

# Implementasi Algoritma *Rabin-Karp* untuk Membantu Pendeteksian Plagiat pada Karya Ilmiah

Doddi Aria Putra<sup>1</sup>, Herry Sujaini<sup>2</sup>, Helen Sasty Pratiwi<sup>3</sup>.

Program Studi Teknik Informatika Fakultas Teknik Universitas Tanjungpura<sup>1,2,3</sup>

*e-mail*: <sup>1</sup>doddiariaputra@gmail.com, <sup>2</sup>herry\_sujaini@yahoo.com, <sup>3</sup>helensastypratiwi@gmail.com

**Abstrak**— Karya ilmiah adalah karangan ilmu pengetahuan yang menyajikan fakta dan ditulis menurut metodologi penulisan yang baik dan benar. Namun tidak dipungkiri adanya orang-orang yang tidak bertanggung jawab menyalahgunakan karya ilmiah dengan mengutip sebagian atau seluruh karya dan karya ilmiah pihak lain yang diakui sebagai karya ilmiahnya, tanpa menyatakan sumber secara tepat dan memadai disebut dengan tindakan plagiat. Penelitian ini bertujuan untuk menghasilkan suatu aplikasi dengan penerapan Algoritma *Rabin-Karp* sebagai alat bantu mendeteksi kemiripan sebuah teks atau dokumen karya ilmiah dengan karya ilmiah lain yang berpotensi mengarah kepada tindakan plagiat. Aplikasi dibangun berbasis *web* dengan menggunakan bahasa pemrograman PHP dan database MySQL. Proses pendeteksian kemiripan karya tulis akan melalui proses *case folding*, *tokenizing*, *filtering*, *stemming*, *k-gram* dan *hashing* kemudian dengan menggunakan Algoritma *Rabin-Karp* akan dicari kemiripannya dan menggunakan *Dice's Similarity Coefficient* untuk menghitung persentase kemiripannya. Pengujian dilakukan dengan metode *white box* dan pengisian kuesioner oleh 100 responden. Berdasarkan hasil pengujian dan pengukuran skala hasil dari kuesioner dengan *Likert's Summated Rating* menunjukkan bahwa aplikasi sangat positif dan dinilai berhasil serta sebanyak 15,4% responden sangat setuju dengan hasil keluaran sistem, 41,1% responden setuju dengan hasil keluaran sistem dan 22% responden cukup setuju dengan hasil keluaran sistem. Sisanya sebanyak 13,8% kurang setuju dan 7,7% tidak setuju dengan hasil keluaran sistem.

**Kata Kunci**— Algoritma *Rabin-Karp*, *Hashing*, Karya ilmiah, *K-Gram*, Plagiat.

## I. PENDAHULUAN

Karangan ilmiah atau karya ilmiah adalah karangan ilmu pengetahuan yang menyajikan fakta dan ditulis menurut metodologi penulisan yang baik dan benar. Karya ilmiah dapat juga berarti tulisan yang didasari oleh hasil pengamatan, peninjauan, penelitian dalam bidang tertentu, disusun menurut metode tertentu dengan sistematika penulisan yang bersantun bahasa dan isinya dapat di pertanggung jawabkan kebenarannya/keilmiahannya [1].

Seiring perkembangan teknologi yang saat ini semakin maju, karya tulis ilmiah selain disimpan dalam bentuk buku (*hardcopy*) juga disimpan pula dalam bentuk file (*softcopy*). Tidak jarang pula karya tulis tersebut di bagikan ke orang lain atau pun ke internet yang dimaksudkan dapat dibaca dan dimanfaatkan sebagai referensi pembelajaran oleh orang lain. Namun tidak dipungkiri adanya orang-orang yang tidak bertanggung jawab menyalahgunakan karya ilmiah yang dibagikan tersebut dengan meng-*copy* isi karya ilmiah tanpa

izin dan sepengetahuan penulis untuk kepentingan pribadinya. Ini yang disebut dengan plagiarisme atau plagiat.

Plagiarisme adalah tindakan mengambil ide atau tulisan orang lain tanpa rujukan dan mengklaim bahwa ide dan tulisan tersebut adalah miliknya [2]. Biasanya untuk melihat adanya atau tidak penjiplakan yang terjadi pada suatu karya ilmiah tersebut dilakukan secara manual dengan membaca semua isinya lalu dibandingkan dengan karya ilmiah lain. Dengan begitu, pemeriksaan tersebut membutuhkan waktu yang cukup banyak.

Algoritma *Rabin-Karp* adalah suatu algoritma pencarian string yang ditemukan oleh Michael Rabin dan Richard Karp. Algoritma ini menggunakan *hashing* untuk menemukan sebuah *substring* dalam sebuah teks [3]. Disebut algoritma pencarian string dan bukan pencocokan string seperti *Knuth-Morris-Pratt* atau *Boyer-Moore* karena memang Algoritma *Rabin-Karp* tidak bertujuan menemukan string yang cocok dengan string masukan, melainkan menemukan pola (*pattern*) yang sekiranya sesuai dengan teks masukan.

Diharapkan aplikasi yang akan dirancang pada penelitian ini dapat membantu pengguna untuk melakukan pendeteksian kemiripan sebuah karya ilmiah dengan karya ilmiah lain yang berpotensi mengarah kepada tindakan plagiat.

## II. URAIAN PENELITIAN

### A. Plagiarisme

Tidaklah mudah untuk mengatakan apakah suatu karya "ya" atau "tidak" mengandung unsur plagiat. Sehingga sangatlah penting untuk memahami definisi plagiat atau plagiarisme dari berbagai sumber.

Plagiat adalah penjiplakan atau pengambilan karangan, pendapat, dan sebagainya dari orang lain dan menjadikannya seolah-olah karangan dan pendapat sendiri [4].

### B. Ekstraksi Dokumen

Teks yang akan dilakukan proses teks mining, pada umumnya memiliki beberapa karakteristik diantaranya adalah memiliki dimensi yang tinggi, terdapat noise pada data, dan terdapat struktur teks yang tidak baik. Diperlukan tahap *preprocessing* yang dilakukan secara umum dalam teks mining pada dokumen, yaitu *case folding*, *tokenizing*, *filtering* dan *stemming*.

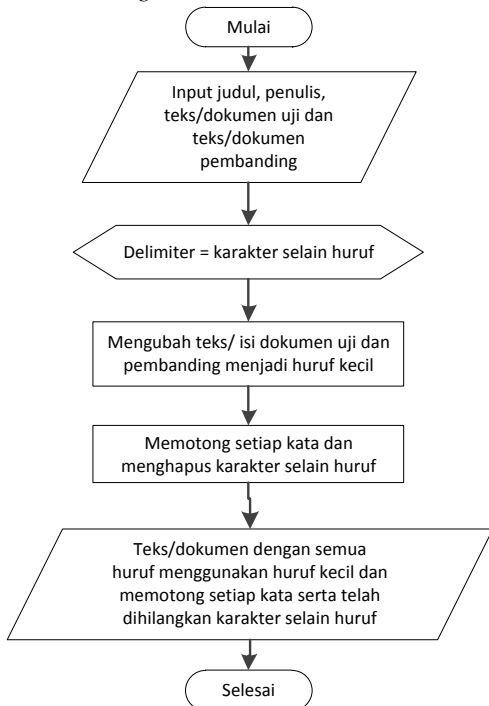
#### *Case Folding*

*Case folding* adalah mengubah semua huruf dalam dokumen menjadi huruf kecil. Ini dilakukan untuk mempermudah

pencarian. Tidak semua dokumen teks konsisten dalam penggunaan huruf kapital. Oleh karena itu peran *case folding* dibutuhkan dalam mengkonversi keseluruhan teks dalam dokumen menjadi suatu bentuk standar (huruf kecil) [5].

**Tokenizing**

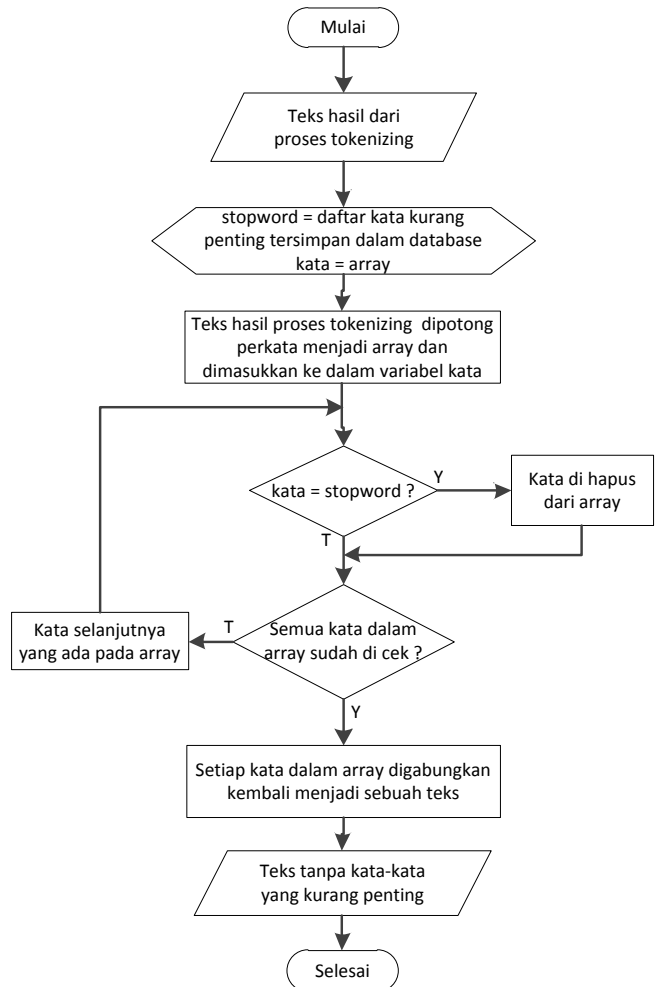
Tahap *tokenizing* adalah tahap pemotongan string masukan berdasarkan tiap kata yang menyusunnya [5]. Karakter selain huruf dihilangkan dan dianggap *delimiter*. *Delimiter* adalah Urutan satu karakter atau lebih yang dipakai untuk membatasi atau memisahkan data yang disajikan dalam kalimat. Salah satu contoh dari *delimiter* adalah tanda koma, titik koma atau titik dua. Berikut ini adalah diagram alir dari proses *case folding* dan *tokenizing*.



Gambar 1 Diagram alir dari proses *case folding* dan *tokenizing*

**Filtering**

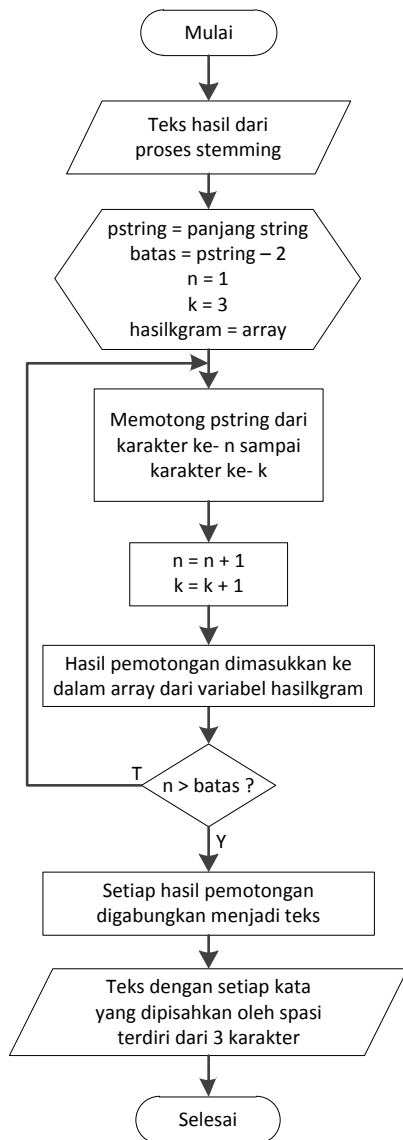
*Filtering* adalah tahap mengambil kata-kata penting dari hasil *tokenizing*. *Filtering* dapat menggunakan algoritma *stoplist* (membuang kata yang kurang penting) atau *wordlist* (menyimpan kata penting). Dalam penelitian ini akan menggunakan *stoplist*. *Stoplist* atau yang biasa juga disebut *stopword* adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan *bag-of-words* (struktur kalimat tidak diperhatikan). Contoh *stopword* adalah “yang”, “dan”, “di”, “dari” dan lain-lain [5]. Berikut ini adalah diagram alir dari proses *filtering*.



Gambar 2 Diagram alir dari proses *filtering*

**Stemming**

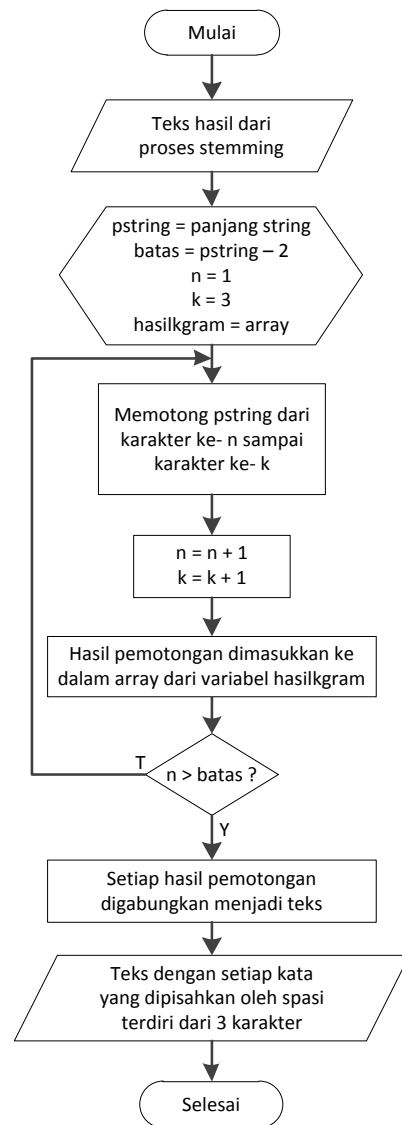
Tahap *stemming* adalah tahap mencari kata dasar kata dari tiap kata hasil *filtering*. Pada tahap ini dilakukan proses pengembalian berbagai bentukan kata ke dalam suatu representasi yang sama. Tahap ini kebanyakan dipakai untuk teks berbahasa Inggris dan lebih sulit diterapkan pada teks berbahasa Indonesia. Hal ini dikarenakan Bahasa Indonesia tidak memiliki rumus bentuk baku yang permanen [5]. Berikut ini adalah diagram alir dari proses *stemming*.



Gambar 3 Diagram alir dari proses *stemming*

**C. 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 [6]. Dalam penelitian ini menggunakan 3-gram. Berikut ini adalah diagram alir dari proses k-gram.



Gambar 4 Diagram alir dari proses k-gram

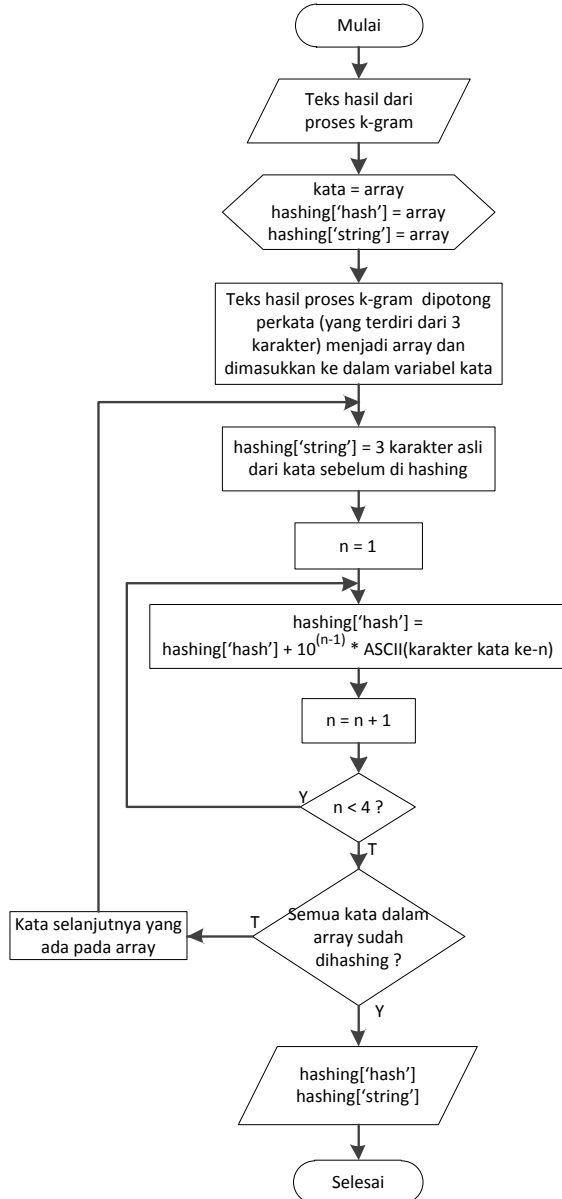
**D. Hashing**

*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*.

Misalnya penggunaan *hashing* dalam pencarian data pada *database*. Apabila tidak di-*hash*, pencarian akan dilakukan karakter per karakter pada nama-nama yang panjangnya bervariasi dan ada 26 kemungkinan pada setiap karakter. Namun pencarian akan menjadi lebih efisien setelah di-*hash* karena hanya akan membandingkan empat digit angka dengan cuma 10 kemungkinan setiap angka.

Algoritma *Rabin-Karp* didasarkan jika dua buah *string* sama maka harga *hash value*-nya pasti sama. Akan tetapi ada dua masalah yang timbul dari hal ini, masalah pertama yaitu ada begitu banyak *string* yang berbeda, permasalahan ini dapat dipecahkan dengan meng-*assign* beberapa *string* dengan *hash value* yang sama. Masalah yang kedua belum tentu *string* yang

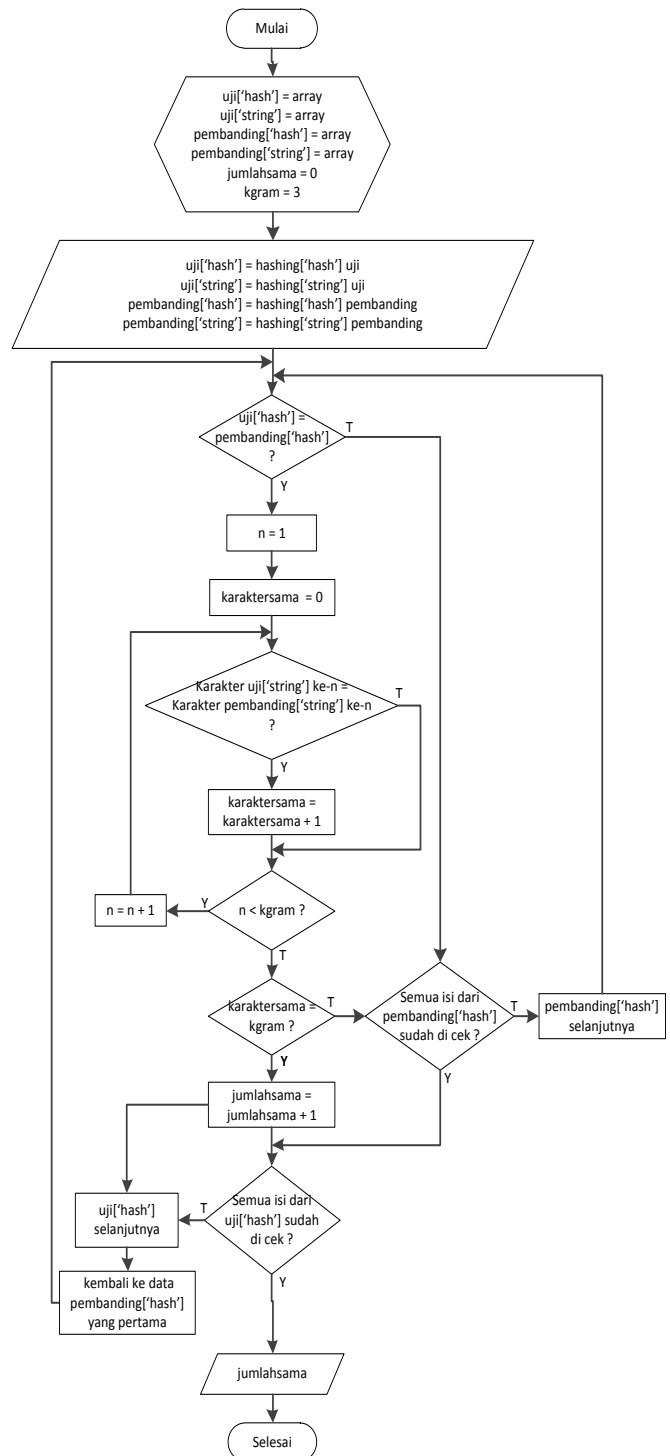
mempunyai *hash value* yang sama cocok untuk mengatasinya maka untuk setiap *string* yang di-assign dilakukan pencocokan *string* secara *Brute-Force*. Kunci agar Algoritma *Rabin-Karp* efisien, terdapat pada pemilihan *hash value*-nya. Salah satu cara yang terkenal dan efektif adalah memperlakukan setiap *substring* sebagai suatu bilangan dengan basis tertentu [7]. Berikut ini adalah diagram alir dari proses *hashing*.



Gambar 5 Diagram alir dari proses *hashing*

**E. Algoritma Rabin-Karp**

Algoritma *Rabin-Karp* adalah algoritma pencocokan *string* yang menggunakan fungsi *hash* sebagai pembanding antara *string* yang dicari (m) dengan *substring* pada teks (n). Apabila *hash value* keduanya sama maka akan dilakukan perbandingan sekali lagi terhadap karakter-karakternya. Apabila hasil keduanya tidak sama, maka *substring* akan bergeser ke kanan. Pergeseran dilakukan sebanyak (n-m) kali. Berikut ini adalah diagram alir dari Algoritma *Rabin-Karp*.



Gambar 6 Diagram alir dari Algoritma *Rabin-Karp*

**F. Perhitungan Nilai Similarity**

Untuk menghitung nilai *similarity* dari dokumen *fingerprint* yang didapat maka digunakan *Dice's Similarity Coefficients* 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.

$$S = \frac{2C}{A+B} \tag{1}$$

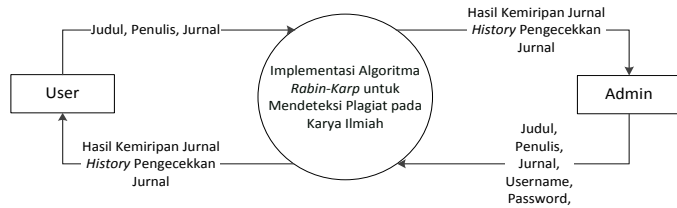
Di mana:

- S : Nilai *similarity*
- A : Jumlah k-gram dari teks 1
- B : Jumlah k-gram dari teks 2
- C : Jumlah k-gram yang sama dari teks 1 dan teks 2

### III. PERANCANGAN SISTEM

#### A. Diagram Konteks Sistem

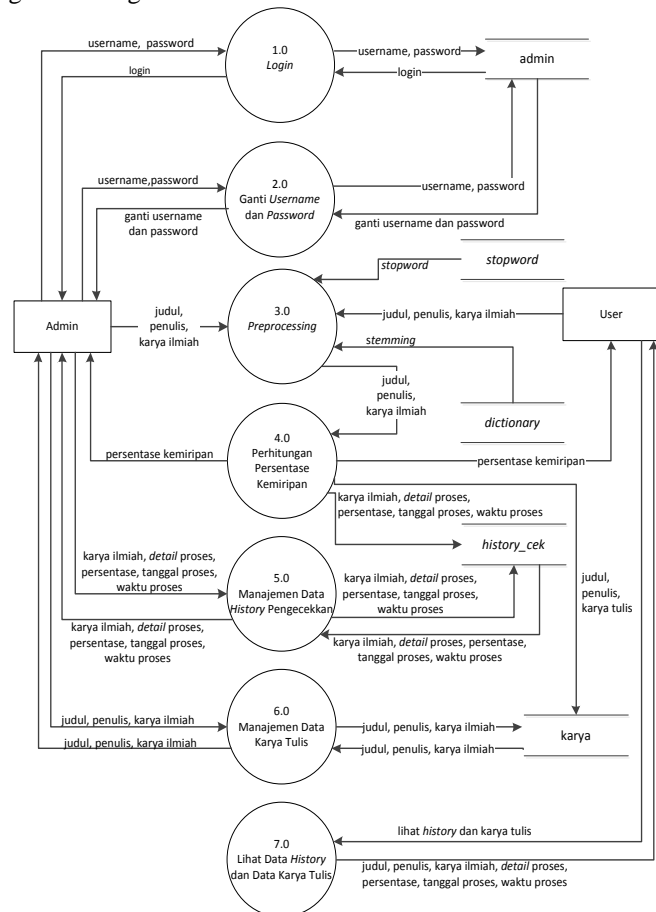
Diagram konteks adalah diagram yang memberikan gambaran umum terhadap kegiatan yang berlangsung pada sistem. Berikut ini adalah gambar yang akan menunjukkan diagram konteks dari sistem ini.



Gambar 7 Diagram konteks sistem

#### B. Diagram Overview Sistem

Diagram *overview* adalah diagram yang menjelaskan urutan-urutan proses dari diagram konteks. Berikut ini adalah gambar diagram *overview* dari sistem ini.

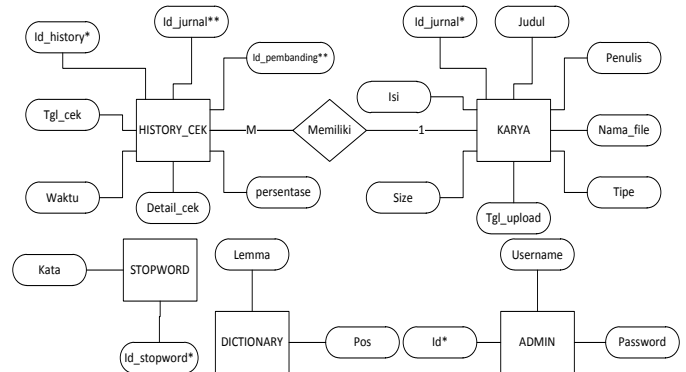


Gambar 8 Diagram *overview* sistem

#### C. Entity Relationship Diagram

*Entity Relationship Diagram* (ERD) merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. ERD untuk memodelkan struktur data dan hubungan antar data, untuk menggambarkannya digunakan beberapa notasi dan simbol.

Berikut ini adalah *Entity Relationship Diagram* dari sistem ini.



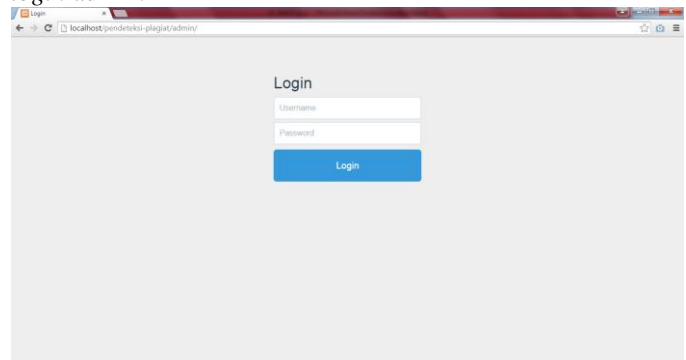
Gambar 9 *Entity relationship diagram* sistem

### IV. HASIL DAN ANALISIS

#### A. Hasil Perancangan

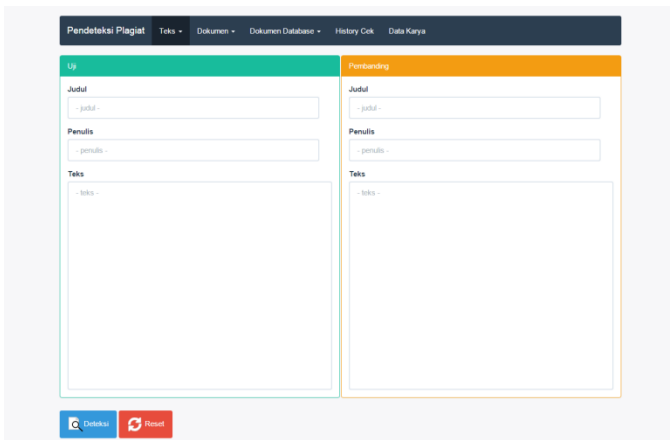
Sistem aplikasi yang dirancang dengan tujuan untuk membantu pengguna dalam menentukan kemiripan antar jurnal yang mana kemiripan tersebut termasuk kedalam tindakan plagiarisme atau peniruan karya ilmiah. Aplikasi ini merupakan aplikasi berbasis web.

Aplikasi yang dibangun hanya ditujukan untuk penggunaan pada sisi end-user dan dapat digunakan oleh siapapun. Oleh karena itu, antarmuka dirancang sesederhana mungkin agar dapat digunakan dengan mudah. Berikut ini adalah halaman *login* admin.

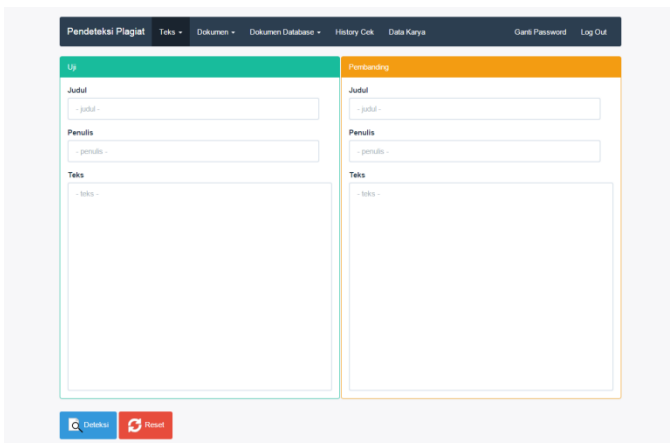


Gambar 10 Antarmuka halaman *login* admin

Setelah melakukan *login* sebagai admin, berikut ini perubahan tampilan antarmuka dari sistem yang digunakan untuk *user* atau ketika tidak *login* sebagai admin pada Gambar 11 dan setelah melakukan *login* pada Gambar 12 yang memiliki penambahan menu ganti *username* dan *password* dan *log out*.

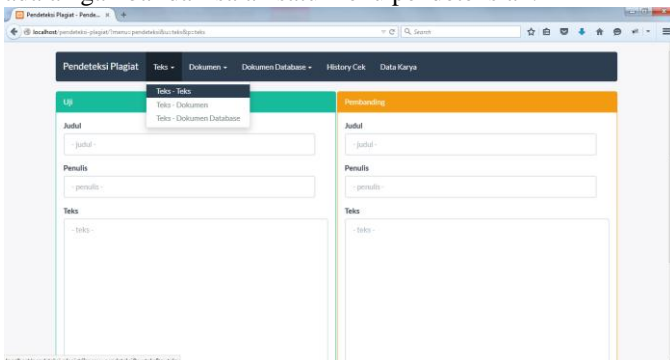


Gambar 11 Antarmuka sebelum login



Gambar 12 Antarmuka setelah login sebagai admin

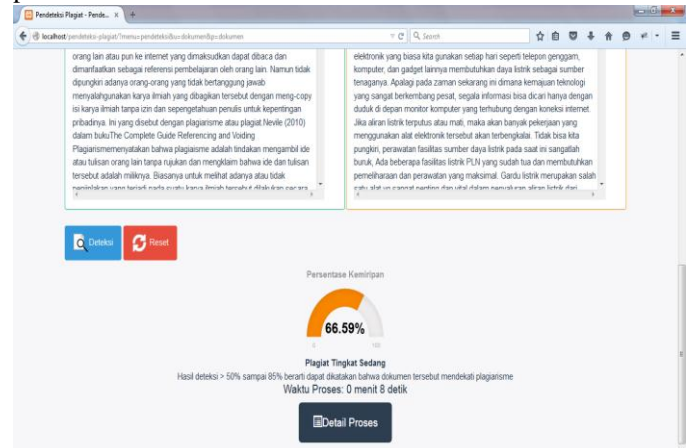
Halaman pendeteksian merupakan halaman inti dari aplikasi ini dapat digunakan oleh admin dan *user* untuk melakukan pendeteksian kemiripan jurnal. Dalam melakukan pendeteksian pengguna harus menuliskan judul dan nama penulis dari jurnal yang dimasukkan ke dalam aplikasi untuk dilakukan pengecekan dan pengguna juga dapat memilih 3 *menu* untuk melakukan pendeteksian yaitu Teks, Dokumen dan Dokumen Database yang mana menu tersebut menentukan *input*-an jurnal kedalam aplikasi untuk di deteksi. Berikut ini adalah gambar dari salah satu menu pendeteksian.



Gambar 13 Antarmuka halaman pendeteksian

Setelah mengklik tombol deteksi maka sistem akan mendeteksi antara dokumen uji dan dokumen pembandingan. Setelah proses pendeteksian selesai maka akan tampil hasil

persentase kemiripan dari kedua dokumen tersebut seperti pada Gambar 14.



Gambar 14 Antarmuka halaman hasil pendeteksian

Pada hasil proses akan ditampilkan lama waktu proses. Selain info lama waktu proses, pada halaman hasil proses juga akan menampilkan persentase kemiripan disertai indikasi warna. Sistem mengkategorikan 5 jenis hasil pendeteksian sebagai berikut [8].

1. Tidak Mirip, yaitu hasil deteksi 0% yang berarti kedua dokumen tersebut berbeda baik dari segi isi dan kalimat secara keseluruhan. Indikasi warna persentase adalah warna biru.
2. Sedikit Mirip, yaitu hasil deteksi lebih dari 0% sampai 25% yang berarti dokumen tersebut hanya mempunyai sedikit kesamaan. Indikasi warna persentase adalah warna hijau.
3. Cukup Mirip, yaitu hasil deteksi lebih dari 25% sampai 50% berarti dapat dikatakan bahwa dokumen tersebut cukup banyak memiliki kemiripan. Indikasi warna persentase adalah warna kuning.
4. Mirip, yaitu hasil deteksi lebih dari 50% sampai 75% berarti dapat dikatakan bahwa dokumen tersebut memiliki banyak kemiripan. Indikasi warna persentase adalah warna oranye.
5. Sangat Mirip, yaitu hasil deteksi lebih dari 75% sampai 100% berarti dokumen tersebut mendekati plagiat. Dokumen harus diperiksa manual, agar mendapatkan hasil yang lebih valid. Indikasi warna persentase adalah warna merah.

### B. Hasil Pengujian Sistem

Pengujian dilakukan pada sistem dengan kuesioner dimana pengujian dilakukan oleh pengguna secara langsung untuk memeriksa apakah sistem dapat berjalan saat digunakan oleh pengguna. Pengujian ini melibatkan data *real* yang didapat secara langsung melalui kuesioner. Kuesioner berisikan 16 pertanyaan dan diberikan kepada 100 mahasiswa. Kuesioner untuk bagian aspek rekayasa perangkat lunak memiliki 6 pertanyaan sedangkan kuesioner untuk bagian persentase kemiripan memiliki 10 pertanyaan.

*Aspek Rekayasa Perangkat Lunak*

Hasil dari kuesioner yang diisi oleh responden terhadap aspek rekayasa perangkat ditunjukkan pada tabel berikut ini.

Tabel 1  
Hasil Kuesioner Rekayasa Perangkat Lunak

Aspek Rekayasa Perangkat Lunak	Tanggapan					Total
	1	2	3	4	5	
Kemudahan menjalankan aplikasi	0	0	12	71	17	100
Kelancaran menjalankan aplikasi	0	0	11	70	19	100
Kemudahan mengakses fitur-fitur aplikasi	0	0	16	72	12	100
Kelengkapan fitur	0	1	40	48	11	100
Memudahkan pengguna melakukan pendeteksian kemiripan teks atau dokumen	0	0	6	58	36	100
Aplikasi membantu mempercepat pengecekan kemiripan teks/dokumen	0	0	5	58	37	100
Total	0	1	90	377	132	600
Persentase	0	0,17	15	62,83	22	100

Keterangan : 1 = Sangat buruk 4 = Baik  
2 = Buruk 5 = Sangat baik  
3 = Cukup baik

Berdasarkan pada Tabel 1 tersebut dapat diketahui bahwa sebanyak 62,83% responden menilai aplikasi ini baik dengan jumlah tanggapan sebanyak 377 tanggapan. Terdapat 1 tanggapan buruk dengan persentase 0,17% dari responden pada penilaian kelengkapan fitur yang mungkin saja dikarenakan responden tersebut merasa aplikasi yang dibangun belum memiliki fitur yang lengkap.

*Persentase Kemiripan*

Sebelumnya telah dilakukan 10 percobaan pendeteksian menggunakan aplikasi yang dibuat. Masing-masing percobaan tersebut menggunakan teks uji dan teks pembanding yang berbeda. Berikut ini adalah teks uji dan teks pembanding dari setiap percobaan.

Tabel 2  
Kalimat Uji dan Kalimat Pembanding

No	Teks Uji	Teks Pembanding
1	Dampak kekerasan terhadap anak usia dini yang disebabkan oleh orang tua	Orang tua yang selalu melakukan kekerasan terhadap anak usia dini
2	Pengelolaan lingkungan dan kondisi masyarakat pada suatu wilayah	Dampak kepadatan masyarakat terhadap lingkungan
3	Lingkungan akan mempengaruhi emosional anak	Pengaruh kebudayaan terhadap pertumbuhan seorang anak
4	Hutan gundul akibat penebangan liar	Penanaman kembali menjadikan hutan tetap hijau
5	Setiap orang akan berusaha untuk menjadi orang sukses	Jangan menyerah ketika mengalami kegagalan saat berusaha
6	Implementasi Sistem Enkripsi Pengirim Pesan Instan Java	Studi dan Implementasi Enkripsi Pengiriman Pesan

No	Teks Uji	Teks Pembanding
		Suara
7	Berikan anak makanan yang bergizi dan bervitamin agar selalu sehat	Jagalah pola makan yang teratur untuk tetap sehat
8	Motor merupakan salah satu kendaraan bermotor beroda dua	Mobil merupakan salah satu kendaraan bermotor beroda empat
9	Upaya pencegahan penyalahgunaan narkoba secara komprehensif	Pencegahan perilaku penyalahgunaan narkoba bagi pelajar
10	Banjir diakibatkan oleh lingkungan yang kurang bersih	Hujan yang deras dalam waktu lama dapat mengakibatkan banjir

Kemudian hasil dari percobaan yang telah dilakukan dimasukkan kedalam kuesioner. Hasil dari kuesioner yang diisi oleh responden terhadap hasil persentase kemiripan yang maksudnya adalah persetujuan sesuai atau tidaknya penalaran responden dengan hasil persentase kemiripan aplikasi ditunjukkan pada tabel berikut ini.

Tabel 3  
Hasil Kuesioner Persentase Kemiripan

Pendeteksian	Keluaran Sistem	Tanggapan					Total
		1	2	3	4	5	
Percobaan 1	61,82 % Mirip	18	20	19	31	12	100
Percobaan 2	49,28 % Cukup mirip	9	20	24	43	4	100
Percobaan 3	37,04 % Sedikit mirip	7	11	31	38	13	100
Percobaan 4	15,87 % Tidak mirip	6	7	11	35	41	100
Percobaan 5	16,67 % Tidak mirip	7	13	9	40	31	100
Percobaan 6	66,67 % Mirip	3	9	34	43	11	100
Percobaan 7	29,03 % Sedikit mirip	5	16	24	42	13	100
Percobaan 8	81,82 % Sangat mirip	9	9	21	50	11	100
Percobaan 9	53,16 % Cukup mirip	3	12	29	49	7	100
Percobaan 10	32,26 % Sedikit mirip	10	21	18	40	11	100
Total		77	138	220	411	154	1000
Persentase		7,7	13,8	22	41,1	15,4	1000

Keterangan : 1 = Tidak setuju 4 = Setuju  
2 = Kurang setuju 5 = Sangat setuju  
3 = Cukup setuju

Berdasarkan pada Tabel 3 dapat diketahui bahwa hampir sebagian besar responden setuju dengan hasil persentase kemiripan keluaran aplikasi yang dibuktikan dengan 41,1% responden setuju dengan jumlah tanggapan sebanyak 411 tanggapan. Terdapat 154 tanggapan dengan persentase 15,4% yang menilai sangat setuju dengan hasil persentase keluaran

aplikasi dan sebanyak 220 tanggapan dengan persentase 22% yang menilai cukup setuju dengan hasil persentase keluaran aplikasi. Namun selain responden sangat setuju, setujuan cukup setuju dengan hasil persentase keluaran aplikasinya, terdapat 134 tanggapan dengan persentase 13,4% yang kurang setuju dengan hasil persentase keluaran aplikasi dan 77 tanggapan dengan persentase 7,7% tidak setuju dengan hasil persentase keluaran aplikasi.

### C. Analisis Hasil Pengujian

Berdasarkan hasil pengujian yang diperoleh, analisis mengenai masing-masing hasil pengujian dapat disimpulkan sebagai berikut.

1. Hasil pengujian *White Box* untuk kode program *case folding, tokenizing, filtering, k-gram, hashing*, Algoritma *Rabin-Karp* dan *Dice's Coeficient Similarity* menunjukkan bahwa dalam penerapan setiap tahapan pendeteksian kemiripan telah berhasil dengan *error nol*, dimana setiap satu rangkaian pernyataan proses pada program telah dieksekusi paling tidak satu kali selama pengujian dan semua kondisi logis telah diuji dan berhasil.
2. Hasil pengujian validitas, kuesioner yang berisikan 16 pertanyaan dan diberikan kepada 100 mahasiswa menunjukkan bahwa semua pertanyaan dalam kuesioner dinyatakan valid karena nilai korelasi setiap pertanyaan lebih dari nilai *r* tabel sebesar 1,654 untuk digunakan dalam pengumpulan data. Begitu pula dengan kuesioner untuk bagian persentase kemiripan yang berisikan 10 pertanyaan diberikan kepada 100 mahasiswa menunjukkan bahwa semua pertanyaan dalam kuesioner dinyatakan valid karena nilai korelasi setiap pertanyaan lebih dari nilai *r* tabel sebesar 1,654 untuk digunakan dalam pengumpulan data.
3. Hasil pengujian reliabilitas, nilai *alpha* kuesioner sebesar 0,606 yang mengartikan bahwa tingkat reliabilitas moderat dan dapat dipercaya. Begitu pula dengan kuesioner untuk bagian persentase kemiripan memiliki nilai *alpha* sebesar 0,622 yang mengartikan bahwa tingkat reliabilitas moderat dan dapat dipercaya.
4. Hasil pengujian *User Acceptance Test* untuk Aspek Rekayasa Perangkat Lunak dengan pengukuran skala menggunakan *Likert's Summated Rating (LSR)* menunjukkan beberapa responden yang menggunakan aplikasi menilai aplikasi sangat positif dan dinilai berhasil yang dibuktikan dengan skor total dari keseluruhan data kuesioner berjumlah 2440 yaitu berada di antara kuartil III dan Maksimal.
5. Hasil pengujian *User Acceptance Test* untuk Persentase Kemiripan dengan pengukuran skala menggunakan *Likert's Summated Rating (LSR)* menunjukkan beberapa responden yang menggunakan aplikasi menilai aplikasi sangat positif dan dinilai berhasil yang dibuktikan dengan skor total dari keseluruhan data kuesioner berjumlah 3427 yaitu berada di antara median dan kuartil III.
6. Berdasarkan hasil pengisian kuesioner, terdapat pernyataan kurang setuju dan tidak setuju pada kuesioner tentang persentase kemiripan dapat mengartikan bahwa

pendeteksian kemiripan menggunakan Algoritma *Rabin-Karp* masih belum berjalan sesuai dengan penalaran pikiran manusia pada umumnya. Algoritma *Rabin-Karp* mendeteksi kemiripan dengan berdasarkan pola huruf yang digunakan pada suatu teks. Dua kata dapat dikatakan mirip ketika kata yang menyusunnya tersebut menggunakan pola huruf yang sama. Contohnya adalah kata "pernah" yang ketika melewati proses pemotongan dengan 3-gram akan menjadi "per", "ern", "rna", "nah" dan kata "pergi" akan menjadi "per", "erg", "rgi" yang seharusnya kata yang berbeda namun apabila menggunakan pendeteksian kemiripan dengan Algoritma *Rabin-Karp* akan dikatakan memiliki kemiripan dikarenakan terdapat 1 pola yang sama yaitu "per".

## V. KESIMPULAN/RINGKASAN

Berdasarkan hasil analisis dan pengujian terhadap aplikasi pendeteksi plagiat karya ilmiah dengan menggunakan Algoritma *Rabin-Karp*, dapat disimpulkan bahwa.

1. Algoritma *Rabin-Karp* dapat diimplementasikan untuk mendeteksi kemiripan antar teks atau dokumen.
2. Algoritma *Rabin-Karp* mendeteksi kemiripan tidak berdasarkan dengan kemiripan setiap kata namun dengan melakukan pendeteksian kemiripan setiap *hash* dari *pattern* antar kedua karya tulis yang merupakan hasil dari proses *Hashing* dan *K-Gram*.
3. Berdasarkan hasil dari kuesioner yang telah diisi oleh 100 orang responden, sebagian besar responden setuju bahwa aplikasi yang dibangun dapat meningkatkan keefektifitasan waktu dalam melakukan proses pendeteksian kemiripan karya ilmiah pengguna. Sebanyak 37 orang sangat setuju, 58 orang setuju dan 5 orang cukup setuju.
4. Berdasarkan hasil dari kuesioner yang telah diisi oleh 100 orang responden dengan 10 pertanyaan yang diajukan mengenai hasil persentase kemiripan, sistem dapat dikatakan sebagai alat bantu untuk mempertimbangkan dalam menentukan plagiat atau tidaknya suatu karya ilmiah karena sebagian besar responden setuju dengan setiap hasil persentase kemiripan dari sistem. Sebanyak 15,4% responden sangat setuju dengan hasil keluaran sistem, 41,1% responden setuju dengan hasil keluaran sistem dan 22% responden cukup setuju dengan hasil keluaran sistem. Sisanya sebanyak 13,8% responden kurang setuju dan 7,7% responden tidak setuju dengan hasil keluaran sistem.
5. Hasil pengujian *User Acceptance Test* untuk aspek rekayasa perangkat lunak dengan pengukuran skala menggunakan *Likert's Summated Rating (LSR)* menunjukkan beberapa responden yang menggunakan aplikasi menilai aplikasi sangat positif dan dinilai berhasil yang dibuktikan dengan skor total dari keseluruhan data kuesioner berjumlah 2440 yaitu berada di antara kuartil III dan Maksimal.
6. Hasil pengujian *User Acceptance Test* untuk persentase kemiripan dengan pengukuran skala menggunakan *Likert's Summated Rating (LSR)* menunjukkan beberapa responden yang menggunakan aplikasi menilai aplikasi



positif dan cukup berhasil yang dibuktikan dengan skor total dari keseluruhan data kuesioner berjumlah 3427 yaitu berada di antara median dan kuartil III.

## VI. DAFTAR PUSTAKA

- [1] Susilo, Madyo Eko dan Triyanto, Bambang. 1995, *Pedoman Penulisan Karya Ilmiah*. Semarang: Dahara Prize.
- [2] Neville, Colin. 2010. *The Complete Guide Referencing and Avoiding Plagiarism*. Maidenhead : Open University Press/McGraw Hill.
- [3] Junaidi dan Alfiah, Fifit. 2014. *Colaborative Methods Model Dalam Membandingkan Dokumen Untuk Mengukur Prosentase Kemiripan*. Konferensi Nasional Sistem Informasi 2014. STMIK Dipanegara-Makassar.
- [4] Departemen Pendidikan dan Kebudayaan. 1997. *Kamus Besar Bahasa Indonesia*. Jakarta: Balai Pustaka.
- [5] Triawati, Chandra. 2009. *Metode Pembobotan Statistical Concept Based untuk Klustering dan Kategorisasi Dokumen Berbahasa Indonesia*. Bandung: Institut Teknologi Telkom.
- [6] Yoga, Kadek Versi Yana. 2012. *Pengembangan Aplikasi Pendeteksi Plagiarisme Pada Dokumen Teks Menggunakan Algoritma Rabin-Karp*. Bali: Fakultas Teknik dan Kejuruan Universitas Pendidikan Ganesha.
- [7] Firdaus, Hari Bagus. 2008. *Deteksi Plagiat Dokumen Menggunakan Algoritma Rabin-Karp*. Bandung: Program Studi Teknik Informatika Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung (ITB).
- [8] Turnitin. 2013. *Interpreting Turnitin Originality Reports*. <https://eat.scm.tees.ac.uk/bb8content/resources/recipes/interpretTurnitin.pdf>.