

## **Sistem Pencegahan *Plagiarism* Tugas Akhir Menggunakan Algoritma Rabin-Karp (Studi Kasus: Sekolah Tinggi Teknik Payakumbuh)**

**Siti Ramadhani**

Program Studi Sistem Informasi Fakultas Ilmu Komputer Universitas Lancang Kuning  
Jl. Yos Sudarso KM. 8 Rumbai, Pekanbaru, Riau, telp. 0811 753 2015  
e-mail: tik\_siti@yahoo.com

### **Abstrak**

*Plagiarisme merupakan tindakan yang dilarang dan termasuk pelanggaran Hak Cipta. Berdasarkan Peraturan Menteri Pendidikan Nasional Republik Indonesia Tahun 2010, Para pelaku plagiarism dapat dikenakan sanksi berupa teguran hingga pembatalan ijazah mahasiswa yang telah lulus. Upaya yang ditempuh untuk mencegah terjadinya plagiarisme adalah dengan mendeteksi terjadinya plagiarisme menggunakan algoritma pencocokan string seperti Boyer Moore, Brute Forte, Knuth Moris Pratt (KMP), dan Rabin-Karp. Algoritma yang paling efektif adalah algoritma Rabin-Karp yang menggunakan fungsi k-gram untuk menentukan panjang potongan pola string dan fungsi hash untuk mengubah setiap pola string menjadi nilai numerik kemudian membandingkan tingkat kesamaan source teks dengan teks pembanding. Studi kasus dilakukan di Sekolah Tinggi Teknologi Payakumbuh melalui pengambilan sampel data berupa proposal judul yang diajukan mahasiswa Teknik Komputer. Dalam upaya pendeteksian plagiarime dilakukan dengan membandingkan potongan sampel source dokumen dan potongan sampel dokumen pembanding mewakili proposal yang lengkap untuk menerapkan analisa meggunakan algoritma Rabin-Karp. Proses analisa berlangsung 4 tahap yaitu : preprocessing, tokenizing k-gram, fingerprint dan perhitungan similarity. Hasil pengujian berupa persentase nilai similariy antara teks A dan B serta teks B dan C menggunakan nilai k-gram 3, 4 dan 5 serta penggunaan basis 3, 5 dan 11 yang mana didapatkan nilai perbandingan tertinggi pada nilai k-gram=3 dan basis=3 yaitu 78.16% untuk teks A dan B, 78,79% untuk teks A dan C.*

**Kata kunci:** Algoritma Rabin-Karp, Hashing, K-gram, Fingerprint, Similarity

### **Abstract**

*Plagiarism is prohibited and a violation of copyright. Based on the Regulation of the Minister of National Education, Republic of Indonesia, 2010, whom who done plagiarism may be the subject of punishment in the form of a reprimand to cancellation of diploma students who has graduated. The efforts that can be taken to prevent plagiarism is to detect the occurrence of plagiarism using string matching algorithms such as Boyer Moore, Brute Forte, Knuth Morris Pratt (KMP), and Rabin-Karp. The most effective algorithm Rabin-Karp algorithm that uses the functions k-gram to determine the length of the string and the pattern pieces hash function to change any pattern string into a numeric value and then comparing the level of similarity source text with the comparison text. The Case studies in Payakumbuh College Technology is done through sampling data in the form of proposals title which are submitted by Computer Engineering student. To detect the plagiarism is done by comparing the sample pieces source documents and comparison sample pieces document. Testing process on 4 stages, they are: preprocessing, tokenizing k-gram, fingerprint and similarity calculation. The research result is the similariy value precentage on comparison of text A and text B also text A and text C in k-gram value is 3, 4 and 5 then basis value on 3, 5 and 11 which better done on k-gram value=3*

---

and basic value 3 they are 78.16% for comparison of text A and B then 78,79 for comparison of text A and text C.

**Keywords:** Algoritma Rabin-Karp, Hashing, K-gram, Fingerprint, Similarity

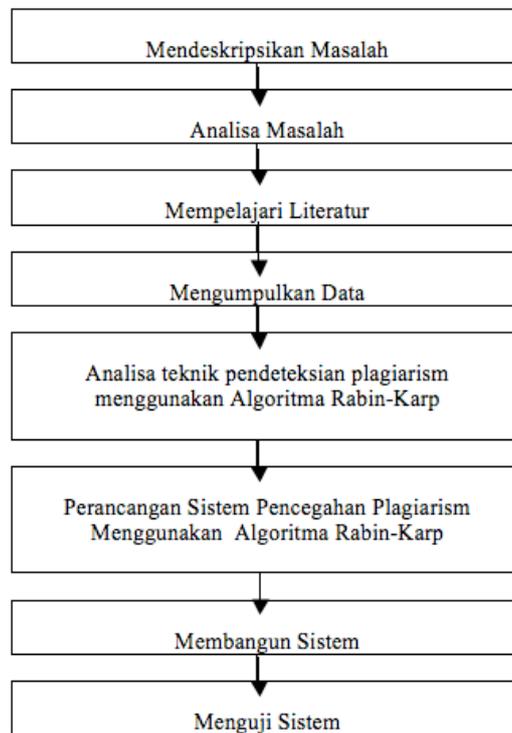
## 1. Pendahuluan

Kemajuan teknologi dan kemudahan akses ilmu pengetahuan melalui internet ternyata tidak hanya memberikan dampak positif terhadap dunia pendidikan tetapi juga memberikan dampak negatif. Salah satu dampak negatifnya adalah sering terjadinya plagiarisme yang merupakan tindakan yang dilarang dan termasuk pelanggaran Hak Cipta. [1] Dian Novian, *et al.*(2012) menuliskan bahwa *plagiarism* atau penjiplakan menurut Kamus Besar Bahasa Indonesia (KBBI) berarti menggambar atau menulis garis-garis gambaran atau tulisan yang telah tersedia (dengan menempelkan kertas kosong pada gambar atau tulisan yang akan ditiru), mencontoh atau meniru tulisan atau pekerjaan orang lain, mencuri karangan orang lain dan mengakui sebagai karangan sendiri, mengutip karangan orang lain tanpa seizin penulisnya. Tindakan ini bisa dipicu oleh tuntutan akademik yang mengharuskan mahasiswa menulis sebagai syarat untuk meraih gelar keilmuan yang sedang ia tempuh dan kurangnya ide membuat mereka terjebak dalam *plagiarism*. Pencegahan dan penanggulangan *plagiarism* di Perguruan Tinggi merupakan hal yang harus dihindarkan dan diatur berdasarkan Peraturan Menteri Pendidikan Nasional Republik Indonesia Tahun 2010. Para pelaku plagiarisme dapat dikenakan sanksi berupa teguran hingga pembatalan ijazah mahasiswa yang telah lulus. Pendeteksian plagiarisme dapat dilakukan melalui pencocokan derajat kesamaan tugas akhir menggunakan salah satu algoritma pencocokan *string* seperti algoritma Boyer Moore, Brute Force, Knuth Morris Pratt (KMP), dan Rabin-Karp. Algoritma pencocokan *string* ini sering digunakan di beberapa aplikasi editor teks seperti : *Query database*, *Bioinformatics* dan *kimiaawi*, pola pencocokan *string*, *Digital Libraries* dan *Search Engine*. Algoritma pencocokan *string* yang paling efektif adalah algoritma Rabin-Karp [2]. Algoritma ini mencoba untuk mempercepat pencocokan *string* dengan menguji persamaan pola *substring* teks menggunakan *Hash Function*. Fungsi Hash adalah fungsi yang mengubah setiap *string* menjadi nilai numerik untuk membandingkan tingkat kesamaan *string* dan *pattern* dalam artikel atau *script* pemrograman [3]. Sistem deteksi *plagiarisme* pernah dilakukan oleh peneliti [4] menggunakan metode *vector space model* untuk deteksi kemiripan dokumen jurnal teknik informatika dan sistem informasi. Tahapan *preprocessing*nya meliputi tokenisasi; penghapusan *stopword* dan *stemming*; pembobotan dan *cosine similarity*; dan peneliti [5] menggunakan algoritma *winnowing* sebagai pembentuk *fingerprint* dokumen sehingga penjiplakan dokumen baik dilakukan di awal maupun di akhir dokumen akan tetap dapat dideteksi. Tahapan *preprocessing* menggunakan *stemming* dan metode *Jaccard Coefficient* sebagai penghitung nilai *similarity*. Upaya yang dilakukan untuk dapat mencegah tindakan *plagiarism* pada Sekolah Tinggi adalah dengan membandingkan tugas akhir mahasiswa dengan tugas akhir yang pernah dibuat sebelumnya. Maka berdasarkan latar belakang dan penjelasan di atas penulis ingin merancang sistem deteksi plagiarisme penulisan tugas akhir menggunakan algoritma Rabin-Karp (studi kasus di Sekolah Tinggi Teknik Payakumbuh).

## 2. Metode Penelitian

Kerangka kerja penelitian ini dilakukan secara sistematis sebagai pedoman peneliti dalam melaksanakan penelitian agar hasil yang dicapai tidak menyimpang dari tujuan yang telah ditetapkan sebelumnya. Kerangka kerja penelitian ini merupakan langkah-langkah yang akan dilakukan dalam rangka menyelesaikan masalah yang akan dibahas.

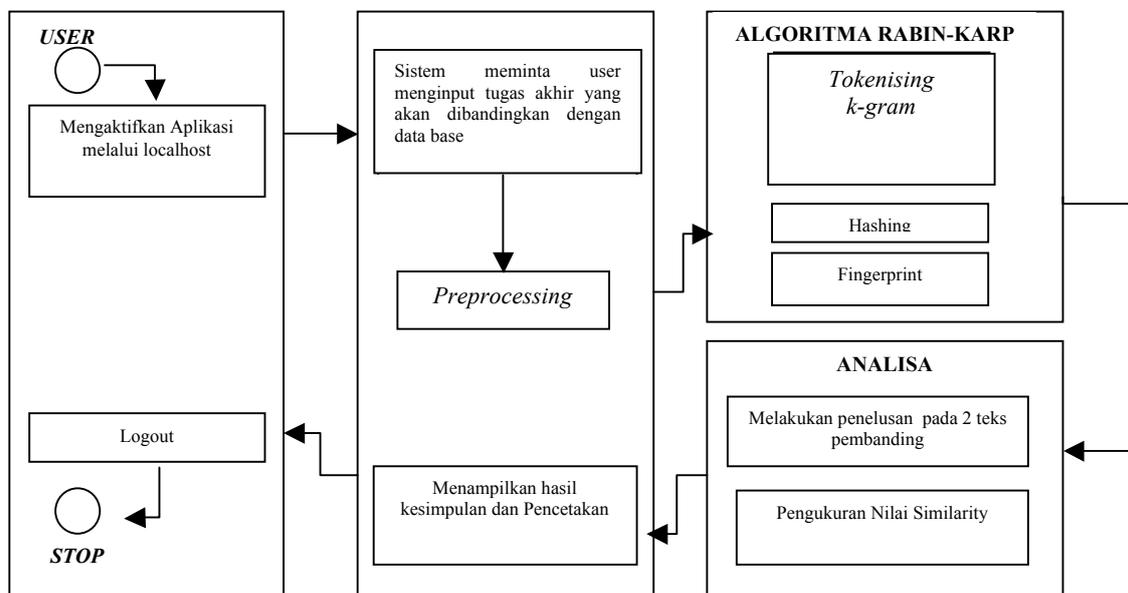
Kerangka kerja dalam melakukan penelitian ini adalah sebagai berikut :



Gambar 1. Kerangka Kerja Penelitian

### 3. Hasil dan Pembahasan

Tahap analisis sistematis kinerja algoritma ini merupakan tahapan yang sangat penting, karena sistem yang telah dipelajari dan diketahui bentuk permasalahan serta rancangan sistem baru yang akan dikembangkan akan mengikuti masing-masing prosedur secara sistematis. Hal ini akan menghasilkan suatu perangkat lunak yang baik yaitu perangkat lunak yang sesuai dengan kebutuhan pengguna. Sistem dapat digambarkan seperti gambar 2.



Gambar 2. Desain Aktifitas Sistem Pencegahan *Plagiarism*

### 3.1. Proses Input

Rancangan aplikasi sistem pencegahan plagiarisme dimulai dengan proses peng-*input*-an dokumen yang akan dideteksi nilai kesamaannya atau dokumen asli dan *input* dokumen pembanding sebanyak 2 dokumen ke dalam sistem. Dokumen yang di *input*-kan adalah dokumen yang berekstensi *.doc* yang di-*input* ke sistem dengan jalan *copy-paste*. Dokumen yang digunakan adalah dokumen *testing* yang mewakili oleh 1 paragraf teks yang mewakili dan secara umum dapat digunakan untuk dokumen lain yang akan diproses pada sistem ini. Dokumen test terdiri atas 2 sampai 3 kalimat dalam 1 paragraf untuk 3 dokumen yang digunakan. Proses lebih menekankan pada penggunaan algoritma Rabin-Karp dalam memproses kesamaan *string* hingga dapat menganalisa plagiarisme pada teks.

Aplikasi pencegahan plagiat dirancang menggunakan bahasa pemrograman PHP. Pada bahasa pemrograman ini dokumen yang dapat dibaca adalah dokumen yang berekstensi *.txt*. Penggunaan cara *copy-paste* adalah agar bahasa pemrograman mengenali teks sebagai *.txt* tanpa proses yang panjang.

**Tabel 1. Contoh Dokumen Input Teks Sumber dan Teks Pembanding**

Dokumen 1
Sekolah Tinggi Teknologi Payakumbuh merupakan sekolah tinggi berbasis teknologi yang memanfaatkan perangkat dan aplikasi komputer untuk membantu aktivitas yang dilakukan agar berjalan cepat, aman dan lancar.

### 3.2 Penerapan Algoritma Rabin-Karp

Penerapan algoritma *hashing* berbasis *K-gram* dalam algoritma Rabin-Karp dilakukan setelah tahapan *preprocessing*. Pada tahapan *preprocessing* teks diambil dan dihilangkan simbol-simbol, *special character*, *space*, tanda baca dan merubah huruf besar ke huruf kecil.

**Tabel 2. Tahap Preprocessing Teks 1**

Teks 1	Hasil Preprocessing
Sekolah Tinggi Teknologi Payakumbuh merupakan sekolah tinggi berbasis teknologi yang memanfaatkan perangkat dan aplikasi komputer untuk membantu aktivitas yang dilakukan agar berjalan cepat, aman dan lancar.	sekolahtinggiteknologipayakumbuhmerupakansekolahtinggiberbasisteknologiyangmemanfaatkanperangkatdanaplikasikomputeruntukmembantuaktivitasyangdilakukanagarberjalancepatamandanlancar

Berdasarkan tabel 2 maka pada tahap *preprocessing* diharapkan teks yang akan dianalisa dapat dihilangkan tanda baca, spasi dan simbol-simbolnya. Hal ini penting agar proses berikutnya dapat berlangsung dengan baik.

#### a. Tahap *Tokenizing k-gram*

Hasil *preprocessing* akan digunakan dan diproses menggunakan algoritma Rabin-Karp dengan menerapkan *K-gram*. Metode ini menghasilkan rangkaian *substring* sejumlah *k-gram*, dimana *k* adalah parameter yang dipilih oleh *user*. *K-gram* mengambil *substring* karakter huruf sejumlah *k* dari sebuah kata yang secara kontinuitas dari teks sumber hingga akhir dokumen. Tahapan ini disebut dengan *Tokenizing K-Gram* atau pemotongan karakter dengan nilai *k* yang diasumsikan 3, 4 dan 5. Nilai *k-gram* bisa kita tetapkan sesuai kebutuhan, namun dalam

penelitian ini digunakan nilai *k-gram* yang lebih kecil seperti 3, 4 dan 5. Tabel 3 salah satu contoh *k-gram* teks 1 dengan nilai *k=3*.

**Tabel 3. Tahap Tokenizing K-Gram Teks 1 dengan Nilai k=3**

Dokumen 1.txt	Hasil Preprocessing
sekolahinggiteknologipayakumbuhm erupakansekolahinggiberbasisteknolo giyangmemanfaatkanperangkatdanapl ikasikomputeruntukmembantuaktivita syangdilakukanagarberjalancepatama ndanlancar	{sek} {eko} {kol} {ola} {lah} {aht} {hti} {tin} {ing} {ngg} {ggi} {git} {ite} {tek} {ekn} {kno} {nol} {olo} {log} {ogi} {gip} {ipa} {pay} {aya} {yak} {aku} {kum} {umb} {mbu} {buh} {uhm} {hme} {mer} {eru} {rup} {upa} {pak} {aka} {kan} {ans} {nse} {sek} {eko} {kol} {ola} {lah} {aht} {hti} {tin} {ing} ...{car}

b. Tahap Hashing

Setelah melakukan proses *preprocessing* dan *tokenizing K-gram* maka langkah selanjutnya adalah melakukan proses *Hashing* terhadap pecahan *string* yang telah dibagi menjadi *K-Gram*. Fungsi ini akan mengganti atau *transpose*-kan data untuk menciptakan *fingerpint* yang biasa disebut *hash value*. Fungsi *hash* yang baik adalah yang menghasilkan sedikit *hash collision*. Solusi yang digunakan untuk mengatasi *hash collision* adalah menggunakan *Rolling Hash*. Dalam prosesnya digunakan basis yang biasanya adalah bilangan prima yang cukup besar, dengan tujuan agar meminimalkan terjadinya tabrakan. Penelitian akan menggunakan basis 3, 5 dan 11. Pada tabel 4 merupakan tahap *hashing* teks 1 dengan nilai *k=3*.

Rumus yang digunakan adalah:

$$c_1 * b^{(k-1)} + c_2 * b^{(k-2)} + \dots + c_{(k-1)} * b + c_k$$

**Tabel 4. Tahap Hashing Teks 1 dengan Nilai k=3**

No	K-Gram k=3			ASCII			Rolling Hash		
	c1	c2	c3	c1	c2	c3	BASIS 3	BASIS 5	BASIS 11
1	s	e	k	115	101	107	1855	4099	16351
2	e	k	o	101	107	111	1773	3817	14797
3	k	o	l	107	111	108	1845	4001	15605
4	o	l	a	111	108	97	1841	4049	16001
5	l	a	h	108	97	104	1762	3878	15410
6	a	h	t	97	104	116	1729	3697	14257
7	h	t	i	104	116	105	1842	3970	15346
8	t	i	n	116	105	110	1894	4170	16566
9	i	n	g	105	110	103	1811	3931	15331
10	n	g	g	110	103	103	1814	3986	15782
11	g	g	i	103	103	105	1755	3815	14939
12	g	i	t	103	105	116	1789	3857	15005
13	i	t	e	105	116	101	1843	3987	15459
14	t	e	k	116	101	107	1864	4124	16472
15	e	k	n	101	107	110	1771	3815	14795
16	s	e	k	115	101	107	1855	4099	16351
17	k	n	o	107	110	111	1845	3997	15589
18	n	o	l	110	111	108	1872	4076	15968
19	o	l	o	111	108	111	1869	4077	16029
20	l	o	g	108	111	103	1844	4016	15716
21	o	g	i	111	103	105	1827	4015	15907
22	g	i	p	103	105	112	1781	3849	14997
23	i	p	a	105	112	97	1811	3939	15363
24	p	a	y	112	97	121	1832	4012	15928
...	...	...	...	...	...	...	...	...	...
178	c	a	r	99	97	114	1701	3673	14341

c. Tahap Fingerprint

Setelah proses *Hashing* selesai, maka nilai *hash* dalam bentuk skema dari dokumen tersebut dikumpulkan, kumpulan nilai *hash* disebut *Fingerpint*. Jika nilai *hash* *valuenya* sama maka tidak perlu dimasukkan lagi ke dalam *fingerpint*. Dengan algoritma Rabin-Karp hasil kedua *fingerpint* dari dua dokumen yang sama akan diambil, sehingga solusi sudah ditemukan.

**Tabel 5. Tahap Fingerprint Dengan Nilai k=3**

No	Rolling Hash Teks 1		
	BASIS 3	BASIS 5	BASIS 11
1	1855	4099	16351
2	1773	3817	14797
3	1845	4001	15605
4	1841	4049	16001
5	1762	3878	15410
6	1729	3697	14257
7	1842	3970	15346
8	1894	4170	16566
9	1811	3931	15331
10	1814	3986	15782
11	1755	3815	14939
12	1789	3857	15005
13	1843	3987	15459
14	1864	4124	16472
15	1771	3815	14795
16	1845	3997	15589
17	1872	4076	15968
18	1869	4077	16029
19	1844	4016	15716
20	1827	4015	15907
21	1781	3849	14997
22	1811	3939	15363
23	1832	4012	15928
24	1793	3829	14593
25	1885	4209	16989
...	...	...	...
107	1701	3673	14341

### 3.3 Analisa Nilai *Similarity*

Setelah menemukan kesamaan nilai *Rolling Hash* dari dokumen, berikutnya adalah menghitung *similarity* dari kumpulan *fingerprinth* tersebut menggunakan *Dice's Similarity Coficient*. *Dices's Similarity Coficient* adalah algoritma genetika analisis komparatif yang digunakan untuk mengetahui dokumen yang paling relevan untuk himpunan kata kunci yang mempunyai nilai sama yang digunakan untuk Nilai *Similaritas* yang dihitung menggunakan rumus:

$$S = \frac{2 * X}{(A + B)} \times 100$$

Dimana :

- X = Jumlah *Hash* yang sama dari dua dokumen
- A dan B = Jumlah Skema *Hash* pada dokumen A dan B

**Tabel 6. Perhitungan *Similarity* Dokumen 1 (A) dan Dokumen 2 (B) Pada k=3**

k=3	Basis		
	3	5	11
Skema has dok 1 (A)	144	144	144
Skema has dok 1 (B)	137	137	137
Skema sama dok 1 dan dok 2 (C)	59	59	59
<b>Similarity (2*X/(A+B))*100</b>	<b>41,99</b>	<b>41,99</b>	<b>41,99</b>

### 3.4 Rancangan Aplikasi Pencegahan Plagiarism

Aplikasi sistem yang dirancang akan menggunakan *form interface* dengan mengintegrasikan setiap tahap dalam proses yang otomatis sehingga *user* mudah dalam mengoperasikannya. Pendeteksian Plagiarisme dilakukan dengan membandingkan 3 dokumen yang dinamakan dokumen A, dokumen B dan dokumen C dengan sistem perbandingan yang berlaku untuk dokumen A dibanding dokumen B dan dokumen A dibandingkan lagi dengan dokumen C. Rancangan *form interface* sistem pencegahan plagiarisme sebagaimana gambar 3 berikut ini:

Aplikasi Pencegahan Plagiat Menggunakan Algoritma Rabin Karp		
<h2>Algoritma Rabin Karp</h2> <p>Algoritma Rabin Karp adalah algoritma yang digunakan untuk melakukan pencocokan string</p> <p style="text-align: center;"><b>Lakukan Proses &gt;&gt;</b></p>		
<b>UPLOAD FILE SOURCE</b>		
Pilih file yang akan dilakukan proses plagiatnya dan masukkan nilai K-Gram		
<b>File Asli</b>		
File Source:	File Pembading 1:	File Pembading 2:
<input type="text"/>	<input type="text"/>	<input type="text"/>
<b>Proses Algoritma Rabin Karp &gt;&gt;</b>		
<b>Preprocessing</b>		
File Source:	Pembading 1	Pembading 2
<input type="text"/>	<input type="text"/>	<input type="text"/>
<b>Tokenizing K-Gram</b>		
File Source:	Pembading 1	Pembading 2
<input type="text"/>	<input type="text"/>	<input type="text"/>

Gambar 3. Rancangan Aplikasi Pencegahan Plagiarism

### 3.5 Rangkuman Hasil Proses

Proses yang terjadi pada sistem berlangsung dalam beberapa tahap. Masing-masing proses dilakukan sebagai berikut :

1. Data Input

Tahapan ini dilakukan dengan melihat data dokumen berekstensi *.doc* diinputkan kemudian ditransformasikan menjadi *.txt*, selanjutnya seharusnya sistem akan langsung melakukan *preprocessing* pada dokumen.txt dan menampilkan ke dalam form *textbox* yang sudah disediakan. Apabila data dokumen input belum dipilih maka seharusnya proses transformasi tidak bisa dilanjutkan dan menginformasikan kepada *user* bahwa data dokumen belum dipilih. Jika tahap ini berhasil maka dilanjutkan ke tahap berikutnya.

2. Proses Deteksi Dokumen

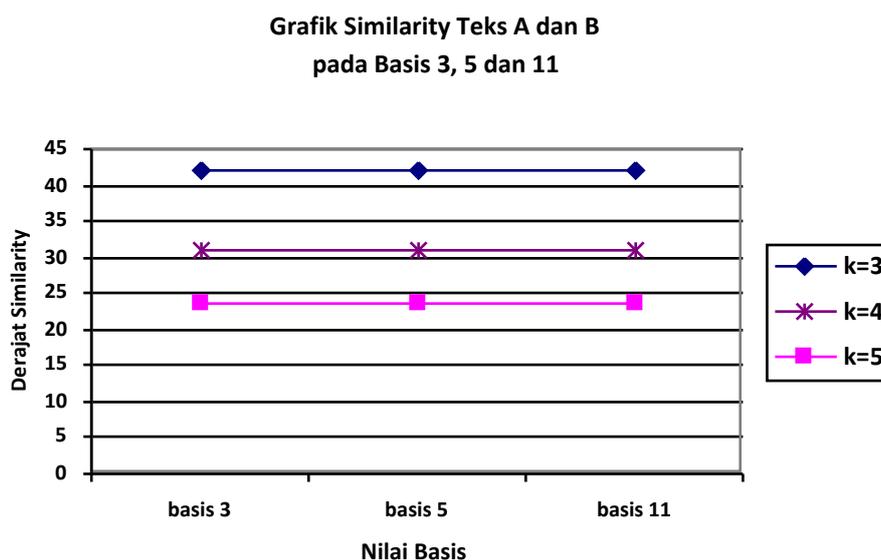
Tahap ini digunakan untuk melihat apakah aplikasi mampu menjalankan proses dalam mendeteksi kesamaan dokumen sesuai dengan algoritma yang digunakan. Salah satu proses yang dilakukan oleh algoritma adalah melakukan pemotongan karakter berdasarkan *k-gram* yang di-inputkan oleh user. Jika *k-gram* yang dipilih user adalah 3 maka seharusnya aplikasi mampu melakukan pemotongan teks dokumen sebanyak 3 karakter. Begitu juga untuk pilihan *k-gram* berikutnya.

### 3. Nilai *Similarity*

Nilai *similarity* ini bertujuan untuk melihat apakah nilai kebenaran dari sistem sesuai dengan hasil hitungan algoritma secara manual. Jika nilai kebenaran tersebut sama maka aplikasi berhasil menjalankan algoritma yang digunakan hingga didapat kesimpulan dari analisa dokumen sumber dan dokumen pembandingan yang dijadikan sebagai dasar dalam membuat keputusan proposal tugas akhir mahasiswa bisa diterima atau ditolak

### 3.6 Pembahasan

Nilai *k-gram* menentukan persentase similaritas. Untuk nilai *k-gram*=3 didapatkan hasil persentase lebih tinggi pada basis yang lebih kecil juga yaitu 3. Jadi, Jika *k-gram* kecil maka nilai persentase *similarity* akan besar. Sebaliknya jika nilai *k-gram* besar maka nilai persentase *similarity* akan menjadi kecil.



**Gambar 4. Grafik *Similarity* Dokumen A dan B**

Dari gambar 4 dapat dilihat kecenderungan terjadinya perubahan nilai *similarity* tertinggi terjadi pada nilai *k*=3.

### 4. Kesimpulan

Berdasarkan analisis dan pembahasan yang dilakukan, maka dapat disimpulkan beberapa hal sebagai berikut.

1. Algoritma Rabin-Karp mampu mendeteksi plagiarisme pada teks sehingga dapat digunakan untuk mendeteksi *similarity* atau kesamaan pada teks sampel yang diambil dari proposal judul tugas akhir.
2. Derajat kesamaan antar teks diperoleh dengan mengubah nilai string menjadi nilai angka pada proses *Hashing*, sehingga proses pembandingan berlangsung lebih cepat karena pembandingan berlangsung antar angka 0-9 bukan dengan abjad a-
3. Berdasarkan perbandingan persentase nilai kesamaan teks maka jika semakin besar nilai *k-gram* maka semakin kecil persentase kesamaan teks yang diperoleh, dan sebaliknya apabila nilai *k-gram* semakin kecil maka semakin besar persentase kesamaan teks yang diperoleh, atau nilai *k-gram* yang semakin kecil menghasilkan akurasi yang lebih baik untuk membandingkan teks.

Saran untuk penelitian kedepannya adalah analisis kesamaan teks yang telah dilakukan mempunyai kekurangan, diantaranya proses *input* teks menggunakan teknik *copy-paste*.

Kekurangan lainnya adalah jumlah teks yang dibandingkan masih sedikit. Sehingga untuk penelitian lebih lanjut disarankan untuk mendeteksi kesamaan dokumen dengan jalan *input* file dokumen sehingga lebih memberikan kemudahan serta penggunaan database sehingga data uji dan data latih yang digunakan lebih bervariasi dan hasil yang diperoleh lebih akurat.

### Daftar Pustaka

- [1] Dian Novian, et al. *Aplikasi Pendeteksian Plagiat Pada Karya Ilmiah Menggunakan Algoritma Rabin-Karp*. Universitas Negeri Gorontalo: Laporan Penelitian. 2012.
  - [2] Amisha Prosad Gope, et al. A Novel Pattern Matching Algorithm in Genome Sequence Analysis. *International Journal of Computer Science and Information Technologies*. 2012; vol 5(no 4): halaman 5450-5457.
  - [3] Akhtar Rasool, et al. String Matching Methodologies:A Comparative Analysis. *International Journal of Computer Science and Information Technologies*. 2012; vol 3(no 2): halaman 3394-3397.
  - [4] Tudesman, Enny Oktalina, Tinaliah, dkk. Sistem Deteksi Plagiarisme Dokumen Bahasa Indonesia Menggunakan Metode *Vector Space Model*. Seminar Perkembangan Penelitian Ilmu Komputer (SPHP-ILKOM). 2014: halaman 392-398.
  - [5] Winangga Milani, Mardji, Achmad Ridok. Deteksi Plagiarisme pada Dokumen Teks Bahasa Indonesia menggunakan Algoritma *Winnowing* dengan *Stemming*. *Jurnal Mahasiswa PTIK UB*. 2014; vol 3(no 12): halaman 1 – 9.
-