ARTIKEL

SISTEM PENDETEKSI PLAGIASI PADA ARTIKEL MENGGUNAKAN ALGORITMA RABIN-KARP DI CV. DIGIMEDIA CORPINDO

Oleh:

IBNU NGAULANSYAH

11.1.03.02.0156

Dibimbing oleh:

1. **Daniel Swanjaya, M. Kom.**
2. **Resty Wulanningrum, M. Kom.**

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS NUSANTARA PGRI KEDIRI

2018

Artikel Skripsi
Universitas Nusantara PGRI Kediri

SURAT PERNYATAAN
ARTIKEL SKRIPSI TAHUN 2018

Yang bertanda tangan di bawah ini:

Nama Lengkap	: IBNU NGAULANSYAH
NPM	: 11.1.03.02.0156
Telepun/HP	: 086790576144
Alamat Surel (Email)	: ellibnu87@gmail.com
Judul Artikel	: Sistem Pendeteksi Plagiasi Pada Artikel Menggunakan Algoritma Rabin-Karp di CV. Digimedia Corpindo
Fakultas – Program Studi	: Fakultas Teknik – Program Studi Teknik Informatika
Nama Perguruan Tinggi	: Universitas Nusantara PGRI Kediri
Alamat Perguruan Tinggi	: Jl. KH. Achmad Dahlan No. 76 Kediri

Dengan ini menyatakan bahwa:

a. artikel yang saya tulis merupakan karya saya pribadi (bersama tim penulis) dan bebas plagiarisme;
b. artikel telah diteliti dan disetujui untuk diterbitkan oleh Dosen Pembimbing I dan II.

Demikian surat pernyataan ini saya buat dengan sesungguhnya. Apabila di kemudian hari ditemukan ketidaksesuaian data dengan pernyataan ini dan atau ada tuntutan dari pihak lain. Saya bersedia bertanggungjawab dan diproses sesuai dengan ketentuan yang berlaku.

Mengetahui		Kediri, 30 Januari 2018
Pembimbing I	Pembimbing II	Penulis,
Daniel Swanjaya, M. Kom. NIDN. 0723098303	Resty Wulanningrum,M.Kom. NIDN. 0719068072	Ibnu Ngaulansyah 11.1.03.02.0156

Ibnu Ngaulansyah | 11.1.03.02.0156
Fak Teknik – Prodi Teknik Informatika
simki.unpkediri.ac.id
|| 1||

SISTEM PENDETEKSI PLAGIASI PADA ARTIKEL MENGGUNAKAN ALGORITMA RABIN-KARP DI CV. DIGIMEDIA CORPINDO

Ibnu Ngaulansyah
11.1.03.02.0156
Fakultas Teknik – Prodi Teknik Informatika
ellibnu87@gmail.com
Daniel Swanjaya, M. Kom.dan Resty Wulanningrum,M.Kom.

UNIVERSITAS NUSANTARA PGRI KEDIRI

ABSTRAK

Sistem pendeteksi plagiasi pada artikel yang dilakukan di CV. Digimedia Corpindo ini merupakan penelitian yang dilatar belakangi oleh berkembangnya internet di kalangan masyarakat sehingga banyak bermunculan *website-website* yang menyediakan informasi kurang akurat sampai dengan berisi dengan hal-hal yang berbau plagiasi. Menghindari plagiasi di beberapa *website* CV. Digimedia Corpindo, penelitian ini dilakukan guna mencari calon *content writer* baru yang lebih kompeten.

Permasalahan yang dialami dalam penelitian ini adalah (1) Bagaimana menerapkan algoritma Rabin-Karp untuk mendeteksi sebuah tingkat kemiripan artikel yang dibual oleh calon *content writer* dengan sumber yang diberikan? (2) Bagaimana menentukan presentase tingkat kemiripan artikel yang dibuat oleh pelamar?

Sistem pendeteksi plagiasi pada artikel ini nantinya akan dibandingkan dengan beberapa artikel sumber yang telah dimasukkan oleh *admin* ke dalam sistem kemudian dibandingkan dengan artikel yang diinputkan oleh pelamar. Dengan menggunakan algoritma *Rabin-Karp* yang sudah terbukti mampu melakukan pencocokan *string* dalam jumlah banyak. Secara garis besar artikel terlebih dahulu akan *diparsin*g dalam bentuk *k-grams* kemudian dihitung nilai *hashing* sampai menentukan tingkat *similatrity* pada artikel yang dibuat oleh pelamar tersebut.

Kesimpulan dari penelitian ini adalah (1) Algoritma *Rabin-Karp* mampu digunakan untuk mendeteksi tingkat kemiripan (similarity) artikel yang dibuat oleh calon *content writer* dibandingkan dengan artikel sumber yang dimasukkan oleh penguji. Caranya yakni dimulai dengan *text preprocessing*, *parsing k-grams*, kemudian dilakukan perhitungan hashing dan *string mathing* sehingga menghasilkan nilai *similarit*y. (2) Dalam penentuan tingkat *similarity*, banyaknya kata, karakter dan ada atau tidaknya spasi dalam pengujian turut mempengaruhi nilai *similarity* dan juga jumlah *parsing k-grams* juga berpengaruh dalam tingkat *similarity*.

Berkaca dari kesimpulan hasil penelitian tersebut, maka disarankan agar lebih melengkapi di bagian *text preprocessing*, dimana bagian tersebut lebih ditekankan di bagian *stemming* agar waktu proses yang berjalan semakin cepat.

KATA KUNCI: Algoritma Rabin-Karp, Plagiasi, *hashing*, *string-matching*, *patterm*, *K-gram.*

I. LATAR BELAKANG

Perkembangan dunia teknologi dan informasi yang menjamur di kalangan masyarakat seakan sudah tak dapat terbendung lagi, bahkan perkembangan tersebut akan semakin berkembang seiring dengan penemuan-penemuan baru yang dipopulerkan oleh berbagai kalangan sehingga pada akhirnya akan lebih mempermudah dalam membantu urusan manusia, baik itu dalam segi kehidupan, maupun dari sisi kemanusiaan. Salah satu contoh nyata yang bisa kita rasakan sampai saat ini adalah kemunculan internet yang seakan sudah tak bisa dilepaskan dari kehidupan manusia saat ini. Selain sebagai kebutuhan yang sangat penting, internet juga menjadi gaya hidup manusia di era modernisasi saat ini.

CV. Digimedia Corpindo merupakan salah satu CV yang bergerak di bidang informasi sekaligus sebagai salah satu publisher adsense yang cukup ternama di Indonesia. CV. Digimedia Corpindo merupakan sebuah startup yang termasuk baru di kalangan dunia internet dimana dalam sistem kerjanya menyajikan berbagai informasi dalam beberapa laman website. CV. Digimedia Corpindo mempunyai berbagai macam website yang terjun dalam dunia informasi diantaranya adalah Indowarta.com,Newsth.com, Oketekno.com dan Smeaker.com

Mengacu pada latar belakang tersebut, maka penulis tertarik melakukan penelitian dengan judul “Sistem Pendeteksi Plagiasi Pada Artikel Menggunakan Algoritma Rabin-Karp Di CV. Digimedia Corpindo”. Alasan mengapa penulis menggunakan Algoritma Rabin-Karp dalam menerapkan sistem ini adalah tak lain dan tak bukan karena kualitas yang sudah diberikan dalam berbagai penelitian memanglah menujukkan hasil yang maksimal lantaran Algoritma ini digunakan karena sangat efektif untuk pencarian lebih dari satu kata (multi pattern) (Karp, 1987). Apalagi dalam Algoritma Rabin-Karp tersebut nantinya akan ada sebuah fungsi yakni hashing yang mana. Fungsi hashing menyediakan metode sederhana untuk menghindari perbandingan jumlah karakter yang quadratic di dalam banyak kasus atau situasi. Dengan munculnya sistem ini, nantinya diharapkan pekerjaan penguji calon karyawan baru bisa lebih diringankan lagi, dan untuk selanjutnya yakni masuk ke tahap lebih tinggi yakni penentuan kecocokan pelamar dalam bidang tertentu.

II. METODE

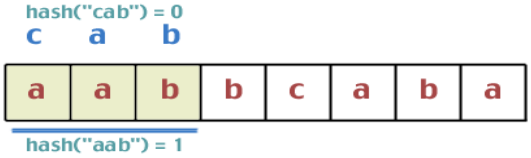
Algoritma Karp-Rabin diciptakan oleh Michael O.Rabin dan Richard M. Karp pada tahun 1978 dengan menggunakan fungsi *hashig* untuk menemukan *pattern* di dalam *string* teks (Fernando, 2009).

Karakteristik Algoritma Karp-Rabin:

1) Menggunakan sebuah fungsi *hashing*
2) Fase *preprocessing* menggunakan kompleksitas waktu O(m)
3) Untuk fase pencarian kompleksitasnya: O(mn)
4) Waktu yang diperlukan O(n+m)

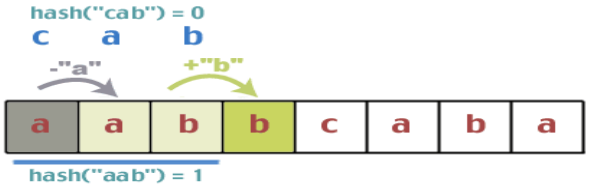
Fungsi *hashing* menyediakan metode sederhana untuk menghindari perbandingan jumlah karekter yang quadratic di dalam banyak kasus dan situasi. Daripada melakukan pemeriksaan terhadap setiap posisi dari teks ketika untuk melakukan pemeriksaan hanya jika teks yang sedang kita proses memilik kemiripan seperti pada *pattern.* Untuk melakukan pengecekan kemiripan anatar dua kata ini digunakan fungsi *hash* (Fernando, 2009).

Berikut ini adalah ilustrasi dari konsep algoritma Rabin-Karp. Diberikan masukan "cab" dan teks "aabbcaba". Fungsi hash yang dipakai misalnya akan menambahkan nilai keterurutan setiap huruf dalam alphabet (a = 1, b = 2, dst) dan melakukan modulo dengan 3. Didapatkan nilai hash dari "cab" adalah 0 dan tiga karakter pertama pada teks yaitu "aab" adalah 1 (Firdaus,2008).



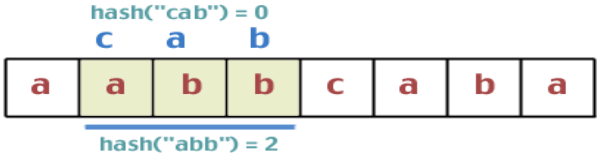
Gambar 2.1 *fingerprint* Awal
(Firdaus, 2008)

Hasil perbandingan ternyata tidak sama, maka substring pada teks akan bergeser satu karakter ke kanan. Algoritma tidak menghitung kembali nilai hash substring. Disinilah dilakukan apa yang disebut rooling hash yaitu mengurangi nilai karakter yang keluar dan menambahkan nilai karakter yang masuk sehingga didapatkan kompleksitas waktu yang relative konstan pada setiap kali pergeseran.



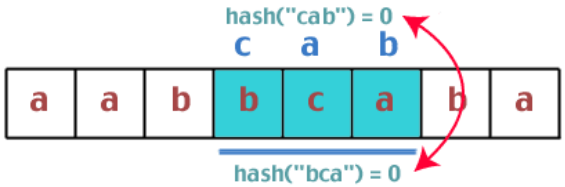
Gambar 2.2 Menggeser *fingerprint*
(Firdaus, 2008)

Setelah pergeseran, didapatkan nilai hash dari fingerprint "abb" (abb = aab – a + b) menjadi dua (2 = 1 – 1 + 2).



Gambar 2.3 Perbandingan kedua
(Firdaus, 2008)

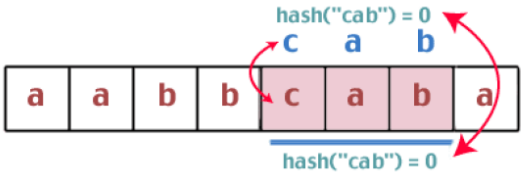
Hasil perbandingan juga tidak sama, maka dilakukan pergeseran. Begitu pula dengan perbandingan ketiga. Pada perbandingan keempat, didapatkan nilai hash yang sama



Gambar 2.4 Perbandingan keempat (nilai hash sama)

(Firdaus, 2008)

Karena nilai hash sama, maka dilakukan perbandingan string karakter per karakter antara "bca" dan "cab". Didapatkan hasil bahwa kedua string tidak sama. Kembali substring bergeser ke kanan.



Gambar 2.5 Perbandingan kelima (srting ditemukan)

(Firdaus, 2008)

Pada perbandingan yang kelima, kedua nilai hash dan karakter pembentuk string sesuai, sehingga solusi ditemukan. Dari hasil perhitungan, kompleksitas waktu yang dibutuhkan adalah O(m+n) dengan m adalah paling panjang string masukan dan n adalah jumlah looping yang dilakukan untuk menemukan solusi. Hasil ini jauh lebih mangkus daripada kompleksitas waktu yang didapat menggunakan algoritma *brute-force* yaitu O(mn) (Firdaus, 2008).

Rabin Karp merepresentasikan setiap karakter ke dalam bentuk desimal digit (digit radix-d) $\Sigma = \{0, 1, 2, 3, \ldots, d\}$, dimana $d = |\Sigma|$. Sehingga didapat masukan string k berturut-turut sebagai perwakilan panjang k desimal. Karakter *string* 31415 sesuai dengan jumlah desimal 31,415. Kemudian pola p dihash menjadi nilai desimal dan *string* direpresentasikan dengan penjumlahan digit-digit angka menggunakan aturan Horner's, misal:

$\{A, B, C, ..., Z\} \rightarrow \{0, 1, 2, ..., 26\}$

- BAN $\rightarrow 1 + 0 + 13 = 14$
- CARD $\rightarrow 2 + 0 + 17 + 3 = 22$

Untuk pola yang panjang dan teks yang besar, algoritma ini menggunakan operasi mod, setelah dikenai operasi mod q, nilainya menjadi lebih kecil dari q, misalnya:

- BAN = 1 + 0 + 13 = 14
  = 14 mod 13 = 1
  = BAN → 1
- CARD = 2 + 0 + 17 + 3 = 22
  = 22 mod 13 = 9
  = CARD → 9

Akan tetapi, tidak semua nilai hash yang cocok berarti polanya cocok. Hal ini sering terjadi pada beberapa kasus, ini disebut *spurious hits*. Kemungkinan terjadinya diantaranya karena:

- Operasi mod terinterfensi oleh keunikan nilai hash (nilai mod q biasanya dipilih bilangan prima sehingga 10q hanya cocok dengan 1 kata komputer).

14 mod 13 = 1

27 mod 13 = 1

- Informasi hilang setelah penjumlahan.

BAN → 1 + 0 + 13 = 14

CAM → 2 + 0 + 12 = 14

Sedangkan *pseudocode* dan rumus matematis uang digunakan adalah sebagai berikut (Cormen, dkk, 2009):

```
RABIN-KARP-MATCHER (T, P, d, q)
n = T.length
m = P.length
h = d^(m-1) mod q
p = 0
t0 = 0
for i = 1 to m // preprocessing
p = (dp + P[i]) mod q
t0 = (dt0 + T[i]) mod q
for s = 0 to n – m
if p == ts // matching
if P[1 .. m] == T[s + 1 .. s + m]
print Pattern occurs with shift s
if s < n – m
ts+1 = (d(ts - T[s + 1] h) + T[s + m + 1]) mod
q
```

Rumus matematis:

t_{s+1} = (d (t_s – T [s + 1] h) + T [s + m + 1] mod q

Dimana:

t_s = Nilai desimal dengan panjang m dari substring

T [s + 1 ..s+m],

untuk s = 0, 1, ..., n – m.

t_{s+1} = Nilai desimal selanjutnya yang dihitung dari ts.

d = Radix desimal (bilangan basis 10).

H = d^{m-1}

n = Panjang teks

m = Panjang pola

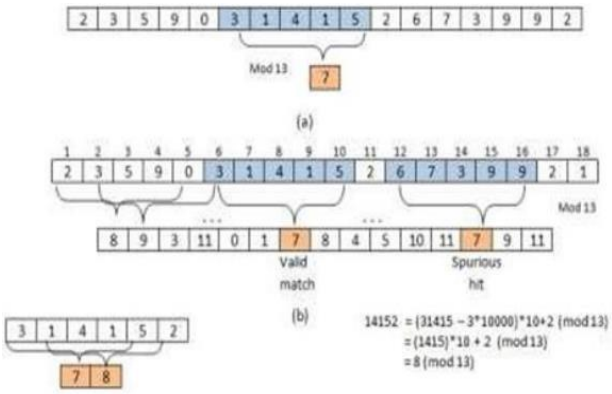
q = Nilai modulo

Pengurangan dengan T [s + 1] * h adalah untuk menghilangkan high-order digit dari t_s, mengalikan hasilnya dengan 10 untuk menggeser satu digit angka ke kiri, dan menambahkan *low-order* digit dengan T [s + m + 1]. Misalnya, jika m = 5 dan ts = 31415, maka kita ingin menghapus high-order digit T [s +1] = 3, masukkan low-order digit baru (anggap T [s +5 +1] = 2) dan modulo = 3 untuk memperoleh

ts +1 = (10 (31415 - 3 * 10000)) + 2 mod 13

= 14152 mod 13

= 8



Gambaran Algoritma Rabin Karp dengan Nilai Mod (Cormen, dkk, 2009)

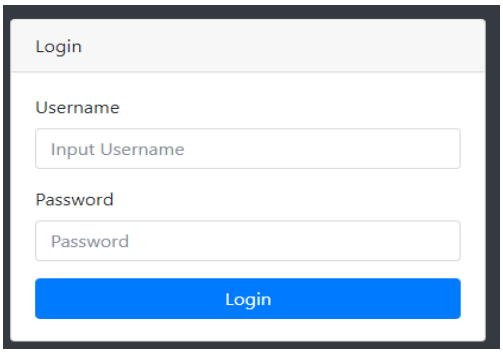
III. HASIL DAN KESIMPULAN

1. Hasil

a. Tampilan Halaman Calon *Content Writer*

Halaman calon *content writer* ini dibuat dengan sederhana dan langsung meminta user untuk memasukkan idenditas diri kemudian artikel yang akan diujikan sesuai dengan tema yang telah ditentukan, setelah menekan *submit button*, aka nada peringatan dimana calon *content writer* diminta untuk mengoreksi lagi hasil kerjaannya, setelah tidak ada yang perlu dikoreksi artikel uji tersebut akan langsung masuk dalam sistem dan langsung di proses dalam halaman seleksi artikel.

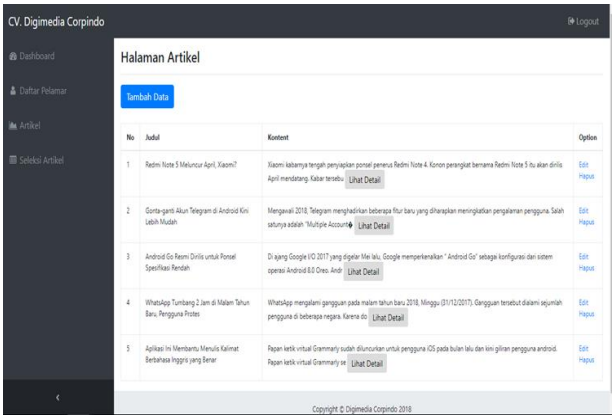
b. Form Login Admin



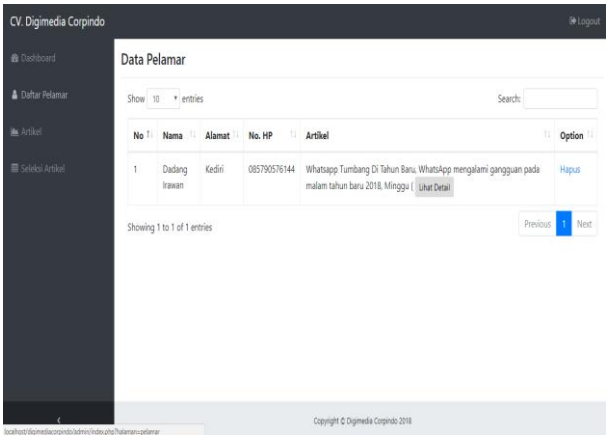
Form login admin merupakan salah satu halaman yang digunakan untuk menguji validasi *username* dan *password* yang telah dibuat, dimana jika *username* dan *password* yang dimasukkan salah maka admin tak akan bisa masuk ke dalam sistem.

c. Halaman Artikel Sumber

Halaman artikel sumber adalah halaman yang berisikan beberapa artikel yang telah dimasukkan *admin* ke dalam sistem, artikel ini sebaiknya dalam 1 tema bahasan yang sama, jadi misalnya tentang teknologi, kesehatan, atau tentang gossip dunia hiburan dan lain sebagainya. Di sini *admin* bisa menambahkan artikel atau menghapus artikel yang tak terpakai.



d. Halaman Artikel Uji

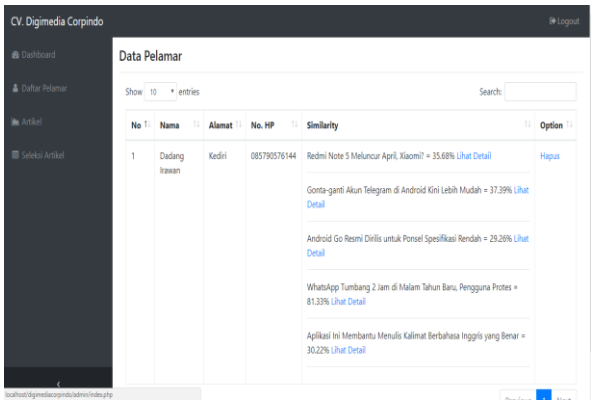


Adalah halaman yang menampilkan hasil pekerjaan yang telah dilakukan oleh calon *content writer* dalam halaman awal, halaman ini menampilkan artikel yang telah dibuat, di sini admin bisa melakukan seleksi dimana calon yang sudah lampau bisa

dihapus agar tidak terlalu banyak menumpuk di dalam *database*.

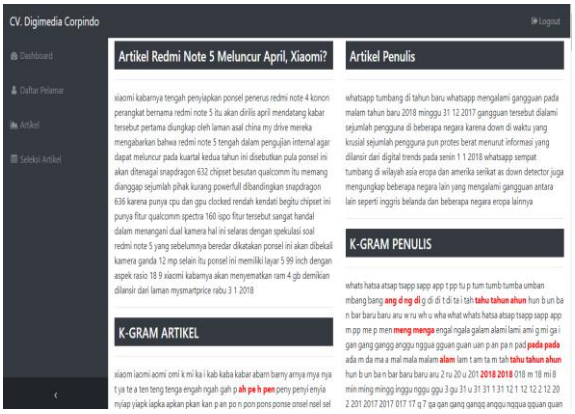
e. Halaman Seleksi Artikel

Halaman seleksi Artikel adalah halaman yang menampilkan hasil dari perhitungan nilai *similarity* artikel yang telah dikerjakan oleh calon *content writer* di bandingkan dengan artikel sumber yang telah diinputkan oleh *admin*. Dalam penampilannya halaman ini menunjukkan tingkat presentase yang diberikan dari masing-masing judul artikel sumber. Bisa jadi dalam prakteknya nanti ada yang cenderung bernilai besar dan cenderung bernilai kecil.



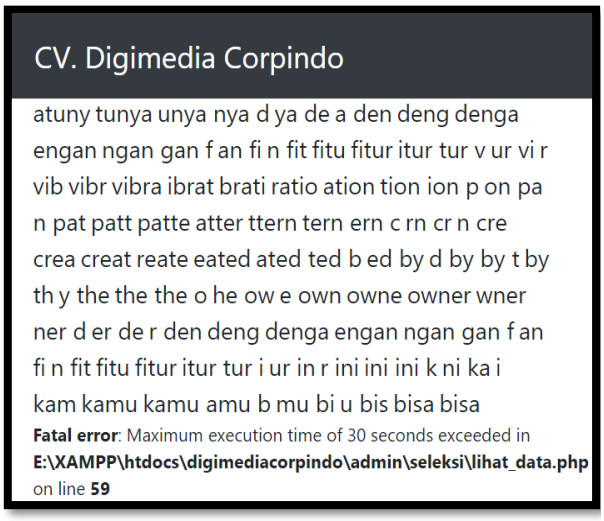
f. Halaman Detail Seleksi Artikel

Halaman detail seleksi artikel ini berisikan perhitungan hasil parsing k gram sampai dengan penilaian hashing yang telah dilakukan oleh sistem terhadap artikel yang di uji, di lajur sebelah kiri berisikan artikel sumber yang diinputkan oleh admin, sedang di lajur kanan berisi artikel dari calon *content writer*.

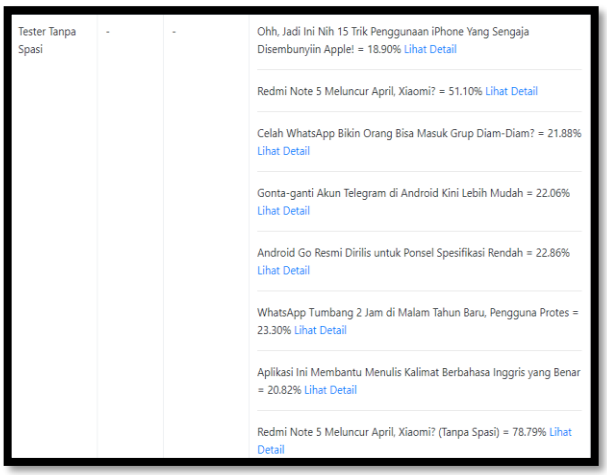


g. Hasil Uji Coba

a) Hasil uji coba yang dilakukan yakni dengan membandingkan artikel *inputan* baru yang disematkan oleh *admin* dengan jumlah kata sekitar seribu kata, kemudian setelah dilakukan dengan input artikel dari pelamar sistem bisa mendeteksi plagiasi yang ada dengan menampilkan presentase *similarity* secara gamblang. Namun dari sisi melihat secara detail kesamaan tersebut sistem menngalami kendala yakni *apache server* tak mampu menampilkan parsing *k-grams* sampai hasil *hashing* lantaran *loading* yang terlalu lama yakni lebih dari 30 detik, penampakan error tersebut bisa dilihat dalam gambar yang mana hal ini masih belum bisa diatasi oleh penulis.



b) Uji coba yang dilakukan selanjutnya yakni dengan menghilangkan karakter spasi dalam setiap artikel sumber maupun artikel pengujian, hal ini dilakukan mengingat ada masukan dari dosen pembimbing 1 yang meminta untuk melakukan ujicoba dengan menghilangkan karakter spasi dalam artikel. Lantaran sistem belum bisa menghilangkan secara otomatis, maka penulis melakukan penghilangan spasi secara manual sehingga menghasilkan tingkat *similarity* yang tercantum dalam gambar berikut



Dalam gambar tersebut bisa disimpulkan bahwa penghilangan spasi cukup mempengaruhi tingkat *similarity* kesamaan artikel, dimana jika dibandingkan dengan yang menggunakan spasi hasilnya adalah 51 % sedangkan jika spasi dihilangkan tingkat kesamaanya bertambah yakni menjadi 78,79 %.

2. Kesimpulan

a) Algoritma *Rabin-Karp* mampu digunakan untuk mendeteksi tingkat kemiripan (*similarity*) artikel yang dibuat oleh calon *content writer* dibandingkan dengan artikel sumber yang dimasukkan oleh penguji. Caranya yakni dimulai dengan *text preprocessing*, parsing *k-grams*, kemudian dilakukan perhitungan *hashing* dan *string mathing* sehingga menghasilkan nilai *similarity*.

b) Dalam penentuan tingkat s*imilarity*, banyaknya kata, karakter dan ada atau tidaknya spasi dalam pengujian turut mempengaruhi nilai *similarity* dan juga, jumlah *parsing k-grams* juga berpengaruh dalam tingkat *similarity*

IV. DAFTAR PUSTAKA

Cormen, Thomas H, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. 2009. *Introduction to Algorithms*. MIT Press, USA.

Fernando, Hary. 2009. *Perbandingan dan Pengujian Beberapa Algoritma*

Rabin-Karp. Progam Studi Teknik Informatika, Institut Teknologi Bandung (ITB). Bandung.

Firdaus, Hari Bagus. 2008. *Deteksi Dokumen Menggunakan Algoritma Rabin-Karp*. Progam Studi Teknik Informatika Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung (ITB). Bandung.

Karp, Richart M., Rabin, Michael O. 1987. *Efficient Randomzed Pattern-Matching Algoritms*