AImplementasi Algoritma Rabin Karp untuk Mendeteksi Kemiripan Dokumen STMIK Bandung (Studi Kasus : Abstrak Jurnal)

Rizky[1], Siti Yuliyanti[2]

[1,2] *STMIK Bandung*
Jalan Cikutra 113. Bandung, Jawa Barat Indonesia
[1] *jstrizky@gmail.com,* [2]*sitiyuliyanti.stmikbandung@gmail.com*

Intisari—Dokumen merupakan bentuk yang dapat dijadikan alat bukti, sebagai karya yang telah dibuat, dimasa digital dokumen sangat rawan tindak penyalinan dokumen secara langsung di kalangan *civitas* akademi, dikarenakan akses mudah untuk mendapatkan dokumen, Kemiripan dokumen merupakan salah satu alternatif yang dapat digunakan untuk mengetahui penjiplakan dalam dokumen. Penjiplakan adalah mencuri hasil karya orang lain dan mengakuinya sebagai karya sendiri, tanpa menyertakan referensi ke sumber aslinya. Algoritma Rabin Karp merupakan algoritma penggalian data dalam pencarian pola *text*, Algoritma ini dapat membantu pencarian kemiripan dokumen dengan penerapan metode *rolling hash* menggunakan k-gram, t*ext mining* (*case folding, tokenizing, filtering, stemming*) dan perhitungan nilai similarity (*dice's similarity coeficients, modulus)*.

Kata kunci— **Rabin Karp, *Rolling Hash*, Kemiripan, *k-gram***

Abstract*— *Document is a form that can be used as proof tool, as the work that has been made, the future of digital documents are vulnerable to the copying of documents directly among the civitas academy, because of the easy access to obtain documents, similarities Document is one of the alternatives that can be used to identify cribbing in the document. similarity is stealing the work of others and admits it as a work of its own, without including references to the original source. Rabin Karp algorithm is a data digging algorithm in the search for text patterns, this algorithm can help search similarity of documents with the application of Rolling Hash method using k-gram, text mining (case folding, tokenizing, filtering, stemming) and the calculation of the similarity value (dice's similarity coefficient's, modulus).

Keywords— ***Rabin Karp, Rolling Hash, Simillarity, k-gram***.

I. PENDAHULUAN

Kemajuan teknologi informasi, beberapa pekerjaan dapat dikerjakan menjadi lebih mudah dengan bantuan teknologi komputer, seperti mengolah data. Data yang diolah dengan bantuan komputer akan terasa lebih efektif dan efisien sehingga menghasilkan informasi yang diinginkan. Dibalik kemudahan yang didapat seperti menyalin berkas digital, kecenderungan hal tersebut dapat menimbulkan dampak negatif untuk kepentingan kelompok maupun perorangan, salah satunya kemiripan dokumen. Di dalam lingkungan *civitas* akademik kemiripan dokumen rawan terjadi seperti, mahasiswa kebingungan mencari atau membuat jurnal sendiri, maka jalan pintas yang ditempuh dapat berupa tindakan penyalinan dokumen untuk mempersingkat waktu.

Suatu tindakan yang dapat mencari tindakan penyalinan dokumen ini adalah dengan melakukan komparasi terhadap jurnal tersebut. Komparasi dilakukan dengan menghitung tingkat persentase kemiripan setiap kata di dalam jurnal.

Pada penelitian ini penyusun menggunakan beberapa metode bagaimana pengaruh variasi nilai k-gram, pada persentase kemiripan dokumen.

Algoritma Rabin Karp ditemukan oleh Michael O. Rabin dan Richard M. Karp. Algoritma ini menggunakan metode *hash* dalam mencari suatu kata. Teori ini jarang digunakan untuk mencari kata tunggal, namun cukup penting dan sangat efektif bila digunakan untuk pencarian jamak (komparasi). K-Grams adalah rangkaian *terms* dengan panjang K, kebanyakan yang digunakan sebagai *terms* adalah kata**Error! Reference source not found.**. *Text Mining* adalah salah satu bidang khusus dari data *mining*. *Text Mining* sebagai suatu proses (*case folding, tokenizing, filtering,* dan *stemming*) menggali informasi di mana seorang *user* berinteraksi dengan sekumpulan dokumen menggunakan *tools* analisis yang merupakan komponen-komponen dalam *data mining***Error! Reference source not found.**.

Oleh karena itu penyusun merancang sebuah aplikasi pengelola kemiripan dokumen sebagai penelitian, ini untuk mengukur persentase kemiripan dokumen.

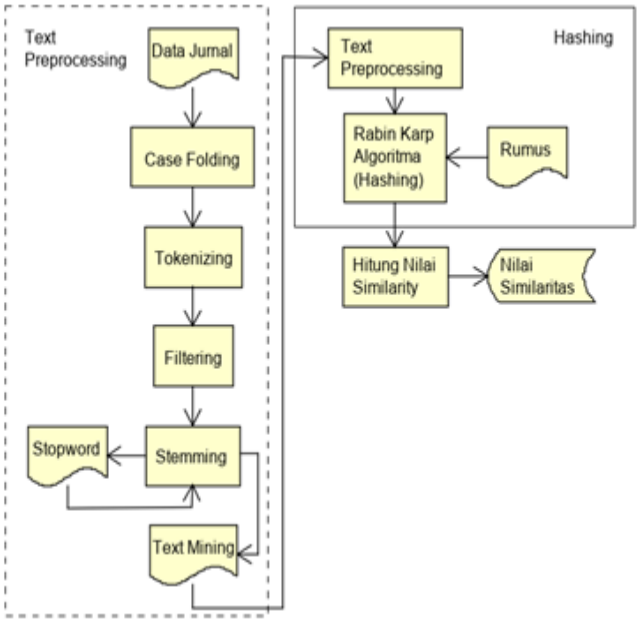
II. METODOLOGI PENELITIAN

Pada tahap ini dibahas tentang proses komparasi kemiripan dokumen yaitu :

1. Melakukan ekstraksi dokumen (*text mining*).
2. Proses pembobotan dokumen menggunakan algoritma rabin karp dengan metode *rolling hash*.
3. Perhitungan nilai kesamaan dokumen.

ISSN: 2337-4055

Berdasarkan latar belakang dan tujuan dalam penelitian ini penyusun menggunakan beberapa proses (ekstraksi dokumen) sebelum melakukan komparasi dokumen, sebagaimana diilustrasikan pada gambar 1.



Gambar 1. Kerangka penelitian.

III. HASIL DAN PEMBAHASAN

Pembahasan pada penelitian ini meliputi beberapa tahapan sebgaimana diilustrasikan pada gambar 1.

A. Pengumpulan dataset

Pada penelitian ini, data diperoleh dari kumpulan dokumen jurnal STMIK Bandung yang diambil dari bagian abstrak. Dataset yang digunakan terdiri dari data latih dan data uji untuk mengevaluasi hasil dari model yang dibangun.

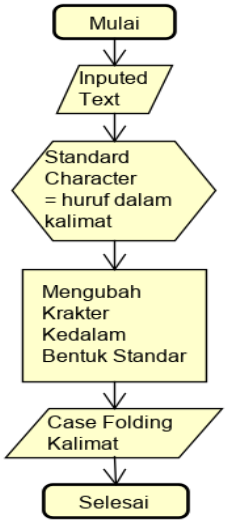
B. Pre-processing

Tujuan dari *text mining* adalah untuk mendapatkan informasi yang berguna dari sekumpulan dokumen**Error! Reference source not found.**. Dalam penelitian ini penyusun melakukan beberapa tahapan proses *text mining* yaitu :

1) *Case folding*, mengkonversi keseluruhan teks dalam dokumen menjadi suatu bentuk standar**Error! Reference source not found.**, dan berikut script casefolding :

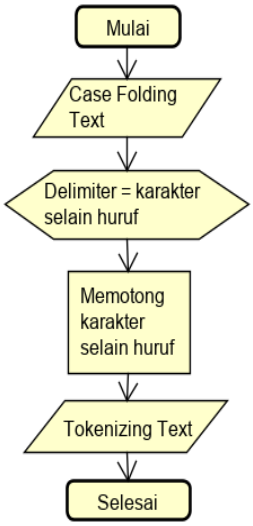
```
function casefolding($kalimat){
$case = strtolower($kalimat);
return $case;
}
```

Adapun *flowchart* dari funsgsi tersebut seperti gambar 2.



Gambar *2.* Flowchart Case Folding

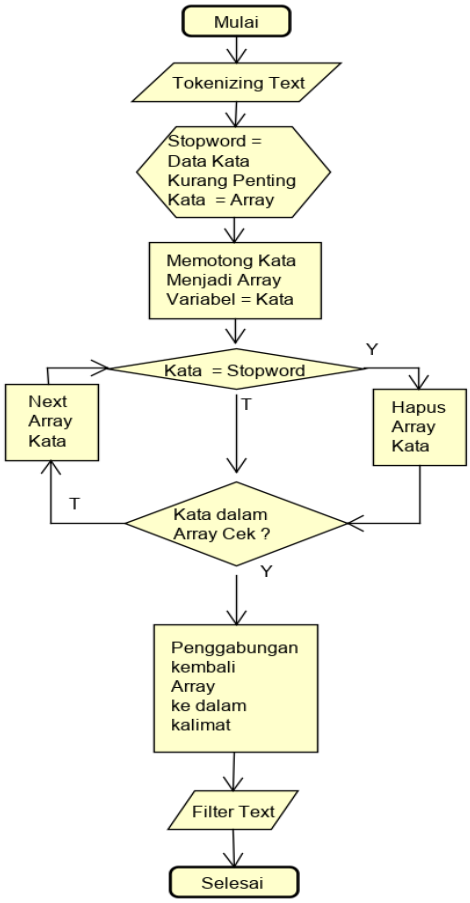
2) *Tokenizing*, pemotongan *string* masukkan karakter selain huruf dihilangkan dan dianggap *delimiter*, *Delimiter* adalah urutan satu karakter atau lebih yang dipakai untuk membatasi atau memisahkan data yang disajikan dalam kalimat. Salah satu contoh dari *delimiter* adalah tanda koma, titik koma atau titik dua**Error! Reference source not found.**, dan berikut gambar 3 *flowchart* dan ***script tokenizing***.



Gambar *3* Flowchart Tokenizing

```
function tokenizing($kalimat){
//$symbo = huruf delimiter
$symbol = array();
$kalimat = str_replace($symbol, "",
$kalimat);
$karakter = preg_replace("/[^a-z]/", " ",
$kalimat);
$token = explode(" ", $karakter);
return $token;
}
```

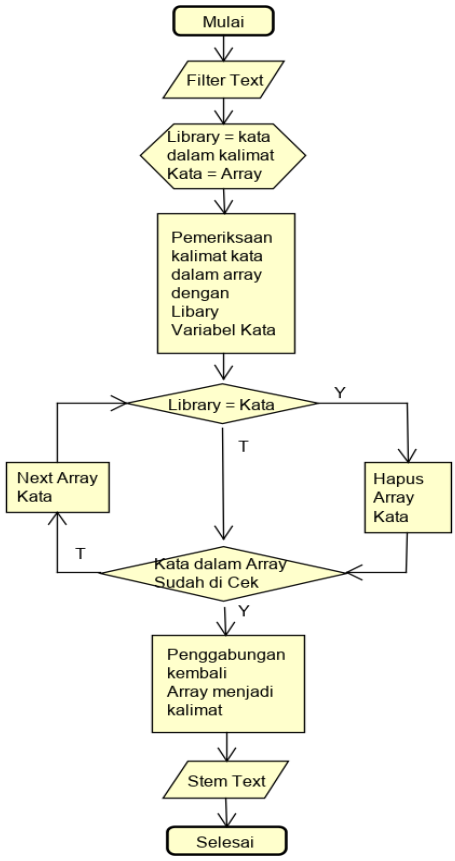
3) *Filtering*, mengambil kata-kata penting dari hasil *tokenizing*. *Filtering* dapat menggunakan algoritma *stoplist*(membuang kata yang kurang penting) atau *wordlist* (menyimpan kata penting)**Error! Reference source not found.**, berikut gambar 4 *flowcart* dan ***script filtering***.



Gambar *4* Flowchart Filtering

```
function filtering($kalimat){
$stopwords = array();
for ($i = 0; $i <
count($stopwords); $i++) {
for ($x = 0; $x <
count($kalimat); $x++) {
if ($kalimat[$x] ==
$stopwords[$i]) {
$kalimat[$x] = null;
  }
 }
}
return $kalimat;
}
```

4) *Stemming*, mencari kata dasar kata dari tiap kata hasil *filtering*. Pada tahap ini dilakukan proses pengembalian berbagai bentukan kata ke dalam suatu representasi yang sama**Error! Reference source not found.**, dan berikut gambar 5 *flowchart* dan ***script stemming***.



Gambar *5* Flowchart Stemming

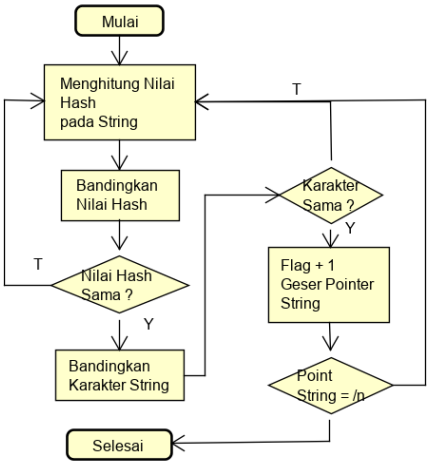
Script stemming

```
function stemming($kalimat){
// create stemmer
//cukup dijalankan sekali saja,
biasanya didaftarkan di service
container
$stemmerFactory = new
\Sastrawi\Stemmer\StemmerFactory();
$stemmer  = $stemmerFactory-
>createStemmer();
$sentence = $kalimat;
$output   = $stemmer-
>stem($sentence);
return $output;
```

C. Algoritma Rabin Karp

Pada dasarnya, algoritma Rabin-Karp akan membandingkan nilai *hash* dari *string* masukan dan *substring* pada teks. Apabila sama, maka akan dilakukan perbandingan sekali lagi terhadap karakter-karakternya. Apabila tidak sama, maka *substring* akan bergeser ke kanan. Kunci utama performa algoritma ini adalah perhitungan yang efisien terhadap nilai *hash substring* pada saat penggeseran dilakukan**Error! Reference source not found.**.

ISSN: 2337-4055



Gambar 6 Flowchart Algoritma Rabin Karp

D. Hashing

Hashing adalah suatu cara untuk mentransformasi sebuah *string* menjadi suatu nilai yang unik dengan panjang tertentu *(fixed-length)* yang berfungsi sebagai penanda *string* tersebut. Fungsi untuk menghasilkan nilai ini disebut fungsi *hash*, sedangkan nilai yang dihasilkan disebut nilai *hash* [1]. Berikut contoh perubahan jenis data keturunan *alphabet* ke dalam bilangan bulat (a = 97, b= 98, dst).Pada laporan yang Penyusun buat urutan *alphabet* menggunakan karakter *code American Standard Code for Information Interchange (ASCII)*. Berikut metode yang ada di dalam *hashing*:

- K-Gram
  K-Gram adalah rangkaian *terms* dengan panjang K. Kebanyakan yang digunakan sebagai *terms* adalah kata. K-Gram merupakan sebuah metode yang diaplikasikan untuk pembangkitan kata atau karakter. Metode K-Gram ini digunakan untuk mengambil potongan-potongan karakter huruf sejumlah k dari sebuah kata yang secara kontinuitas dibaca dari teks sumber hingga akhir dari dokumen [3].
- Modulus
  Sebagai pembagi nilai *hash text* agar memiliki keunikan tersendiri

Pada dasarnya penelitian ini menerapkan metode *rolling hash* untuk pencarian pola *text* untuk membantu pendeteksian lebih akurat

Metode *Rolling Hash* dapat memecahkan masalah pada fungsi *hash* yang terdapat di algoritma Rabin-Karp. Karena menggunakan *rolling hash*, nilai *hash* yang bisa diperbarui dengan rumus tertentu berdasarkan karakter yang dibuang dan ditambahkan [2].

ASCII printable characters

DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo
32	20h	espacio	64	40h	@	96	60h	`
33	21h	!	65	41h	A	97	61h	a
34	22h	"	66	42h	B	98	62h	b
35	23h	#	67	43h	C	99	63h	c
36	24h	$	68	44h	D	100	64h	d
37	25h	%	69	45h	E	101	65h	e
38	26h	&	70	46h	F	102	66h	f
39	27h	'	71	47h	G	103	67h	g
40	28h	(	72	48h	H	104	68h	h
41	29h	)	73	49h	I	105	69h	i
42	2Ah	*	74	4Ah	J	106	6Ah	j
43	2Bh	+	75	4Bh	K	107	6Bh	k
44	2Ch	,	76	4Ch	L	108	6Ch	l
45	2Dh	-	77	4Dh	M	109	6Dh	m
46	2Eh	.	78	4Eh	N	110	6Eh	n
47	2Fh	/	79	4Fh	O	111	6Fh	o
48	30h	0	80	50h	P	112	70h	p
49	31h	1	81	51h	Q	113	71h	q
50	32h	2	82	52h	R	114	72h	r
51	33h	3	83	53h	S	115	73h	s
52	34h	4	84	54h	T	116	74h	t
53	35h	5	85	55h	U	117	75h	u
54	36h	6	86	56h	V	118	76h	v
55	37h	7	87	57h	W	119	77h	w
56	38h	8	88	58h	X	120	78h	x
57	39h	9	89	59h	Y	121	79h	y
58	3Ah	:	90	5Ah	Z	122	7Ah	z
59	3Bh	;	91	5Bh	[	123	7Bh	{
60	3Ch	<	92	5Ch	\	124	7Ch	|
61	3Dh	=	93	5Dh	]	125	7Dh	}
62	3Eh	>	94	5Eh	^	126	7Eh	~
63	3Fh	?	95	5Fh	_			

theASCIIcode.com.ar

Gambar 1 Tabel ASCII0

Berikut rumus matematis**Error! Reference source not found.**, dan *script rollinghash* pada persamaan 1.

$$ts+1 = (d\,(ts - T[s+1]\,h) + T[s+m+1] \bmod q \qquad (1)$$

ts : nilai *decimal* dengan panjang m dari *substring* T[s+1+..s+m], untuk s= 0,1,..,n,-m
ts+1 : nilai *decimal* selanjutnya yang dihitung dari ts
d : *radix decimal (modulus)*
h : d^{m-1}
n : panjang teks
m : panjang pola
q : nilai modulus

Script rollinghash

```
function rollinghash($value,
$igrams, $basis, $modulus){
$base = $basis;
$mod = $modulus;
$panjangkar = strlen($value);
$rhash = 0;
 for ($i = 0; $i < $panjangkar; $i++
$ascii = ord($value[$i]);
$rhash += $ascii * pow($base,
$panjangkar - ($i - 1));
$rhash = $rhash % $mod;
}
return $rhash;
}
```

E. Pembobotan Nilai Similarity

Pada proses ini dilakukan pembobotan dokumen dengan metode *hashing* ,implementasi algoritma rabin karp dan penilaian *similarity* setelah proses ekstraksi dokumen selesai. Pada pembobotan dokumen ini perhitungan nilai *similarity*;

Menghitung nilai *similarity* dari dokumen digunakan *Dice's Similarity Coeficients* dengan cara menghitung nilai dari jumlah K-Gram yang digunakan pada kedua dokumen yang diuji, sedangkan dokumen *fingerprint* didapat dari jumlah nilai K-Gram yang sama. Nilai *Similarity* tersebut dapat dihitung dengan menggunakan rumusan pada persamaan 2 **Error! Reference source not found.**.

$$S = \frac{KC}{(A+B)} \qquad (2)$$

S : Nilai *Similarity*
K : *Dice's Similarity Coeficiets*
C : Jumlah K-Grams yang sama pada dokumen 1 dan dokumen 2
A : Jumlah K-Grams dokumen 1
B : Jumlah K-Grams dokumen 2

Script similarity

```
function similarity($fprint,
$uhash, $ahash, $dice){
return floatval((($dice *
count($fprint) / (count($uhash) +
count($ahash))) * 100));
}
```

Berikut hasil pengujian dengan menerapkan rumus yang berbeda pada pengujian yang dokumen yang sama:Merah menandakan Pengaruh K-Gram, Biru menandakan Pengaruh Basis, Hijau menandakan Pengaruh Modulus. Hasil Pengujian terlampir pada Tabel 1 sampai dengan Tabel 8.

Tabel 1
Pengujian Rumus No 1

No Dokumen		Kemiripan Judul	Kemiripan Abstrak
No 1	No 2	14.55%	49.84%
No 1	No 3	0.00%	58.42%
No 1	No 4	17.31%	69.47%
No 1	No 5	5.26%	51.94%
No 1	No 6	22.22%	49.10%
No 1	No 7	15.79%	52.58%

Tabel 2
Pengujian Rumus No 2

No Dokumen		Kemiripan Judul	Kemiripan Abstrak
No 1	No 2	14.55%	49.84%
No 1	No 3	0.00%	58.42%
No 1	No 4	17.31%	69.47%
No 1	No 5	5.26%	51.94%
No 1	No 6	22.22%	49.10%
No 1	No 7	15.79%	52.58%

Tabel *3*
Pengujian Rumus No 3

No Dokumen		Kemiripan Judul	Kemiripan Abstrak
No 1	No 2	14.55%	45.53%
No 1	No 3	0.00%	47.73%
No 1	No 4	13.36%	58.95%
No 1	No 5	5.26%	43.28%
No 1	No 6	20.00%	43.84%
No 1	No 7	15.79%	38.70%

Tabel 43
Pengujian Rumus No 4

No Dokumen		Kemiripan Judul	Kemiripan Abstrak
No 1	No 2	14.55%	45.53%
No 1	No 3	0.00%	47.73%
No 1	No 4	13.36%	58.95%
No 1	No 5	5.26%	43.28%
No 1	No 6	20.00%	43.84%
No 1	No 7	15.79%	38.70%

Tabel *5*
Pengujian Rumus No 5

No Dokumen		**Kemiripan Judul**	**Kemiripan Abstrak**
No 1	No 2	0.00%	10.52%
No 1	No 3	0.00%	12.37%
No 1	No 4	4.17%	21.30%
No 1	No 5	0.00%	13.66%
No 1	No 6	4.88%	12.08%
No 1	No 7	3.77%	11.64%

Tabel *6*
Pengujian Rumus No 6

No Dokumen		Kemiripan Judul	Kemiripan Abstrak
No 1	No 2	0.00%	5.50%
No 1	No 3	0.00%	8.98%
No 1	No 4	4.17%	16.93%
No 1	No 5	0.00%	9.76%
No 1	No 6	4.88%	9.21%
No 1	No 7	3.77%	9.00%

Tabel *7*
Pengujian Rumus No 7

No Dokumen		Kemiripan Judul	Kemiripan Abstrak
No 1	No 2	1.96%	9.87%
No 1	No 3	0.00%	13.11%
No 1	No 4	6.26%	17.46%
No 1	No 5	2.94%	14.56%
No 1	No 6	4.88%	13.60%
No 1	No 7	5.66%	11.95%

Tabel 4
Pengujian Rumus No 8

No Dokumen		Kemiripan Judul	Kemiripan Abstrak
No 1	No 2	0.00%	5.66%
No 1	No 3	0.00%	9.86%
No 1	No 4	4.17%	17.59%
No 1	No 5	0.00%	10.21%
No 1	No 6	4.88%	7.85%

ISSN: 2337-4055

No 1	No 7	3.77%	8.22%

Dari hasil tersebut k-grams, basis, dan modulus berpengaruh terhadap perhitungan similarity.

IV. KESIMPULAN

Penelitian yang dilakukan penyusun dapat disimpulkan bahwa :

1. Nilai k-gram sangat berpengaruh terhadap nilai persentase suatu kemiripan dokumen.
2. Semakin tinggi nilai k-gram, maka akan semakin rendah nilai persentase dokumen tersebut.
3. Pemakaian nilai tinggi disarankan untuk kalimat yang cukup banyak □isbanding dengan kalimat yang sedikit.
4. Basis dan modulus tidak terlalu signifikan terhadap persentase nilai kemiripan dokumen.
5. Penggunaan nilai basis dan modulus yang tepat pada dokumen dapat membantu menurunkan atau meningkatkan nilai persentase dokumen.

UCAPAN TERIMA KASIH

Penyusun juga ucapkan terimakasih kepada keluarga, rekan-rekan dan seluruh civitas akademika STMIK abndung atas bantuan dan kontribusinya dalam penyusunan jurnal ini yang tidak dapat penyusun sebutkan satu persatu. Akhir kata semoga jurnal ini bermanfaat dan dapat dijadikan salah satu refernsi untuk pengembangan penelitian khususnya dibidang teks mining.

REFERENSI

[1] W. Faizzani and F. H. R., "Sistem FAQ Konsultasi Dokter Gigi Menggunakan Algoritma Winnowing dan Synonym Replacement," *Jurnal SimanteC,* vol. 4, no. 2, pp. 105 - 114, 2014.

[2] J. Priambodo, "Pendeteksian Plagiarisme menggunakan Algoritma Rabin-Karp Dengan Metode Rolling Hash," *Jurnal Informatika Universitas Pamulang,* vol. 3, no. 1, pp. 39 – 45, 2018.

[3] D. N. Sari and D. P. Utomo, "Implementasi Algoritma Rabin-Karp pada Pencarian Quotes Tokoh Terkenal," *Pelita Informatika : Informasi dan Informatika,* vol. 9, no. 1, pp. 33 – 45 , 2020.

[4] S. Yuliyanti, T. Djatna and H. Sukoco, "Sentiment Mining of Community Development Program," *TELKOMNIKA,* vol. 15, no. 4, pp. 1858 – 1864, 2017.

[5] S. Suhada and S. Bahri, "Implementasi Algoritma Rabin Karp Dan Stemming Najief Andriani Untuk Deteksi Plagiarisme Dokumen," *SWABUMI ,* vol. 5, no. 1, pp. 84 – 89, 2017.

[6] Salmuasih and S. Andi, "Implementasi Algoritma Rabin Karp untuk Pendeteksian Plagiat Dokumen Teks Menggunakan Konsep Similarity," in *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*, Yogyakarta, 2013.

[7] Theasciicode.com.ar.(22 November 2019) diambil kembali dari https://theasciicode.com.ar/ascii-printable-characters/capital-letter-z-uppercase-ascii-code-90.html.