

Pencegah Plagiasi Dengan Deteksi Kemiripan Judul Skripsi

Ardi Sanjaya¹, Irfan Fauzi², Moh Fuad Uddin³

^{1, 2, 3}Teknik Informatika, Fakultas Teknik, UN PGRI Kediri

Email : ¹dersky@gmail.com ³mohfuaduddin09@gmail.com

Abstrak — Penelitian ini dibuat bertujuan untuk membuat sistem pendeteksi kemiripan judul skripsi pada program studi Teknik Informatika Universitas Nusantara PGRI Kediri. Sistem ini diimplementasikan dalam bentuk pendaftaran judul proposal skripsi secara *online* dan akan diintegrasikan dengan web prodi. Sistem ini menggunakan metode Rabin Karp dengan nilai gram 2. Teks judul yang diuji dan judul yang dijadikan acuan di-*hash* dan didapatkan substring. Selanjutnya substring digabungkan dan dihilangkan yang memiliki kesamaan. Substring judul yang diuji dibandingkan kesamaan terhadap substring yang telah digabungkan. Apabila nilai kemiripan judul baru kurang dari ambang yang ditentukan maka proses pendaftaran seminar proposal tidak dapat dilanjutkan. Hasil pengujian 10 judul skripsi baru menunjukkan rata-rata tingkat kemiripan kemiripan 0% dan pengujian dengan mengacak 10 judul yang telah ada rata-rata memiliki tingkat kemiripan 100%

Kata Kunci—Plagiasi, Rabin Karp, kemiripan judul.

I. PENDAHULUAN

A. LATAR BELAKANG

Seiring dengan perkembangan teknologi, beberapa pekerjaan manusia dapat menjadi lebih mudah dengan bantuan teknologi komputer seperti dalam hal mengolah data. Dengan bantuan komputer, data yang

diolah akan lebih efektif dan efisien untuk menghasilkan informasi yang diinginkan. Data merupakan fakta yang dapat disimpan dan memiliki arti [1].

Disatu sisi dibalik kemudahan yang didapat seperti kemudahan dalam menyalin dokumen atau data digital, terdapat kecenderungan tindakan negatif yang didominasi kepentingan pribadi salah satunya yaitu plagiarisme. Plagiarisme adalah kegiatan meniru suatu hasil karya orang lain untuk kemudian diakui sebagai karya pribadi. Fenomena plagiarisme juga rawan terjadi pada lingkungan akademis. Misalnya mahasiswa yang kebingungan akan mengerjakan tugas akhir atau skripsi dan belum memiliki tema atau judul skripsi, maka jalan pintas yang bisa ditempuh adalah dengan melakukan plagiarisme. Apalagi jika belum ada sistem yang mampu mencegah kegiatan plagiarisme tersebut.

Program Studi (prodi) Teknik Informatika, Fakultas Teknik Universitas Nusantara PGRI Kediri, merupakan prodi yang memiliki kurang lebih 1700 mahasiswa dan

sekitar 400 diantaranya tingkat 4 atau mahasiswa tingkat akhir (data tahun 2015). Dimana mahasiswa tingkat akhir memiliki salah satu kewajiban yaitu mengerjakan skripsi. Meskipun belum ada laporan tentang plagiarisme di lingkungan prodi, namun tentu saja hal tersebut rawan terjadi mengingat sistem pendaftaran untuk pengajuan judul skripsi masih di kelola secara manual. Selain itu rawan juga terjadi kemiripan/kesamaan judul yang diajukan oleh mahasiswa meski sebenarnya itu faktor kebetulan.

Ada banyak cara dalam mengetahui kemiripan suatu data atau judul skripsi. Diantaranya dengan menggunakan algoritma Rabin Karp [2]. Algoritma Rabin Karp kurang mangkus apabila digunakan dalam pencocokan single pattern dan mengalami keunggulan untuk multiple pattern dibandingkan algoritma lainnya [3].

Berdasarkan uraian tersebut diatas, penulis bermaksud membuat suatu sistem pendaftaran judul skripsi secara online dan deteksi plagiarsime atau kemiripan judul dilakukan oleh sistem menggunakan algoritma Rabin Karp. Mahasiswa diharuskan mengisi data yang diperlukan termasuk judul skripsi yang diajukan. Kemudian sistem secara otomatis akan memeriksa apakah judul yang diajukan sudah pernah ada sebelumnya serta menghitung tingkat

kemiripan jika memang sudah pernah ada.

B. RUMUSAN MASALAH

Berdasarkan latar belakang yang telah diuraikan, maka permasalahan yang dapat dirumuskan pada penelitian ini adalah bagaimana membuat sebuah sistem pendeteksi kemiripan judul skripsi menggunakan algoritma Rabin Karp ?

C. BATASAN PENELITIAN

Pada penelitian ini ditetapkan beberapa hal yang menjadi batasan masalah diantaranya adalah sebagai berikut :

1. Penelitian akan dilakukan pada program studi Teknik Informatika Fakultas Teknik Universitas Nuasnatara PGRI Kediri.
2. Algoritma yang digunakan adalah Rabin Karp.
3. Kemiripan yang diukur hanya judul skripsi.
4. Bahasa pemrograman yang digunakan adalah PHP dan *database* MySql
5. *Output* dari sistem berupa data pendaftaran judul skripsi dengan nilai prosentase kemiripan.

D. TUJUAN

Adapun tujuan dilakukannya penelitian ini adalah untuk membuat sebuah sistem pendeteksi kemiripan judul skripsi menggunakan algoritma Rabin Karp.

II. LANDASAN TEORI

A. Pengertian Sistem

Karakteristik sebuah sistem adalah terdiri dari bagian-bagian yang saling berkaitan dan beroperasi untuk mencapai suatu tujuan. Sebuah sistem bukanlah seperangkat unsur yang tersusun secara tidak teratur, namun sistem terdiri dari unsur yang dapat dikenal untuk saling melengkapi karena memiliki maksud, tujuan dan sasaran tertentu. Sistem adalah suatu kesatuan yang terdiri dari dua atau lebih komponen atau subsistem yang berinteraksi untuk mencapai suatu tujuan [4].

B. Pencocokan Kata

String Matching atau pencocokan *string* adalah suatu metode yang digunakan untuk menemukan suatu keakuratan/hasil dari satu atau beberapa pola teks yang diberikan. *String Matching* merupakan pokok bahasan yang penting dalam ilmu computer karena teks merupakan bentuk utama dari pertukaran informasi antar manusia, misalkan pada literature, karya ilmiah, halaman web dsb [5].

String Matching focus pada pencarian satu, atau lebih umum, semua kehadiran sebuah kata (lebih umum disebut *pattern*) dalam sebuah teks. Semua algoritma yang akan dibahas mengeluarkan semua kehadiran pola teks. Pola dinotasikan

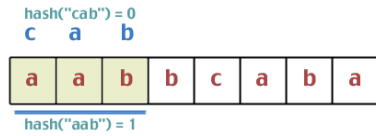
sebagai $x = x[0\dots m-1]$; m adalah panjangnya.

Teks dinotasikan sebagai $y = y[0\dots n-1]$; n adalah panjangnya. Kedua string dibentuk dari set karakter yang disebut alphabet dinotasikan Σ dengan ukuran $\sigma[6]$.

C. Algoritma Rabin Karp

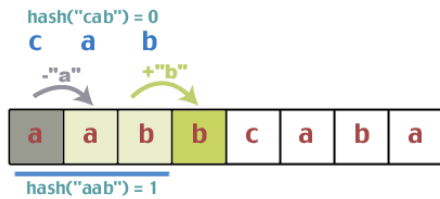
Algoritma Karp-Rabin diciptakan oleh Michael O.Rabin dan Richard M. Karp pada tahun 1978 dengan menggunakan fungsi *hashig* untuk menemukan *pattern* di dalam string teks [7]. Pada dasarnya, algoritma Rabin-Karp akan membandingkan nilai *hash* dari string masukan dan substring pada teks. Apabila sama, 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 [8].

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 [9].



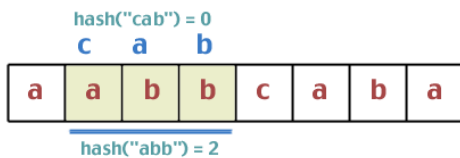
Gambar 1. *Fingerprint* awal

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 *rolling hash* yaitu mengurangi nilai karakter yang keluar dan menambahkan nilai karakter yang masuk sehingga didapatkan kompleksitas waktu yang relative konstan pada setiap kali pergeseran.



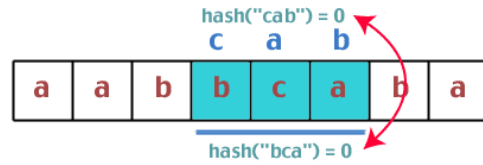
Gambar 2. Menggeser *fingerprint*

Setelah pergeseran, didapatkan nilai *hash* dari *fingerprint* “abb” ($abb = aab - a + b$) menjadi dua ($2 = 1 - 1 + 2$).



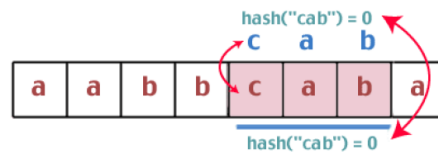
Gambar 3. Perbandingan kedua

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



Gambar 4. Perbandingan keempat (nilai *hash* sama)

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 5. Perbandingan kelima (string ditemukan)

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 $O(mn)$ [8].

D. K-Gram

K-grams adalah rangkaian *terms* dengan panjang K . kebanyakan yang digunakan sebagai *terms* adalah kata.

K-grams merupakan sebuah metode yang diaplikasikan untuk pembandingan kata atau karakter. Metode *K-grams* ini digunakan untuk mengambil potongan-potongan karakter huruf sejumlah *k* dari sebuah kata yang secara kontinuitas dibaca dari teks sumber hingga akhir dari dokumen.

E. Pengukuran Nilai Similarity

Inti dari pendekatan *K-grams* dibagi menjadi dua tahap. Pada tahap pertama, membagi kata menjadi *K-grams*. Sedangkan pada tahap kedua, mengelompokkan hasil *terms* dari *K-grams* yang sama. Kemudian untuk menghitung *similarity* dari kumpulan kata tersebut, maka digunakan *Dice's Similarity Coefficient* untuk pasangan kata yang digunakan [5].

Untuk menghitung nilai similaritas, digunakan hitungan sebagai berikut :

$$S = \frac{2C}{A+B}$$

Dimana *S* adalah nilai *similarity*, *A* dan *B* adalah jumlah dari kumpulan *K-grams* dalam teks 1 dan teks 2. *C* adalah jumlah dari *K-grams* yang sama dari kedua teks yang dibandingkan.

Contoh perhitungan nilai *similarity* 3 kata dengan *K* = 2 (*bi-grams*)

<i>Kata yang dibandingkan (*)</i>	<i>K-grams yang sama</i>	<i>Similarity</i>
<i>Photography (9) dan Photographic (10)</i>	<i>Ph ho ot to gr ra ap ph = 8</i>	$2*8/(9+10)=0.84$
<i>Photography (9) dan Phonetic (7)</i>	<i>Ph ho = 2</i>	$2*2/(9+7) = 0.25$
<i>Photographic (10) dan Phonetic (7)</i>	<i>Ph ho ic = 3</i>	$2*3/(10+7)=0.35$

Tabel 1. Perhitungan Nilai *Similarity* 3 kata dengan *K* = 2

F. Pengukuran

Untuk menentukan jenis plagiarisme antara dokumen yang diuji ada 5 jenis penilaian persentase *similarity* [9] :

- 0% : Hasil uji 0% berarti kedua dokumen tersebut benar-benar berbeda baik dari segi isi dan kalimat secara keseluruhan.
- < 15% : Hasil uji 15% berarti kedua dokumen tersebut hanya mempunyai sedikit kesamaan.
- 15-50% : Hasil uji 15-50% berarti menandakan dokumen tersebut termasuk plagiat tingkat sedang.
- > 50% : Hasil uji lebih besar 50% berarti dapat dikatakan bahwa

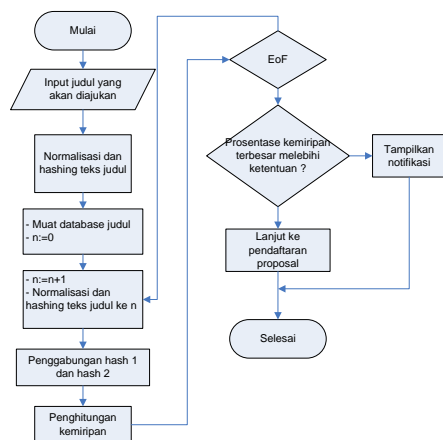
dokumen tersebut mendekati plagiarisme.

- 100% : Hasil uji 100% menandakan bahwa dokumen tersebut adalah plagiat karena dari awal sampai akhir mempunyai isi yang sama persis.

III. HASIL DAN PEMBAHASAN

A. Alur sistem

Proses awal dari proses pengecekan plagiasi adalah melalui form yang harus diisi oleh mahasiswa sebelum melakukan pendaftaran sidang proposal skripsi. Salah satu yang harus diisi adalah judul proposal skripsi. Apabila hasil prosentase kemiripan melebihi yang telah ditentukan, maka sistem akan memberikan notifikasi bahwa judul tersebut memiliki nilai plagiasi yang melebihi batasan. Apabila nilai prosentase lebih rendah maka mahasiswa bisa melanjutkan untuk mendaftar sidang proposal. Alur pendaftaran sidang proposal disajikan pada gambar berikut :



Gambar 6. Alur sistem

B. Proses Hashing

Proses hashing menggunakan kode program seperti berikut :

```

for
($i=0;$i<=strlen($teks1)-$gram;$i++)
{
    $hash1=$hash1.'
['.substr($teks1,$i,$gram).']';
    array
    $cek=0;
    for
    ($j=0;$j<=$arr_i_h1;$j++)
    {
        if
        (substr($teks1,$i,$gram)=$arr_h1[$j])
        { $cek=1; }
        if ($cek==0)
        {
            $arr_i_h1=$arr_i_h1+1;

            $arr_h1[$arr_i_h1]=substr($teks1,$i,$gram);
        }
    }
}
  
```

C. Menghitung jumlah substring

Setelah dilakukan hashing, didapat substring dari masing-masing judul yang diuji dan judul yang dijadikan acuan. Seluruh substring digabungkan dan dihilangkan substring yang ganda. Didapatkan juga nilai jumlah total substring hasil penggabungan. Penggabungan menggunakan kode program berikut :

```
for
($i=1;$i<=$arr_i_h2;$i++)
{ $arr_h1[$arr_i_h1 +
$i]=$arr_h2[$i]; }
```

Kemudian setiap substring dari judul yang diuji dicocokkan terhadap hasil penggabungan substring. Maka didapat nilai jumlah substring yang sama. Penentuan kesamaan dari jumlah substring judul yang diuji terhadap penggabungan substring menggunakan kode program sebagai berikut :

```
$arr_i=0;
for ($i=1;$i<=($arr_i_h1
+ $arr_i_h2);$i++)
{
    $cek=0;
    for ($j=$i+1;$j<=
($arr_i_h1 +
$arr_i_h2);$j++)//cek
kedepan apa ada yg sama
    {
        if
($arr_h1[$i]==$arr_h1[$j]
){ $cek=1;};
    }
    if ($cek==0)
    {
        $arr_i=$arr_i + 1;

    $arr[$arr_i]=$arr_h1[$i]
;
    }
}
```

D. Penentuan prosentase kemiripan

Untuk menentukan prosentase nilai kemiripan, apabila jumlah

substring hash judul yang diinputkan lebih kecil dari jumlah substring judul acuan yang ada pada database, maka jumlah substring data uji yang mirip dengan substing hasil penggabungan dibagi jumlah penggabungan substring dikali 100. Apabila jumlah substring hash judul yang diinputkan lebih besar dari jumlah substring judul acuan yang ada pada database, maka jumlah substring data acuan yang mirip dengan substing hasil penggabungan dibagi jumlah penggabungan substring dikali 100.

Jika jumlah substring hash judul yang diinputkan lebih kecil dari jumlah substring judul acuan yang ada pada database, maka jumlah substring data uji yang mirip dengan substing hasil penggabungan dibagi jumlah penggabungan substring dikali 100. Kode programnya disajikan sebagai berikut :

```
if ($arr_i_h1<$arr_i_h2)
{
    $k=($arr_i_h1/$arr_i)*100;
}
if ($arr_i_h1>$arr_i_h2)
{
    $k=($arr_i_h2/$arr_i)*100;
}
if ($arr_i_h1==$arr_i_h2)
{
    $k=($arr_i_h1/$arr_i)*100;
}
```

E. Testing

Proses pengujian atau *testing* dilakukan dengan cara menguji 10

judul skripsi baru yang diinputkan dan 10 judul skripsi lama yang diambil secara acak terhadap *database* judul sejumlah 300. Didapat hasil sebagai berikut :

Tabel Daftar 2. judul skripsi baru

NO	JUDUL	KEMIRIPAN
1	SISTEM APLIKASI PENGENALAN POLA BATIK DI KEDIRI	0%
2	SISTEM PENYELEKSI TELUR MENGGUNAKAN CITRA EDGE BERBASIS ANDROID	0%
3	PREDIKSI WAHANA PERMAINAN FAVORIT GAME FANTASIA DI KEDIRI MAL	0%
4	IMPLEMENTASI FUZZY TSUKAMOTO DALAM PENENTUAN HARGA CETAK FOTO DIGITAL	0%
5	APLIKASI PEMBELAJARAN BAHASA ARAB BERBASIS ANDROID	0%
6	SISTEM INFORMASI PENCARIAN RUTE TERPENDEK SARANA OLAHRAGA DI	0%

	KOTA KEDIRI UNTUK ANDROID	
7	PENENTUAN HARGA PENJUALAN TAS SECARA GROSIR MENGGUNAKAN METODE FUZZY TSUKAMOTO	0%
8	APLIKASI PEMBACA KEPRIBADIAN DENGAN METODE TES MBTI	0%
9	PENERAPAN METODE FUZZY TSUKAMOTO UNTUK ESTIMASI STOK BARANG	0%
10	ANALISIS PREDIKSI PENJUALAN KRECEK KRUPUK CV.SINAR PAMENANG KEDIRI	0%

Tabel 3. Daftar 10 judul skripsi lama acak

NO	JUDUL	KEMIRIPAN
1	APLIKASI PREDIKSI HASIL PANEN PADI DENGAN METODE LEAST SQUARE	100%
2	IMPLEMENTASI HIERARCHICAL CLUSTERING DAN BRANCH AND BOUND PADA SIMULASI PENDISTRIBUSIAN PAKET POS	100%

3	PREDIKSI PRODUKSI PENJUALAN SENAPAN ANGIN CV. BIMA ASTA MENGUNAKAN METODE TREND MOMENT	100%	ONLINE BERBASIS WEB DAN SMS GATEWAY PADA SMK PGRI 2 KEDIRI
4	PERANCANGAN APLIKASI PENDUKUNG KEPUTUSAN PEMILIHAN GURU TELADAN MENGUNAKAN METODE ANALYTICAL HIERARCHY PROCESS (AHP) DI SMK NEGERI 1 KEDIRI	100%	9 OPTIMALISASI JUMLAH PRODUKSI JAMU JAYA ASLI DENGAN MENGUNAKAN METODE TSUKAMOTO
5	PENERAPAN METODE TREND MOMENT UNTUK MEMPREDIKSI KEBUTUHAN OBAT PERIODE MENDATANG DI PUSKESMAS NGADILUWIH	100%	10 SISTEM REKOMENDASI PENERIMAAN BEASISWA
6	CLUSTERING NILAI SISWA MENGUNAKAN METODE K- MEANS	100%	
7	PERANCANGAN SISTEM PENDUKUNG KEPUTUSAN BEASISWA MENGUNAKAN METODE PROFILE MATCHING	100%	
8	SISTEM INFORMASI PERPUSTAKAAN	100%	

Pencegahan Plagiarisme Skripsi
Prodi Teknik Informatika | Fakultas Teknik

Judul Anda :
SISTEM REKOMENDASI PENERIMAAN BEASISWA

Hasil :
Maaf, silakan mencari judul yang lain. Nilai kemiripan 100% terhadap judul :
SISTEM REKOMENDASI PENERIMAAN BEASISWA
>> Oleh DARUDIN (2016)

[Kembali](#)

Gambar 7. Screenshot aplikasi

IV. SIMPULAN DAN SARAN

A. Simpulan

Berdasarkan dari hasil percobaan dapat disimpulkan bahwa :

1. Telah berhasil dibuat aplikasi atau sistem pendeteksi kemiripan judul skripsi pada prodi Teknik Informatika .
2. Hasil pengujian 10 judul skripsi baru menunjukkan hasil kemiripan rata-rata 0%. Sedangkan hasil pengujian 10 judul skripsi lama yang diambil acak menunjukkan hasil kemiripan rata-rata 100%.

B. Saran

Kedepannya, penelitian akan dilanjutkan untuk mengembangkan sistem pencegah plagiasi tidak hanya sebatas kemiripan judul skripsi saja namun pada kemiripan abstrak, latar belakang dan tujuan.

DAFTAR PUSTAKA

- [1]. Navathe., Elmasri. 2013. *Fundamentals Of Database Systems*, http://www.uoitc.edu.iq/images/documents/informatics-institute/Competitive_exam/Databas_e_Systems.pdf, diakses 23 Mei 2016
- [2]. Baedlowi, Najib., Aditia, Deka. 2006. *String Matching dengan menggunakan Algoritma Rabin Karp*, <http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2005-2006/Makalah2006/MakalahStmik2006-15.pdf>, diakses 23 Mei 2016
- [3]. Sumalsih., Andi. 2013. *Implementasi Algoritma Rabin Karp Untuk Pendeteksian Plagiat Dokumen Teks Menggunakan Konsep Similarity*. http://repository.amikom.ac.id/files/Publikasi_08.11.2281.pdf, diakses 23 Mei 2016
- [4]. Jogyanto, 2001. *Analisa dan Desain Sistem Informasi*, Penerbit Andi, Yogyakarta
- [5]. Nugroho, Eko. 2011. *Perancangan Sistem Deteksi Plagiarisme Dokumen Teks Dengan Menggunakan Algoritma Rabin-Karp*, Program Studi Ilmu Komputer, Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Brawijaya.
- [6]. Atmopawiro, Alasian. 2006. *Pengkajian Dan Analisis Tiga Algoritma Efisien Rabin-Karp, Knut-Morris-Pratt, Dan Boyer-Moore Dalam Pencarian Pola Dalam Suatu Teks*. Program Studi Teknik Informatika, Institut Teknologi Bandung.
- [7]. Fernando, Hary, 2009. *Perbandingan dan Pengujian Beberapa Algoritma Pencocokan String*, Program Studi Teknik Informatika, Institut Teknologi Bandung (ITB). Bandung.
- [8] Firdaus, Hari Bagus, 2008. *Deteksi Plagiat Dokumen Menggunakan Algoritma Rabin-Karp*, Program Studi Teknik Informatika Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung (ITB), Bandung
- [9] A. Benny dan Sinta, 2008. *Aplikasi Anti Plagiarisme Dengan Algoritma Rabin Karp Pada Penulisan Ilmiah Universitas Gunadarma*, http://www.gunadarma.ac.id/library/articles/graduate/industrial-technology/2008/Artikel_50404716.pdf, diunduh 5 April 2016