Analisa Penggunaan K-Gram pada Karakter, Kata dan Kalimat untuk Mendeteksi Kesamaan Dokumen

by Ida Widaningrum, Dyah Mustikasari, Rizal Arifin, Erika Diyah Cahyani

Submission date: 19-Sep-2023 11:03AM (UTC+0700)
Submission ID: 2170317732
File name: 1._Analisa_Penggunaan_K-Gram_pada_Karakter,_Kata_dan_Kalimat.pdf (282.83K)
Word count: 3022
Character count: 18685

SEMINAR NASIONAL TEKNOKA

Vol. 5, 2020
ISSN No. 2502-8782

Analisa Penggunaan K-Gram pada Karakter, Kata dan Kalimat untuk Mendeteksi Kesamaan Dokumen

Ida Widaningrum[1)], Dyah Mustikasari[2)], Rizal Arifin[3)], & Erika Diyah Cahyani[4)]
[1,2,4)]Program Studi Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Ponorogo
[3)]Program Studi Teknik Mesin, Fakultas Teknik, Universitas Muhammadiyah Ponorogo
E-mail: iwidaningrum.as@gmail.com, dyah.mustikasari@gmail.com, rizalarifin@gmail.com, erikadyah7@gmail.com

Abstrak – Pemanfaatan teknologi digital menjadi sebuah kebutuhan saat ini, salah satu komponennya berupa dokumen. Pendeteksian kesamaan bisa menggunakan berbagi macam cara, diantaranya adalah metode fingerprinting. Fingerprint memiliki prinsip kerja menggunakan teknik hashing dan K-gram. Penelitian ini difokuskan pada model deteksi menggunakan K-gram dengan menggunakan algoritma winnowing dan python sebagai bahasa pemrograman. Pengujian parsing k-gram menggunakan 5 buah k yaitu k=2 k=3 k=4 k=5 k=6. Hasilnya, parsing karakter mendapatkan presentase lebih besar dari presentase manual karakter. Presentase parsing kata, memiliki presentase yang paling mendekati dari presentase manual. Sedangkan pada kalimat, presentasenya paling rendah dari presentase manual.

Kata Kunci: *Python, Karakter K-Gram, Kata K-Gram, Kalimat K-Gram, Algoritma Winnowing, Kemiripan Dokumen*

Abstract – *The use of digital technology is now a necessity; one of its components is documents. Similarity detection can use a variety of methods, including the fingerprinting method. Fingerprint has a working principle using hashing techniques and K-gram. This research is focused on the detection model using K-gram using the winnowing algorithm and python as a programming language. The k-gram parsing test uses 5 k pieces, namely k = 2 k = 3 k = 4 k = 5 k = 6. As a result, the character parsing gets a larger percentage than the manual character percentage. The percentage of word parsing has the closest percentage of the manual percentage. while in sentences, the percentage is the lowest than the manual percentage.*

Keyword: *Python, K-Gram characters, K-Gram words, K-Gram sentences, Winnowing Algorithm, Document Similarity*

1 PENDAHULUAN

Pemanfaatan teknologi digital menjadi sebuah kebutuhan saat ini, diantaranya berupa dokumen. Dokumen digital sangat mudah untuk diakses. Penelusuran karya-karya dalam bentuk digital, salah satu diantaranya adalah artikel jurnal atau hasil penelitian bisa kita dapatkan melalui internet. Hal ini memungkinkan siapa saja untuk meniru atau mengutip hasil karya orang lain. Penggunaan ide, karya atau tulisan orang lain tanpa seijin orang tersebut dianggap plagiat. Menurut KBBI online, plagiasi adalah meniru pekerjaan orang lain, mengambil karangan orang lain tanpa sepengetahuan penulisnya [1]. Tindakan ini disebabkan oleh tuntutan yang mengharuskan kita menulis, sebagai syarat untuk meraih gelar akademik yang ditempuh atau suatu jenjang jabatan akademik. Hal ini disebabkan dari belum terbiasa atau kurangnya ide untuk membuat penelitian, sehingga terjebak dalam *plagiarism*. Salah satu cara yang digunakan untuk mengindari plagiat adalah dengan melakukan sitasi, yaitu kita harus mencantumkan sumber kutipan yang kita gunakan [2].

Untuk mengetahui apakah artikel atau tulisan kita buat termasuk kedalam plagiasi atau bukan, biasa digunakan alat untuk mendeteksinya. Sudah banyak alat pendeteksi plagiasi yang ada dipasaran, baik itu yang berbayar ataupun tidak. Metode yang digunakan untuk mendeteksi plagiasi banyak macamnya, salah satu dari metode tersebut adalah *fingerprinting*. *Fingerprinting* memiliki prinsip kerja dengan menggunakan teknik *hashing*. Teknik *hashing* adalah merubah data menjadi serangkaian bilangan bulat. Jumlah data yang digunakan dinyatakan dengan K-gram [3].

K-gram merupakan salah satu proses yang digunakan dalam *text mining* dan pengolahan bahasa. K-gram merupakan sekumpulan karakter, kata atau kalimat yang ada pada dokumen dan disaat menghitung k-gram dilakukan dengan menggerakkan

Dipresentasikan pada Tanggal 28 November 2020

satu string maju ke depan [4]. Berdasarkan KBBI online [1] karakter adalah huruf, angka, ruang atau simbol khusus, kata merupakan satuan Bahasa yang dapat berdiri sendiri, sedangkan kalimat adalah kesatuan ujar yang mengungkapkan suatu konsep pikiran.

Untuk mendeteksi kesamaan dokumen, maka dibuat *system* dengan teknik *Hashing* dengan menggunakan Algoritma Winnowing.

Penelitian sebelumnya yang sudah ada adalah, menggunakan 5 (karakter)-gram dalam metode String Matching dengan Algoritma Rabin-Karp untuk mendeteksi plagiasi [5], sedangkan [6] menggunakan Algoritma Winnowing. Penelitian [7] menggunakan 5 (karakter)-gram ini untuk mendeteksi plagiasi dalam Bahasa Inggris, sedangkan [8] menggunakan 9 karakter untuk mendeteksi judul kemiripan skripsi.

Pada penelitian ini akan dibandingkan penggunaan n-gram dalam karakter, kata dan kalimat dengan menggunakan algoritma Winnowing. Diharapkan dari penelitian ini bisa dilihat, manakah yang lebih efektif digunakan dari ketiga pilihan tersebut. Penelitian ini merupakan bagian dari serangkaian penelitian yang didanai oleh Direktorat Riset dan Pengabdian Masyarakat Direktorat Jenderal Penguatan Riset dan Pengembangan Kementerian Riset, Teknologi, dan Pendidikan Tinggi.

2 LANDASAN TEORI

Berdasarkan KBBI, plagiasi merupakan pengambilan karya orang lain dan menjadikannya seolah-olah karya sendiri [1]. K-gram merupakan salah satu proses yang digunakan dalam *text mining* dan pengolahan bahasa. K-gram merupakan sekumpulan karakter, kata atau kalimat yang ada pada dokumen dan disaat menghitung k-gram dilakukan dengan menggerakkan satu string maju ke depan [4]. Dalam K-Gram karakter, karakter akan dikumpulkan kedalam kelompok yang terdiri dari sejumlah k (misalnya uni, bi, tri dan seterusnya). Misalnya dalam kata "BELAJAR", akan didapatkan k-gram karakter sebagai berikut.

Unigram: B, E, L, A, J, A, R
Bigram: _B, BE, EL, LA, AJ, JA, R_
Trigram: _BE, BEL, ELA, LAJ, AJA, JAR, AR_, R_ _

K-Gram kata, sebagai contoh terdapat sebuah kalimat "Kemarin aku beli sepeda baru keren". Akan didapatkan k-gram kata sebagai berikut.

Unigram: kemarin, aku, beli, sepeda, baru, keren.
Bigram: kemarin aku, aku beli, beli sepeda, sepeda baru, baru keren.
Trigram: kemarin aku beli, sepeda baru keren.

K-Gram kalimat Terdapat sebuah paragraf "Universitas Muhammadiyah Ponorogo adalah universitas berintegritas. Mempunyai komitmen tinggi untuk menjadi lembaga pendidikan dan pelatihan kejuruan berstandar nasional / internasional, berwawasan unggul, kompetitif dan pofesional dengan berdasarkan imtaq. Menjadi universitas yang unggul dalam penguasaan ilmu pengetahuan, teknologi, dan seni berdasarkan nilai-nilai islami". Didapatkan k-gram kalimat sebagai berikut.

Unigram: universitas muhammadiyah ponorogo adalah universitas berintegritas; mempunyai komitmen tinggi untuk menjadi lembaga pendidikan dan pelatihan kejuruan berstandar nasional / internasional, berwawasan unggul, kompetitif dan pofesional dengan berdasarkan imtaq; menjadi universitas yang unggul dalam penguasaan ilmu pengetahuan, teknologi, dan seni berdasarkan nilai-nilai islami.

Rolling hash adalah salah satu metode dengan teknik *hash,* digunakan untuk menemukan nilai *hash* dari rangkaian grams. *Rolling hash* merupakan fitur yang menghitung nilai *hash* tanpa mengulangi seluruh *string*. Nilai *hash* adalah nilai numerik yang dibentuk dari kode *ASCII* [9].

Algoritma Winnowing merupakan salah satu algoritma *text mining* dengan menggunakan dokumen *fingerprint* yang kemudian memakai teknik hashing dalam mencocokkan dua atau lebih dokumen. Teknik *hashing* sendiri digunakan untuk mengubah setiap karakter dalam dokumen menggunakan *ASCII*. Fungsi hash yang digunakan Winnowing adalah *Rolling Hashing* [10].

Koefisien Jaccard digunakan untuk menghitung kemiripan himpunan string yang nilai hashnya telah dihitung, koefisien Jaccard ini dapat mengukur prosentase kemiripan teks [11].

3 METODOLOGI PENELITIAN

Penghitungan Data menggunakan algoritma *Winnowing*.

a. Pembuangan tanda baca, spasi dan symbol-simbol seperti =, #, %, &, (,), -,_, $, @, !, /,",
b. Pembentukan rangkaian metode K-Gram berupa karakter, kata, dan kalimat. Dan menggunakan 5 jenis K yaitu yaitu K=2 K=3 K=4 K=5 K=6.
c. Perhitungan Fungsi Hash untuk tiap k-gram. *Rolling hash* adalah salah satu metode dari teknik *hashing* untuk mencari nilai *hash* dari rangkaian

DOI: 10.22236/teknoka.v5i.333

ISSN No. 2502-8782

grams seluruh *karakter*. Nilai *hash* adalah angka yang dibentuk dari kode *ASCII*.

d. *Rolling hash* adalah fungsi yang menghitung nilai hash tanpa mengulangi seluruh string

$$H(ch) = c_1 * b^{(k-1)} + c_2 * b^{(k-2)} + \cdots + c_k * b^{(k-k)} \quad (1)$$

Keterangan:
c = nilai ascii karakter
b = basis (bilangan prima)
k = banyak karakter

e. Membagi ke dalam *Window* tertentu dari Nilai Hash

f. Pemilihan *Fingerprint* dari Setiap *Window*. Langkah terakhir yaitu memilih nilai terkecil dari setiap *window* untuk dijadikan dokumen *fingerprint*

g. Similaritas *Jaccard* dapat digunakan sebagai menghitung kemiripan dari kumpulan string yang telah dihitung nilai hash dan dapat mengukur prosentase kemiripan teks

Berikut ini rumus persamaan *Jaccard Coefficient* [8]

$$Similarity = \frac{Jumlah\ fingerprint\ sama}{Total\ seluruh\ fingerprint} x\ 100 \quad (2)$$

Desain dan metode penelitian yang digunakan ditampilkan pada gambar 1.

***Gambar 1** Alur penghitungan K-gram menggunakan Algoritma Winnowing*

4 HASIL DAN PEMBAHASAN

Berikut akan sedikit dijelaskan contoh penerapan **k-gram pada parsing karater** pada algoritma winnowing:

a. Langkah pertama adalah penghilangan karakter yang tidak diperlukan, seperti simbol, tanda baca, tanda spasi, mengubah huruf besar menjadi kecil, dan simbol-simbol selain huruf a sampai z.

b. Langkah kedua adalah *parsing* karakter menggunakan k=6

Untuk kalimat I terbentuklah rangkaian gram sebagai berikut :

univer nivers iversi versit ersita rsitas..... .....
.... ponoro onorog norogo
Untuk kalimat II terbentuklah rangkaian gram sebagai berikut :
univer nivers iversi versit ersita rsitas ....
.....madiun

c. Langkah ketiga selanjutnya mencari nilai *Hash* dari setiap gram yang dibentuk. Sebagai contoh ditentukan nilai basisnya adalah 3 dan gram nya adalah 6.

Perhitungan untuk kalimat I:
$H_{(univer)}$ = ascii(u) * $3^{(5)}$ + ascii(n) * $3^{(4)}$ + ascii(i) * $3^{(3)}$ + ascii(v) * $3^{(2)}$ + ascii(e) * $3^{(1)}$ + ascii(r) * $3^{(0)}$
= 117 * 243+ 110 * 81 + 105 * 27 + 118 * 9 + 101 * 3 + 114 * 1
= 28.431 + 8.910 + 2.835 + 1.062 + 303 + 114
= 41.655
dan seterusnya, sehingga diperoleh perhitungan sebagai berikut:
41655 39787 39276 ....
Perhitungan untuk kalimat II:
$H_{(univer)}$ = ascii(u) * $3^{(5)}$ + ascii(n) * $3^{(4)}$ + ascii(i) * $3^{(3)}$ + ascii(v) * $3^{(2)}$ + ascii(e) * $3^{(1)}$ + ascii(r) * $3^{(0)}$
= 117 * 243+ 110 * 81 + 105 * 27 + 118 * 9 + 101 * 3 + 114 * 1
= 28.431 + 8.910 + 2.835 + 1.062 + 303 + 114
= 41.655
Dan seterusnya, sehingga diperoleh perhitungan sebagai berikut:
41655 39787 39276 ....

d. Setelah mendapatkan nilai hash dari proses *Rolling Hash,* selanjutnya adalah membagi nilai-nilai hash tersebut menurut *window*:

Untuk kalimat I, *window*nya sebagai berikut:
(41655 39787 39276 41399 38272)
(..... ....)
Kemudian untuk kalimat 2 *window*nya adalah sebagai berikut:

DOI: 10.22236/teknoka.v5i.333

ISSN No. 2502-8782

(41655 39787 39276 41399 38272)
(..... ....)

e. Langkah selanjutnya, memilih nilai hash terkecil dengan posisi paling kanan dari setiap *window* untuk dijadikan *fingerprint*. Berikut ini adalah nilai *fingerprint* yang didapat:

Untuk Kalimat I *fingerprint* yang didapat adalah:

Jumlah *fingerprint*: 5
(38272, 37291, 36621, 35990, 36459)
Untuk Kalimat II *fingerprint* yang didapat adalah:
Jumlah *fingerprint*: 4
(38272, 37180, 36667, 36351)

f. Kemudian selanjutnya adalah menghitung prosentase kemiripan dengan menggunakan persamaan *jaccard similarity coefficient* .

Fingerprint sama: 1
Fingerprint berbeda: 7

$$Similarity = \frac{Jumlah\ fingerprint\ sama}{Total seluruh\ fingerprint} x\ 100 = = \frac{1}{8} x\ 100\% = 12{,}5\%$$

Jadi, hasil Similaritasnya adalah 12.5%

Contoh **penerapan k-gram pada *parsing* kata** pada algoritma *winnowing*:

a. Pertama, menghilangkan karakter yang tidak diperlukan, seperti simbol, tanda baca, tanda spasi, mengubah huruf besar menjadi kecil, dan simbol-simbol selain huruf a sampai z.

b. Pembentukan rangkaian gram,

Untuk kalimat I, terbentuk rangkaian gram sebagai berikut:
sebanyakmahasiswa mahasiswauniversitas universitasmerdeka merdekaunmer unmermadiun madiunakan akanmengikuti mengikutikuliah kuliahkerja kerjanyata nyatakkn kkndi dikabupaten kabupatenmadiun madiunmulai mulaijanuari januarihingga hinggafebruari februariacara acarapelepasan pelepasanpemberangkatan pemberangkatanpeserta pesertakkn kkndi digedung gedunggraha grahasamiarto samiartounmer unmerkota kotamadiun

Untuk kalimat II, terbentuk rangkaian gram sebagai berikut:
mahasiswauniversitas universitasmuhammadiyah muhammadiyahponorogo ponorogopada padahari harikamis kamisdi diberangkatkan berangkatkanoleh olehrector rectorumpo umpountuk untukkkn kknkuliah kuliahkerja kerjanyata nyatainternasional internasionalke kekamboja kambojaphilipina philipinadan danmalaysia malaysia

c. Proses *Rolling Hash*, untuk menghasilkan nilai hash dari setiap gram yang dibentuk. Contohnya, ditentukan nilai basis = 3 dan gram = 2.

$H_{(}$mahasiswauniversitas$_{)}$
= ascii(m) + ascii(a) + ascii(h) + ascii(a) + ascii(s) + ascii(i) + ascii(s) + ascii(w) + ascii(a) + ascii(u) + ascii(n) + ascii(i) + ascii(v) + ascii(e) + ascii(r) + ascii(s) + ascii(i) + ascii(t) + ascii(a) + ascii(s)
=(109 + 97 + 104 + 97 + 115 + 105 + 115 + 119 + 97) * 3^1 + (117 + 110 + 105 + 118 + 101+ 114 + 115 + 105 + 116 + 97 +115) * 3^0
= 958 * 3^1 + 1213 * 3^0 = 4.087
dan seterusnya, sehingga diperoleh perhitungan sebagai berikut :
4087 4908 4690 3055 1638 ....
Perhitungan untuk kalimat II:
$H_{(}$sebanyakmahasiswa$_{)}$
= ascii(s) + ascii(e) + ascii(b) + ascii(a) + ascii(n) + ascii(y) + ascii(a) + ascii(k) + ascii(m) + ascii(a) + ascii(h) + ascii(a) + ascii(s) + ascii(i) + ascii(s) + ascii(w) + ascii(a)
=(115 + 101 + 98 + 97 + 110 + 121 + 97 + 107)*3^1 + (109 + 97 + 104 + 97 + 115 + 105 + 115 + 119 + 97)*3^0
= 846 * 3^1 + 958 * 3^0 = 3496
Dan seterusnya, sehingga diperoleh perhitungan sebagai berikut:
3496 4087 4368 2738 2291 ....

d. Bagi nilai-nilai hash tersebut menurut *window*.

e. Pilih nilai hash terkecil dengan posisi paling kanan dari setiap *window*, untuk dijadikan *fingerprint*. Berikut adalah nilai *fingerprint* yang didapat
Untuk Kalimat I.
Jumlah *fingerprint*: 6
[2874, 1638, 1793, 1914, 1610, 1349]
Untuk Kalimat II.
Jumlah *fingerprint*: 11
[2291, 2206, 2116, 1947, 1177, 1570, 2354, 2465, 1249, 1931, 1914]
Kemudian selanjutnya adalah menghitung prosentase kemiripan dengan menggunakan persamaan *jaccard similarity coefficient* .

Fingerprint sama: 1

Fingerprint berbeda: 16

$$Similarity = \frac{Jumlah\ fingerprint\ sama}{Total seluruh\ fingerprint} x\ 100 = = \frac{1}{17} x\ 100\% = 5{,}8\%$$

Jadi, hasil Similaritasnya adalah 5.8%

DOI: 10.22236/teknoka.v5i.333

ISSN No. 2502-8782

Contoh penerapan **k-gram pada *parsing* kalimat** pada algoritma *winnowing*:

a. Sama seperti sebelumnya, pertama hilangkan karakter yang tidak diperlukan, seperti simbol, tanda baca, tanda spasi. Ubah huruf besar menjadi kecil.

b. Kedua, *parsing* kalimat menggunakan k=2

c. Langkah ketiga, proses *Rolling Hash* untuk menghasilkan nilai dari setiap *gram* yang dibentuk. Sebagai contoh ditentukan nilai basisnya adalah 3 dan gram nya 2.

Perhitungan untuk kalimat I dan diperoleh perhitungan sebagai berikut:
27256 38644 50133 ....
Perhitungan untuk kalimat II dan diperoleh perhitungan sebagai berikut:
25722 38644 47018 58241 ....

d. Setelah mendapatkan nilai hash dari proses *Rolling Hash* maka langkah keempat selanjutnya adalah membagi nilai-nilai *hash* tersebut menurut *window*.

Untuk kalimat I, *window*nya sebagai berikut:
(27256 38644 50133)
(38644 50133 64471)
(..... ....)
Kemudian untuk kalimat 2 *window*nya adalah sebagai berikut:
(27256 38644 50133)
(38644 47018 58241)
(..... ....)

e. Langkah selanjutnya adalah memilih nilai hash terkecil dengan posisi paling kanan dari setiap *window* untuk dijadikan *fingerprint* (penanda). Berikut ini adalah nilai *fingerprint* yang didapat

Untuk Kalimat I *fingerprint* yang didapat adalah:
Jumlah *fingerprint*: 8

Fingerprint dokumen pertama:

[27256, 38644, 50133, 56613, 26367, 20050, 13400, 23866]
Untuk Kalimat II *fingerprint* yang didapat adalah:
Jumlah *fingerprint*: 8
Fingerprint dokumen kedua:
[25722, 38644, 47018, 56613, 26367, 26731, 23952, 25807]
Kemudian selanjutnya adalah menghitung prosentase kemiripan dengan menggunakan persamaan *jaccard similarity coefficient*.
Fingerprint sama: 3

Fingerprint berbeda: 10

$$Similarity = \frac{Jumlah\ fingerprint\ sama}{Total\ seluruh\ fingerprint} x\ 100 = \frac{3}{13} x\ 100\% = 23{,}1\%$$

Jadi, hasil Similaritasnya adalah 23.1 %

Analisis 3 *Parsing* K-Gram

Pengujian parsing k-gram dengan menggunakan 5 buah k yaitu k=2 k=3 k=4 k=5 k=6. k-gram pada karakter, kata, kalimat dengan ketentuan window = 5 dan basis = 3 dengan dokumen uji adalah uji.py dan dokumen pembanding.py. Kemudian dilihat selisih dari presentase perhitungan manual berdasarkan setiap parsing k-gram.

Hasil parsing karakter, dokumen 1 memiliki total 17.768 karakter. Dokumen 2 adalah dokumen yang sama, namun dihilangkan beberapa kalimat untuk membedakan isi dari dokumen sehingga dokumen 2 memiliki total 15.123 karakter. Menurut perhitungan manual prosentase kemiripan antar dua dokumen tersebut seharusnya menghasilkan 15,123/17,768 x 100% = 85,11% .

Pada parsing kata, dokumen 1 memiliki total 2.802 kata, dan dokumen 2 adalah dokumen yang sama namun dihilangkan beberapa kalimat untuk membedakan isi dari dokumen sehingga dokumen 2 memiliki total 2.406 kata. Menurut perhitungan manual prosentase kemiripan antar dua dokumen tersebut seharusnya menghasilkan 2,406/2,802 x 100% = 85,87% .

Pada parsing kalimat, dokumen 1 memiliki total 200 kalimat dan dokumen 2 adalah dokumen yang sama namun dihilangkan beberapa kalimat untuk membedakan isi dari dokumen sehingga dokumen 2 memiliki total 171 kalimat. Menurut perhitungan manual prosentase kemiripan antar dua dokumen tersebut seharusnya menghasilkan 200/171 x 100% = 85,5% .

Hasil analisis ketiga parsing k-gram dapat dilihat pada tabel 1.

***Tabel 1** Hasil analisis parsing k-gram*

No.	Parsing	K-	Hasil	Hasil manual	Selisih
1.	karakter	ii	97.7%	85.11	12.59%
2.		iii	96.8%	%	11.69%
3.		iiii	97.5%		12.39%
4.		iiiii	93.8%		8.69%
5.		iiiiii	90.0%		4.89%
6.	Kata	ii	84.8%	85.87	-1.07%
7.		iii	79.8%	%	-6.07%
8.		iiii	77.0%		-8.87%
9.		iiiii	75.0%		-10.87%
10.		iiiiii	73.2%		-12.67%

DOI: 10.22236/teknoka.v5i.333

ISSN No. 2502-8782

11.	Kalimat	ii	58.7%	85.5%	-26.8%
12.		iii	55.4%		-30.1%
13.		iiii	55.3%		-30.2%
14.		iiiii	47.7%		-37.8%
15.		iiiiii	41.6%		-43.9%

5 SIMPULAN

Hasil parsing karakter mendapatkan presentase yang lebih besar dari presentase manual karakter. Walaupun menggunakan 5 buah k, hasil menunjukan presentase yang cukup besar pada parsing karakter dari pada parsing kata dan kalimat. Presentase pada parsing kata memiliki presentase yang paling mendekati hasil presentase manual. Sedangkan pada kalimat, hasil presentasenya paling rendah dari pada hasil presentase manual.

KEPUSTAKAAN

[1] Kemdikbud. (2016, 22 Januari 2020). *KBBI online*.

[2] P. Istiana, "Membuat Sitasi dan Daftar Pustaka," in "Materi Pelatihan Kursus Pelatihan Instruktur Literasi Informasi. ," Universitas Padjajaran Bandung, Universitas Sanata Dharma, Yogyakarta 2013, vol. 27 December 2014.

[3] I. Widiastuti, C. Rahmad, and Y. Ariyanto, "Aplikasi Pendeteksi Kemiripanpada Dokumen Menggunakan Algoritma Rabin Karp," *Jurnal Informatika Polinema,* vol. 1, no. 2, pp. 13-13, 2015.

[4] S. Sunardi, A. Yudhana, and I. A. Mukaromah, "Implementasi Deteksi Plagiarisme Menggunakan Metode N-Gram Dan Jaccard Similarity Terhadap Algoritma Winnowing," 2018.

[5] A. Prastyanti and S. N. Endah, "Sistem deteksi kemiripan kata pada dua dokumen menggunakan algoritma Rabin-Karp," Universitas Diponegoro, 2014.

[6] R. Y. Dillak, F. Laumal, and L. J. Kadja, "Sistem Deteksi Dini Plagiarisme Tugas Akhir Mahasiswa Menggunakan Algoritma Ngrams dan Winnowing," *Jurnal Ilmiah Flash,* vol. 2, no. 1, pp. 12-18, 2016.

[7] A. Kurniawati and I. Wicaksana, "Perbandingan pendekatan deteksi plagiarism dokumen dalam bahasa inggris," in *Proceeding, Seminar Ilmiah Nasional Komputer dan Sistem Intelijen (KOMMIT 2008)*, 2008: Gunadarma University.

[8] N. Alamsyah, "Perbandingan Algoritma Winnowing Dengan Algoritma Rabin Karp Untuk Mendeteksi Plagiarisme Pada Kemiripan Teks Judul Skripsi," *Technologia: Jurnal Ilmiah,* vol. 8, no. 3, pp. 124-134, 2017.

[9] B. Zaman, E. Hariyanti, and E. Purwanti, "Sistem Deteksi Bahasa pada Dokumen menggunakan N-Gram," *Multinetics,* vol. 1, no. 2, pp. 21-26, 2015.

[10] A. Radili and S. Sanjaya, "Penerapan Metode Winnowing Fingerprint dan Naive Bayes untuk Pengelompokan Dokumen," *Jurnal CoreIT: Jurnal Hasil Penelitian Ilmu Komputer dan Teknologi Informasi,* vol. 3, no. 2, pp. 69-75, 2018.

[11] S. Niwattanakul, J. Singthongchai, E. Naenudorn, and S. Wanapu, "Using of Jaccard coefficient for keywords similarity," in *Proceedings of the international multiconference of engineers and computer scientists*, 2013, vol. 1, no. 6, pp. 380-384.

DOI: 10.22236/teknoka.v5i.333

ISSN No. 2502-8782

Analisa Penggunaan K-Gram pada Karakter, Kata dan Kalimat untuk Mendeteksi Kesamaan Dokumen

ORIGINALITY REPORT

7%	5%	0%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	download.garuda.ristekdikti.go.id Internet Source	2%
2	jurnal.fikom.umi.ac.id Internet Source	2%
3	pasca.umpo.ac.id Internet Source	2%
4	Submitted to Universitas Mercu Buana Student Paper	2%

Exclude quotes	On	Exclude matches	< 2%
Exclude bibliography	On		