh → buah
- Penghapusan : buah → bah

- Substitusi : buah → buas
- Transposisi : bias → bisa

Ketiga operasi sederhana ini dapat digeneraslisasi kembali dengan menambahkan karakter NULL bila dilakukan proses penyisipan dan penghapusan sebagai berikut:

- Penyisipan : bu*h → buah
- Penghapusan : buah → b*ah
- Substitusi : buah → buas

Dengan menggunakan metode perubahan susunan karakter seperti diatas, beberapa metode kedekatan kata menerapkan nilai-nilai acuan yang berbeda untuk masing-masing operasi. Metode-metode yang dikembangkan untuk mencari kedekatan kata diantaranya:

- Metode Levenshtein
  Pada metode ini, proses yang dilakukan terhadap kata pencarian dan data yang dicari hanya proses: penyisipan, penghapusan dan substitusi. Pada algoritma ini kedekatan kata ditentukan dengan semakin kecilnya nilai atau jumlah perubahan proses penyisipan, penghapusan dan substitusi yang diperlukan untuk mencocokkan kata pada data dasar terhadap kata yang dicari (P. E. Black 2017).
- Metode Damerau–Levenshtein
  Merupakan pengembangan dari metode Levenshtein yang mengijinkan dilakukannya proses transposisi 2 buah karakter yang berdekatan antara 2 kata yang memiliki jumlah karakter yang sama untuk merubah kata pada data dasar untuk menjadi kata yang dicari[]. Metode Damerau-Levenshtein digunakan pada aplikasi pengecekan pengejaan, pada pengecekan nama di perusahaan export amerika I. T. Administration (2017) dan lain sebagainya.
- Metode *Longest Common Subsequence* (LCS)
  Merupakan suatu metode pencarian kata dengan kesesuaian yang hanya mengizinkan penyisipan dan penghapusan (P. E. Black 2017).
- Metode Hamming
  Metode ini hanya mengijinkan dilakukannya pertukaran atau substitusi kata dan hanya pada kata yang memiliki panjang karakter yang sama (L. BOYTSOV 2017).
- Metode Jaro-winkler
  Metode Jaro-winkler dilakukan dengan mengukur total persentasi dari kata yang sesuai dan transposisi untuk setiap kumpulan kata atau data (W. E. Winkler 2006).

Metode Jaro-Winkler

Metode ini dikembangkan dari metode Jaro *Distance Metric* yaitu sebuah metode yang digunakan untuk mengukur kesamaan antara dua kata, biasanya metode ini digunakan di dalam pendeteksian duplikat. Semakin tinggi nilai Jaro-winkler untuk dua kata maka kedua kata tersebut semakin sesuai.
Dasar dari algoritma ini memiliki tiga bagian yakni:

1. Menghitung panjang kata.
2. Menemukan jumlah karakter yang sama di dalam dua kata.
3. Menemukan jumlah transposisi.

Metode Jaro-winkler diturunkan dari metode [1]. Untuk menentukan jarak/ nilai kesamaan (dj) antara dua buah kata s1 dan s2, metode jaro menggunakan persamaan sebagai berikut:

$$d_j = \begin{cases} 0 \\ \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right) \end{cases} \quad \text{.....(1)}$$

Dimana:

m = jumlah karakter yang sama yang berada di posisi yang sama
$|s_1|$ = panjang karakter pada kata pertama
$|s_2|$ = panjang karakter pada kata kedua
t = ½jumlah karakter yang bertukar posisi diantara kedua kata

Jurnal Teknovasi
Volume 04, Nomor 02, 2017, 69 – 78
ISSN : 2540-8389

Nilai dj akan menjadi 0 bila m=0. Notasi $|s_1|$ dan $|s_2|$ merupakan panjang karakter dari kata s1 dan s2. Notasi m merupakan jumlah karakter yang benar pada urutan yang benar dengan jarak antara karakter tidak lebih dari 1 karakter. Misalkan 2 buah kata yang akan dikomparasi adalah **dekat** dan **tekad**, nilai m untuk kedua kata tersebut adalah 3 yakni untuk karakter **e,k,a**. Walaupun karakter **d** dan **t** sama-sama ada pada kata tersebut, namun jarak antara keduanya lebih dari 1 karakter Nilai t merupakan setengah dari jumlah karakter yang bertransposisi. Untuk kada **dekat** dan **tekad** memiliki 2 karakter yang bertransposisi, sehingga nilai t=1. Untuk perhitungan nilai dj untuk kata **dekat** dan **tekad** adalah sebagai berikut:

$$dj = \frac{1}{3}\left(\frac{c}{|s_1|} + \frac{c}{|s_2|} + \frac{c - t}{c}\right)$$

$$= \left(\frac{1}{3} + \frac{3}{5} + \frac{3}{5}\right) = \frac{3-1}{3} = 0.622$$

Nilai dj akan dianggap benar bila sama dan tidak melebihi batas yang dinyatakan dalam persamaan 2 berikut.

$$\text{Jarak max} = \left\lfloor \frac{\max(|s_1|, |s_2|)}{2} \right\rfloor - 1. \qquad \text{.....(2)}$$

Untuk kata **tekad** dan **dekat** jarak yang menjadi batas adalah setengah dari panjang maksimum karakter di antara kedua kata dikurangi dengan 1. Dari persamaan 2 diatas, nilai dari jarak maksimum adalah (5/2)-1=1. Sehingga kata **tekad** dan **dekat** dapat dikatakan memiliki kedekatan.
Pada metode Jaro-winkler, jarak antara kata ditunjukkan dengan persamaan jarak antara kata ditunjukkan dengan persamaan 3 berikut:

$$d_w = d_j + (\ell p(1 - d_j)) \qquad \text{.....(3)}$$

Pada metode Jaro-winkler *prefix scale* (p) merupakan nilai konstanta yang diberikan untuk memberikan penyesuaian yang mana nilai yang digunakan umumnya adalah 0.1. Nilai p digunakan ntuk memberikan tingkatan nilai yang lebih pada kata yang memiliki kesesuaian yang sama diawal proses pencocokan kata. Notasi ℓ merupakan panjang karakter yang sama di awal pencocokan kata. Untuk kata **tekad** dan **dekat**, nilai ℓ adalah 0 sehingga nilai d_w dan nilai d_j adalah sama.

Perancangan Metode Jaro Winkler Dengan Data Terindeks

1. Parameter Pengukuran dan Pengamatan

Parameter yang digunakan dalam pencarian data adalah informasi yang umum digunakan dan dalam bentuk string yakni informasi nama. Pengukuran hasil pengujian ditetapkan dalam bentuk lama waktu yang diperlukan sistem yang menerapkan metode Jaro-winkler dalam proses pencarian kata secara bersama oleh beberapa responden. Lama waktu akses dibandingkan untuk menentukan apakah metode yang diajukan dalam penelitian ini lebih baik dari pada metode sebelumnya. Pengukuran terhadap keseluruhan nilai menggunakan *Mean Average Precision* (MAP). Pada penelitian ini akan digunakan 2 buah tabel dengan jumlah rekod 16.926 dan 69.145. Jumlah data yang berbeda digunakan untuk menganalisa kinerja metode dengan jumlah data yang cukup besar. Data yang akan digunakan untuk pencarian adalah data nama.

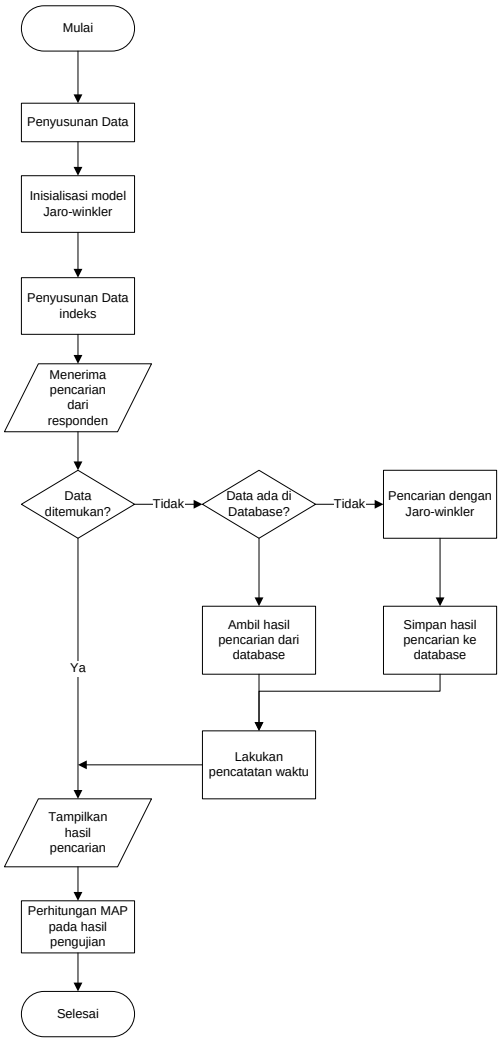
2. Model Penelitian

Model yang akan dirancang untuk menjamin terlaksananya pengujian pada penelitian ini adalah sebagai berikut:

a) Penyusunan model untuk pengelolaan dan data pada database. Model digunakan untuk melakukan proses pengambilan dan penyusunan data. Pengelolaan data tidak akan berkaitan dengan metode yang digunakan secara langsung.
b) Penyusunan model Jaro-winkler dengan dan tanpa data terindeks untuk dilakukan pengujian
c) Pengujian model pencarian kata dengan menggunakan beberapa kata.
d) Pengolahan hasil responden untuk memperoleh waktu pengolahan dan pemrosesan data yang diperlukan untuk melakukan pencarian data dengan menggunakan metode Jaro-winkler dengan dan tanpa data terindeks.

3. Rancangan Penelitian

Untuk perancangan pengujian dan penelitian, dilakukan dengan mengikuti flowchart sebagai berikut:



Gambar 3. Flowchart rancangan penelitian

4. Pengujian Sistem

Pengujian dilakukan dengan menggunakan data nama. Terdapat 2 pengujian yakni pengujian dengan menggunakan metode Jaro Winkler dan metode Jaro Winkler dengan data terindeks. Sejumlah kata yang digunakan untuk menguji metode ini adalah sbb:

Tabel 1. Daftar Nama Uji

ade	nia	ani	ita	adi
rudi	mila	wati	andi	budi
indah	benar	yonna	diana	anita
selena	vanesa	kirana	kurasa	melati
katerin	vanessa	permata	gunawan	kamelia
sulastri	sutrisno	rowandha	benedith	meredith
sumarjono	sukmawati	heriyanto	stephanie	alexander
evangelina	diah putri	kurniawati	hardiyanto	para digma
sudharmanto	cokrominoto	bartolomeus	sri hartati	grande ansi

Kata-kata ini dipilih secara acak. Penyusunan kata berdasarkan jumlah karakter di setiap kata pencarian. Jumlah karakter setiap kata adalah 3,4,5,6,7,8,9,10 dan 11. Untuk setiap panjang karakter kata yang dicari, digunakan 5 buah kata yang berbeda.
Pengujian dilakukan terhadap 2 buah kumpulan rekod nama dengan jumlah yang berbeda yakni 16.926 dan 69.145 buah rekod. Pengujian dilakukan untuk mengetahui pengaruh dari jumlah data terhadap lama waktu yang diperlukan dalam menjalankan pencarian terhadap data. Perbandingan dilakukan dengan metode:

1. Pencarian dengan kata tepat
2. Pencarian kata terdekat dengan menggunakan metode Jaro Winkler, dan
3. Pencarian kata terdekat dengan menggunakan metode Jaro Winkler dengan data terindeks

HASIL DAN PEMBAHASAN

Hasil pengujian dalam bentuk waktu yang menunjukkan lama waktu yang diperlukan untuk dapat melakukan prose pencarian kata. Hasil pengujian dengan menggunakan jumlah data 16.926 ditunjukkan pada Tabel 2, dan hasil pengujian dengan jumlah data 69.145 ditunjukkan pada Tabel 3. Satuan waktu yang digunakan adalah detik.

Tabel 2. Hasil Pengujian Dengan Jumlah Data 16.926

Kata	Pencarian Tepat		Jaro Winkler		Jaro Winkler Dengan Data Terindeks	
	Max	Min	Max	Min	Max	Min
ade	0.083	0.014	8.84951	6.9314	7.97546	0.94105
adi	0.24201	0.038	6.9554	5.69233	6.62538	0.26902
ani	0.004	0.003	7.75244	6.11535	8.26847	0.64204
ita	0.099	0.007	8.6805	6.42937	6.57438	1.42708
nia	0.13001	0.082	6.16735	5.85533	6.29936	0.66704
andi	0.16001	0.02	7.0264	6.91039	7.95345	0.034
budi	0.23801	0.036	7.19341	6.88639	7.57343	0.22101
mila	0.17001	0.037	11.2997	9.37254	9.01652	1.05806
rudi	0.045	0.041	10.1486	9.70956	13.7768	1.40208
wati	0.27002	0.049	9.53355	7.45943	10.1226	0.91405

anita	0.19701	0.16301	9.85256	9.33853	9.58555	0.17101
benar	0.16601	0.15801	13.0047	10.9956	14.3038	0.87905
Diana	0.047	0.044	16.1069	12.5677	13.4208	0.88505
indah	0.17901	0.16901	12.1197	10.5846	16.1909	1.34608
yonna	0.21801	0.18001	10.0916	9.80456	10.0696	0.72104
kirana	0.29202	0.10601	14.2228	12.4737	15.5789	1.64009
kurasa	0.23501	0.078	15.0239	13.2338	14.0658	1.31308
melati	0.18501	0.04	15.7229	10.9056	15.2429	0.76404
selena	0.23401	0.19501	15.3919	11.9037	14.4808	1.64109
vanesa	0.08	0.046	15.1039	14.2518	15.0249	1.48309
gunawan	0.16101	0.05	18.103	13.2088	19.5071	0.68204
kamelia	0.17801	0.04	27.9826	12.3637	12.6917	1.6901
katerin	0.082	0.048	17.229	16.849	18.4941	1.64009
permata	0.21001	0.14601	17.08	13.0067	13.9148	1.49709
vanessa	0.14601	0.042	18.306	15.5199	18.8281	1.32908
Benedith	0.068	0.053	23.3843	18.9071	21.8582	2.05712
meredith	0.53303	0.14601	21.4072	14.4518	14.2918	1.90211
rowandha	0.32902	0.17901	17.778	15.5199	21.6862	1.64109
sulastri	0.14201	0.09801	23.4763	22.0843	22.7603	1.93011
sutrisno	0.26502	0.13301	18.396	15.8799	16.781	1.97111
Alexander	0.14201	0.031	16.718	15.8919	15.1389	0.56203
heriyanto	0.21001	0.047	24.2634	21.6492	24.4554	1.96411
stephanie	0.28102	0.043	22.4413	21.9703	22.5143	1.7311
sukmawati	0.20101	0.11301	21.7532	18.003	23.6844	2.20213
sumarjono	0.11401	0.051	29.0127	22.7253	23.2023	2.55615
diah putri	0.14601	0.068	22.0453	20.4302	25.0234	2.04912
evangelina	0.15401	0.049	23.5194	16.4139	26.9745	1.7661
hardiyanto	0.17601	0.16601	22.1803	21.5912	26.1275	1.94111
kurniawati	0.21701	0.10101	25.6265	15.8559	23.1343	2.25213
para digma	0.30302	0.19001	16.6189	16.4299	22.0143	1.04306
bartolomeus	0.24401	0.032	17.078	16.677	18.2701	1.6791
cokrominoto	0.22701	0.18501	19.0281	16.3059	19.3491	1.7821
Grande Ansi	0.23801	0.16801	21.5632	17.374	18.292	0.63004
sri hartati	0.31202	0.05	23.7544	16.75	28.0426	1.8201
sudharmanto	0.20301	0.061	28.2886	17.312	27.4276	1.96811

Tabel 3. Hasil Pengujian Dengan Jumlah Data 69.145

Kata	Pencarian Tepat		Jaro Winkler		Jaro Winkler Dengan Data Terindeks	
	Max	Min	Max	Min	Max	Min
ade	0.002	0.001	34.381	27.2686	40.6883	3.63921
adi	0.11101	0.002	24.9734	22.7493	26.7945	0.99906
ani	0.001	0.001	31.0358	29.4057	28.4756	4.19224
ita	0.001	0.001	28.9707	23.7674	36.4701	2.97917

nia	0.13401	0.076	30.8068	22.9733	26.0125	3.76922
andi	0.11801	0.078	31.1238	29.9207	29.6107	1.43408
budi	0.38102	0.16001	31.1758	27.3806	31.5108	0.69004
mila	0.016	0.015	40.8883	36.9381	40.3293	3.67021
rudi	0.23501	0.09301	64.5107	49.2508	47.4417	4.87028
wati	0.39502	0.007	41.8044	36.9221	43.2285	3.4092
anita	0.21301	0.10401	42.7734	40.1553	43.8075	0.65504
benar	0.24801	0.21301	52.655	38.5692	55.3862	2.98817
Diana	0.15101	0.14501	66.9978	61.9305	60.0164	9.77556
indah	0.77104	0.011	83.9978	64.8097	56.6342	5.1893
yonna	0.13001	0.046	48.9418	40.4163	43.4625	3.77622
kirana	0.44503	0.15501	66.5928	63.2306	67.0588	7.59743
kurasa	0.28802	0.09201	63.1146	58.2673	66.7698	4.67527
melati	0.11601	0.031	60.9365	52.831	62.4526	3.17718
selena	0.20901	0.10101	69.438	63.2726	69.241	7.78245
vanesa	0.15101	0.10601	67.3888	64.8257	67.7249	5.92134
gunawan	0.08	0.01	83.8728	60.7205	68.3179	2.65615
kamelia	0.93105	0.20701	91.4702	59.6114	64.6077	2.94917
katerin	0.31302	0.18001	78.2755	62.7096	93.2413	4.92928
permata	0.11101	0.10301	83.2918	81.3196	94.1404	6.11635
vanessa	0.19301	0.055	87.472	80.4256	90.9592	5.81333
Benedith	0.11501	0.09101	107.227	88.5671	116.423	7.13641
meredith	0.26501	0.26302	121.52	92.0033	105.062	7.09541
rowandha	0.20601	0.18001	98.8356	90.7352	103.005	6.9404
sulastri	0.15601	0.09901	94.7864	86.848	104.684	6.22836
sutrisno	0.14201	0.09	103.172	102.041	103.488	7.98746
Alexander	0.35302	0.003	103.624	80.0926	120.218	1.88211
heriyanto	0.21701	0.052	125.104	116.721	128.97	8.92351
stephanie	0.16901	0.13901	110.036	95.2944	124.045	7.29042
sukmawati	0.22101	0.10301	119.423	117.27	134.697	8.19147
sumarjono	0.13201	0.08501	114.812	110.042	133.413	6.80139
diah putri	0.11301	0.09301	123.746	121.265	125.777	7.12041
evangelina	0.12901	0.089	98.0206	86.562	98.4046	7.23241
hardiyanto	0.14001	0.11501	131.681	128.075	140.344	6.47937
kurniawati	0.28802	0.16801	138.773	120.309	129.143	8.50549
para digma	0.23301	0.091	118.376	87.831	114.183	4.18024
bartolomeus	0.44502	0.14401	110.663	103.743	110.894	7.0364
cokrominoto	0.14301	0.12901	130.13	113.281	115.666	9.29353
Grande Ansi	0.29902	0.19001	110.051	101.668	101.668	1.56109
sri hartati	0.26002	0.15001	144.039	132.285	156.192	8.7175
sudharmanto	0.15001	0.13401	109.731	109.068	116.43	7.79745

Jurnal Teknovasi
Volume 04, Nomor 02, 2017, 69 – 78
ISSN : 2540-8389

Tabel 2 dan Tabel 3 menunjukkan jumlah waktu maksimum dan jumlah waktu minimun yang diperlukan untuk melakukan proses pencarian. Pada metode Jaro Winkler dengan data terindeks, antara waktu maksimum yang diperlukan dan waktu minimum yang diperlukan untuk dapat melakukan pencarian memiliki perbedaan waktu yang cukup besar. Hasl ini disebabkan oleh proses pengindeksan data pertama sekali yang dilakukan bila kata yang dicari belum pernah dicari sebelumnya. Namun bila kata yang dimaksud pernah dicari, maka hasil pencarian akan menjadi lebih singkat yakni ditunjukkan oleh jumlah waktu minimum. Dari hasil tersebut juga tampak bahwa waktu yang diperlukan untuk melakukan pencarian kata yang mirip cendrung meningkat seiring dengan meningkatnya jumlah karakter yang dicari. Untuk mengetahui peningkatan waktu proses pencarian antara metode Jaro Winkler dan metode Jaro Winkler dengan data terindeks, pada Tabel ditunjukkan peningkatan kecepatan waktu proses dalam persentasi dan perbandingan.

Tabel 4 Rekapitulasi Peningkatan Kecepatan

Jumlah Data	Peningkatan Kecepatan	Persen
16.926	1669.417	90.2033
69.145	1679.044	92.22565

Dengan menggunakan MAP, rata rata peningkatan kecepatan proses pencarian adalah 90% untuk jumlah data 16.926 data dan 92% untuk 69145 data. Dengan pengujian dan hasil yang ditunjukkan pada tabel 4 maka pengujian perbaikan menunjukkan hasil yang sangat baik.

KESIMPULAN

Dari hasil pengujian perbaikan metode Jaro Winkler dengan data terindeks, diperoleh bahwa:

1. Peningkatan pencarian dengan metode Jaro Winkler dengan data terindeks berhasil dilaksanakan dengan peningkatan kecepatan hasil pencarian mencapai 90% lebih cepat untuk jumlah data 16.926 dan 92% lebih cepat untuk jumlah data 69.145 dari pada penggunaan secara langsung metode Jaro Winkler.
2. Peningkatan jumlah data yang digunakan menunjukkan peningkatan waktu yang diperlukan untuk mencari kata yang mendekati.
3. Semakin panjang kata, waktu yang diperlukan semakin besar.
4. Perbaikan metode Jaro Winkler dengan data terindeks memerlukan waktu yang relatif sama dengan metode Jaro Winkler tanpa data terindeks hanya untuk pencarian kata baru pertama sekali.

SARAN

Jaro Winkler merupakan suatu metode pencarian kedekatan string yang cukup mudah dan murah sumber daya karena menggunakan algoritma yang sederhana. Dengan menggunakan metode Jaro Winkler dengan data terindeks, dapat diterapkan pada sistem tertanam dalam metode pencarian data terdekat, terutama untuk data yang statis.

Jurnal Teknovasi
Volume 04, Nomor 02, 2017, 69 – 78
ISSN : 2540-8389

DAFTAR PUSTAKA

G. Navarro, "A Guided Tour to Approximate String Matching," *ACM Computing Surveys,* vol. 33, no. 1, pp. 31-88, 2001.

Y. Rochmawati and R. Kusumaningrum, "Studi Perbandingan Algoritma Pencarian String dalam Metode Approximate String Matching untuk Identifikasi Kesalahan Pengetikan Teks," *Jurnal Buana Informatika,* pp. 125-134, April 2016.

K. Dressler and A.-C. N. Ngomo, "On the Efficient Execution of Bounded Jaro-Winkler Distances," 12 September 2014. [Online]. Available: http://www.semantic-web-journal.net/content/efficient-execution-bounded-jaro-winkler-distances. [Accessed 15 April 2017].

P. E. Black, "Dictionary of Algorithms and Data Structures," National Institute of Standard and Technology, 27 May 2014. [Online]. Available: https://xlinux.nist.gov/dads/HTML/jaroWinkler.html. [Accessed 15 April 2017].

I. T. Administration, "Damerau–Levenshtein distance usage,ITA's Data Services Platform," U.S. Exporting Data Authoritative , April 2017. [Online]. Available: http://developer.trade.gov/. [Accessed April 2017].

L. BOYTSOV, "Indexing Methods for Approximate Dictionary Searching: Comparative Analysis," Asociation for Computing Machinery, May 2011. [Online]. Available: http://doi.acm.org/10.1145/1963190.1963191. [Accessed 14 April 2017].

W. E. Winkler, "Overview of Record Linkage," Statistical Research Division,U.S. Census Bureau, Washington, 2006.