BAB II

LANDASAN TEORI

2.1 Plagiarisme

Plagiarisme adalah penyalahgunaan hak kekayaan intelektual milik orang lain dan karya tersebut diakui secara tidak sah sebagai hasil karya pribadi (Sulianta,2017). Seseorang yang menjadikan ide dari orang lain menjadi milik sendiri tanpa mencantumkan sumbernya disebut sebagai plagiator. Tindakan kriminal yang memalsukan pekerjaan orang lain juga termasuk sebagai plagiat. Disebut sebagai plagiat jika tingkat persentase kesamaan dari teks yang dicocokan adalah di atas 25% sedangkan batas penerimaannya adalah di bawah 15% (tees, 2019) Dalam kenyataannya, plagiarisme tidak selalu dilakukan secara sengaja. Beberapa orang melakukan plagiarisme dikarenakan adanya kekurangan informasi atau referensi dalam membuat pekerjaan saintifik. Berikut beberapa contoh tipe – tipe dari plagiarism (Putri & Siahaan, 2017).

1. Kebetulan

   Terjadi karena kurangnya pengertian akan plagiarism dan referensi dalam penulisan. Biasanya terjadi ketika menulis makalah saintifik tidak didasari pada tinjauan literatur.

2. Tidak Disengaja

   Ketika seseorang melakukan penulisan kembali mengenai informasi yang biasanya sering dibahas dan ditulis dengan kata–kata yang hampir sama. Ide yang sama dapat dihasilkan berbeda jika ditulis dengan sedemikian rupa sehingga plagiarisme dapat dihindari.

3. Disengaja

Sebuah kegiatan yang dengan sengaja mengambil kalimat atau keseluruhan pekerjaan orang lain tanpa sitasi yang merujuk pada kreator aslinya.

4. Plagiarisme sendiri

   Penggunaan kembali pekerjaan sendiri tanpa melakukan pengembangan kembali dari nilai – nilai atau variabel yang berada dari pekerjaan sebelumnya.

Pendeteksian dari plagiarisme dapat terbagi menjadi dua bagian, *fingerprinting* dan *full-text comparison* (Putri & Siahaan, 2017).

1. *Fingerprinting Comparison*

   Sebuah teknik untuk memeriksa hubungan dari dua dokumen dengan menggunakan semua teks yang terkandung dari kedua dokumen. Teknik ini akan membagi kata-kata dalam dokumen menjadi karakter–karakter dengan panjang tertentu. Teknik ini disebut sebagai *hashing*. Algoritma yang paling sering dipakai adalah Rabin-Karp.

2. *Full-text Comparison*

   Sebuah teknik yang membandingkan kedua dokumen dengan membandingkan teks satu persatu pada masing masing dokumen. Kekurangannya adalah akan memakan waktu yang lama dalam mencocokan dokumen yang berukuran besar. Perbandingan teks yang kompleks tidak dapat dilakukan pada dokumen yang tidak terdapat pada penyimpanan yang sama.

2.2 Rabin Karp

Rabin-Karp adalah algoritma yang digunakan untuk melakukan pencarian pola pada substring di dalam teks menggunakan *hashing* (Putri & Siahaan, 2017). Sangat efektif untuk penyamaan ragam pola kata. Salah satu aplikasi praktis dari algoritma ini adalah untuk deteksi plagiarisme. Rabin-Karp bergantung pada fungsi *hash* untuk menentukan persentase dari plagiarisme. Tingkat keakurasian bisa diatur berdasarkan fitur *hash* ini. Fungsi *hash* adalah sebuah fungsi yang menentukan nilai dari kalimat-kalimat tertentu. Fungsi *hash* akan mengubah setiap string menjadi angka, yang disebut sebagai nilai *hash*. Algoritma Rabin-Karp menentukan nilai *hash* berdasarkan kata yang sama.

Hashing merupakan fungsi paling penting dalam algoritma *Rabin-Karp*. Hasil dari *hashing* diperoleh dari perkalian antara nilai ASCII dengan angka yang sudah ditentukan sebagai basis. Rabin-Karp memiliki ketentuan sebagai berikut (Putri & Siahaan, 2017).

1. Jika kedua string adalah string yang sama maka hasil nilai *hash*nya akan sama juga.
2. K-Gram ditentukan dari uji coba menyesuaikan panjang string kata yang akan di-*hash*.
3. Basis ditentukan sendiri untuk menyesuaikan panjang hasil *hash* yang diinginkan.

Rabin-Karp memiliki proses sebagai berikut (Putri & Siahaan, 2017).

1. *Hashing*

   Proses ini merubah kata–kata menjadi nilai *hash* yang kemudian nilai ini dibandingkan. Berikut contoh dari proses *hash* algoritma Rabin-Karp. Diasumsikan katanya adalah TANGERANG.

$Hash = (A(1)*Basis^{K\text{-}gram-1}) + (A(2)*Basis^{K\text{-}gram-2}) + ... + (A(K\text{-}gram)*Basis^{K\text{-}gram-K\text{-}gram})$ ...(2.1)

K-Gram = Jumlah Karakter Kata

Basis = Nilai Pengali

A = Kata yang akan di *Hash*

K-Gram = 9

Basis = 3

A = TANGERANG

A(1) = 84

A(2) = 65

A(3) = 78

A(4) = 71

A(5) = 69

A(6) = 82

A(7) = 65

A(8) = 78

A(9) = 71

$Hash = (84 * 3^8)+(65 * 3^7) +(78 * 3^6)+(71 * 3^5)+(69 * 3^4)+(82 * 3^3) +(65 * 3^2)+(78 * 3^1) +(71 * 3^0) = 776087$

2. Perbandingan Hasil *Hash*

Hasil *hash* kemudian disusun dalam tabel hingga semua list kata–kata dari dokumen terisi penuh. Sebagai contoh berikut tabel simulasi replika dari dua dokumen yang sudah dilakukan proses *hash*.

19875	16830	23124	17433	20546
21489	26753	13498	23846	16528
21848	28447	29994	10301	13009
18832	27217	23157	25854	22492
14952	14337	29348	19978	28809
13485	14188	13131	21215	12053
25669	13809	26508	19455	25356
29964	17723	26633	17445	11803
19477	27142	24814	15155	26266
28432	19007	21896	16625	20681

Gambar 2.1 Nilai *hash* dari dokumen pertama

(Sumber: penelitian Putri dan Ranti tahun 2017*)*

28432	26406	28424	13930	19187
18049	10867	18516	26753	19975
10152	13053	24120	21896	18351
12605	25101	21215	20750	15513
22949	26006	25045	25932	10695
13254	21504	20286	22492	10615
25565	29941	17403	23018	22666
19744	19769	19877	29535	13139
25669	16830	14297	20916	24640
16960	20681	13131	13009	18947

Gambar 2.2 Nilai *hash* dari dokumen kedua

(Sumber: penelitian Putri dan Ranti tahun 2017*)*

Setelah dilakukan pencocokan terdapat 10 nilai *hash* yang sama dari kedua tabel tersebut. Langkah selanjutnya adalah melakukan perhitungan persentase kesamaan dari dua dokumen di atas. Rumus yang digunakan untuk menghitung adalah sebagai berikut.

$$P = \frac{2*SH}{THA+THB} * 100\% \quad ...(2.2)$$

P = Persentase Kesamaan Dokumen

SH = Jumlah *Hash* yang Identik

THA = Total *Hash* di Dokumen A

THB = Total *Hash* di Dokumen B

Dilakukan perhitungan dari tingkat kesamaan antar kedua dokumen menggunakan rumus 2 sebagai berikut:

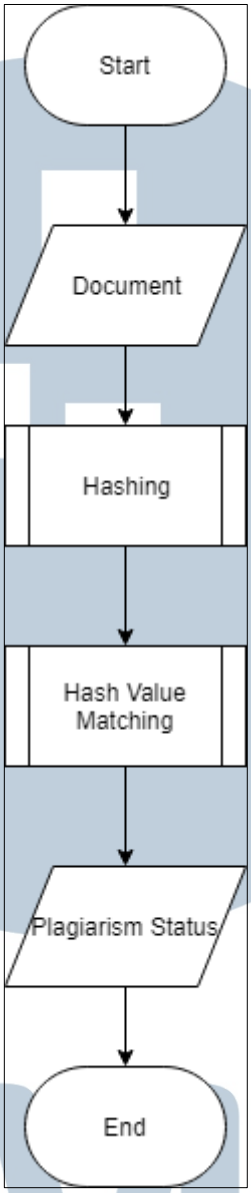
$$P = \frac{2*10}{50+50} * 100\%$$

$$= 20/100 * 100\%$$

$$= 0.5 * 100\%$$

$$= 20\%$$

Sehingga tingkat kesamaan dari kedua dokumen didapat 20%, yang berarti berada di bawah batas dari tingkat plagiarisme yaitu 25%. Berikut *flowchart* dari algoritma Rabin-Karp.



Gambar 2.3 *Flowchart Rabin-Karp*

2.3 Confusion Matrix

Confusion Matrix adalah suatu metode yang banyak digunakan untuk mengukur kinerja dari sistem atau metode yang digunakan. *Precision* adalah tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh system. Sedangkan, *recall* adalah tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi.

Tabel 2.1 *Confusion Matrix*

(Sumber: penelitian David, Bowes, dkk*)*

	Observe True	**Observe False**
Predicted True	TP (True Positive) Correct result	FP (False Positive) Unexpected result
Predicted False	FN (False Negative) Missing result	TN (True Negative) Correct absence of result

$$Precision = \frac{TP}{TP+FP} \quad \ldots(2.3)$$

$$Recall = \frac{TP}{TP+FN} \quad \ldots(2.4)$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad \ldots(2.5)$$

Sebagai contoh akan mengukur kinerja dari sebuah mesin pemisah dokumen yang bertugas memisahkan dokumen txt dari semua dokumen yang telah didapat. Untuk mengujinya kita akan memasukkan 100 dokumen txt dan 900 dokumen lain (bukan dokumen txt). Hasilnya mesin memisahkan 110 yang dideteksi sebagai dokumen txt. Ke 110 dokumen tersebut kemudian dicek kembali oleh manusia, ternyata dari 110 dokumen tersebut hanya 90 dokumen yang merupakan dokumen txt, sedangkan 20 lainnya merupakan dokumen lain. Dalam kasuus tersebut dapat simpulkan bahwa mesin tersebut memilki *precision* sebesar 82%, *recall* 90%, dan *accuracy* 97% yang didapatkan dari perhitungan berikut.

Tabel 2.2 *Confusion Matrix* pemisah dokumen

	Observe True	**Observe False**
Predicted True	90	20
Predicted False	10	880

$$Precision = \frac{90}{110} = 0.82 = 82\%$$

$$Recall = \frac{90}{100} = 0.9 = 90\%$$

$$Accuracy = \frac{90+880}{1000} = 0.97 = 97\%$$