Deteksi Kemiripan Halaman pada Al-Qur'an dengan Menggunakan Algoritma Rabin Karp dan Jaccard Similarity

Winda Eka Samodra[1], Moch Arif Bijaksana[2].

[1,2,3]Fakultas Informatika, Universitas Telkom, Bandung
[1]samoded@student.telkomuniversity.ac.id, [2]arifbijaksana@telkomuniversity.ac.id,

Abstrak

Plagiarisme merupakan suatu tindakan yang dilakukan untuk mendapatkan pengakuan secara ilmiah tanpa memberikan sumber asli. Di lain sisi plagiarisme dapat digunakan untuk mencari *similarity* antara satu dokumen dengan dokumen yang lain. Cukup sulit mengukur kesamaan dokumen hanya dengan membaca atau mengukurnya secara manual. Namun konsep plagiarisme ini dapat digunakan untuk mendeteksi kemiripan antar ayat pada Al-Qur'an. Dataset yang digunakan merupakan sekumpulan ayat pada halaman Al-Qur'an. Oleh karena itu, dibutuhkan suatu sistem yang dapat melakukan deteksi kemiripan. Untuk membangun sistem tersebut, pada tugas akhir ini menggunakan metode *Rabin Karp* dengan diolah menggunakan parsing *N-gram* pada setiap dokumen inputnya. Data akan dicari nilai similarity diantara ayat kesatu pada halaman pertama dengan semua ayat pada halaman kedua. Selajutnya akan diambil nilai kemiripan yang bernilai 1 dan 0. Sistem yang dibangun menggunakan salah satu metode kemiripan yaitu, *Jaccard Similarity*. Dengan menggunakan teknik hashing dan menggunakan Algoritma *Rabin Karp* sistem dapat menghasilkan nilai *precision* terbaik sebesar 32,76 % dan menghasilan *f-Measure* sebesar 39,81%.

Kata kunci : Plagiarisme, Al-Quran, Rabin Karp, Jaccard Similarity

Abstract

Plagiarism is an action to get scientific recognition without providing an original source. On the other side, plagiarism can be used to find similarity between one document to another. It's quite difficult to measure the similarity of documents only by reading or measuring them manually. But this concept of plagiarism can be used to detect similarities between verses in the Qur'an. The dataset used is a set of verses on the pages of the Qur'an. Therefore, a system is needed that can carry out similarity detection. To build the system, this final project uses the Rabin Karp method to be processed using N-gram parsing on each input document. The data will be searched for the similarity value between paragraphs one on the first page with all the verses on the second page. Next, a similarity value of 1 and 0 will be taken. The system is built using one of the similarity methods namely, Jaccard Similarity. By using hashing techniques and using the Rabin Karp Algorithm the system can produce the best precision values of 77.4% and produce f-Measure of 55.8%.

Keywords: Plagiarism, Qur'an, Rabin Karp, Jaccard Similarity

1. Pendahuluan

Meniru atau menjiplak merupakan salah satu kebiasaan dasar manusia dalam proses petumbuhan menjadi dewasa. Plagiarisme merupakan suatu tindakan yang dilakukan untuk mendapatkan pengakuan secara ilmiah tanpa memberikan sumber asli. Konsep plagiarisme dapat digunakan untuk mendapatkan nilai similarity diantara dua dokumen yang berbeda. Karena cukup sulit mengukur kesamaan dokumen hanya dengan membaca atau mengukurnya secara manual. Metode ini termasuk dalam salah satu metode pendeteksi plagiarisme bagian *fingerprint*. Yang terbagi menjadi Perbandingan Teks Lengkap, Dokumen *Fingerprint*, dan Kesamaan Kata Kunci [1]. Untuk Dataset atau data masukkan yang digunakan adalah sekumpulan ayat pada halaman Al-Quran yang terpilih. Sistem ini memungkinkan untuk mengetahui kedekatan makna antar ayat ayat pada halaman Al-Quran. Al-Quran sendiri memiliki 30 Juz, 114 Surat dan 6236 ayat.

Dalam membangun sistem ini diperlukan beberapa tahapan penting yaitu, *Hashing* dan *Jaccard Coeffisient*. Selain itu sistem ini menggunakan Algoritma *Rabin Karp* untuk proses *fingerprint*. Algoritma *Rabin-Karp* tidak bertujuan menemukan string yang cocok dengan string masukan, melainkan menemukan pola (*pattern*) yang sekiranya sesuai dengan teks masukan [2]. Kunci utama penggunaan algoritma rabin karp adalah perhitungan yang efisien terhadap nilai *hash substring* [3]. Data input yang dipakai harus melewati peroses *preprocessing* terlebih dahulu. *Preprocessing* digunakan untuk menghilangkan *punctuation*, *case folding*, dan *whitespace insentivity*. Lalu data akan di *Hashing* untuk mendapatkan nilai *ascii* dari data masukkan. Ascii merupakan kode standar amerika yang biasanya digunakan untuk pertukaran informasi berupa angka. *Hashing* adalah metode yang menggunakan fungsi *hash* untuk mengubah suatu jenis data menjadi beberapa bilangan bulat sederhana [2]. Pada tahapan akhir dari sistem ini merupakan proses medapatkan nilai antara 2 dokumen yang dibandingkan, yaitu

halaman pertama dan halaman kedua. Hasil dari perbandingan tersebut merupakan nilai *similarity*. Proses pencarian nilai *similarity* menggunakan *Jaccard Coeffisient*.

Sistem ini dibangun untuk mendapatkan hasil analisis dari sebuah sistem yang dapat menghasilkan nilai *similarity* terhadap 2 dokumen yang dibandingkan dengan menggunakan algoritma *Rabin Karp* dengan karakter *N-Gram*. Selanjutnya adalah mendapatkan nilai performansi dari sistem yang dibangun, serta pengaruh *N-gram* terhadap performansi sistem. Untuk batasan masalah yang digunakan pada Tugas Akhir antara lain, sistem yang dibangun hanya menggunakan data masukkan dengan bahasa inggris, sistem tidak memperhatikan kesalahan ejaan atau penulisan pada dokumen, *output* yang dihasilkan oleh sistem merupakan nilai kemmiripan dari dua dokumen yang dibandingkan, serta nilai dari *precision*, *recall*, akurasi, dan *F-Measure* yang dihasilkan oleh sistem.

2. Studi Terkait

a. Plagiarisme

Plagiarisme merupakan tindakan penyalahgunaan, pencurian/perampasan, penerbitan, pernyataan, atau menyatakan sebagai milik sendiri sebuah pikiran, ide, tulisan, atau ciptaan yang sebenarnya milik orang lain [4]. Plagiarisme dapat dianggap sebagai tindak kejahatan yang bisa mendapatkan tindak pidana pada seseorang yang melakukan. Ada berbagai 4 jenis plagiarisme. Pertama plagiarisme total, kedua plagiarisme parsial, ketiga auto-plagiasi (*self plagiarism*), dan keempat plagiarisme antarbahasa [5]. Plagiarisme sering dinyatakan menyalin pekerjaan orang lain dan lalai untuk memberikan pengakuan dari sumber (pencetus bahan yang ditiru) [6]. Sebuah penelitian yang dilakukan oleh McCabe pada tahun 2005 dikemukakan bahwa 70% siswa mengakui melakukan plagiarisme dimana setengahnya merasa bersalah melakukan kecurangan pada tugas tertulis 40% siswa mengaku menggunakan metode "*cut-paste*" saat menyelesaikan tugas mereka.

Pada metode pendeteksi plagiarisme dapat dikategorikan menjadi 3 bagian antara lain, Perbandingan Teks Lengkap, Dokumen *Fingerprinting*, Kesamaan Kata Kunci [1]. Perbandingan Teks Lengkap merupakan sebuah metode yang digunakan untuk membadingkan semua isi dokumen yang sudah dipilih sebelumnya, serta metode ini dapat diterapkan untuk dokumen dengan skala yang besar. Dokumen *Fingerprinting* merupakan suatu metode yang digunakan untuk mendeteksi keakuratan salinan antar dokumen, baik semua teks yang terdapat di dalam dokumen atau hanya sebagian teks saja. Prinsip kerja dari metode dokumen *fingerprinting* ini adalah dengan menggunakan teknik *hashing*. Prinsip dari metode Kesamaan Kata Kunci ini adalah mengekstrak kata kunci dari dokumen dan kemudian di bandingkan dengan kata kunci pada dokumen yang lain. Pendekatan yang digunakan pada metode ini adalah teknik dot.

b. Algoritma *Rabin-Karp*

Algoritma Rabin-Karp adalah suatu algoritma pencarian string yang ditemukan oleh Michael Rabin dan Richard Karp [7]. Algoritma ini menggunakan *hashing* untuk menemukan sebuah substring dalam sebuah teks. Algoritma Rabin-Karp menghasilkan efisiensi waktu yang baik dalam mendeteksi string yang memiliki lebih dari satu pola. Hal inilah yang membuat algoritma *Rabin-Karp* digunakan untuk melakukan pendeteksian kesamaan terhadap dokumen yang dibandingkan.

```
1 function RabinKarp(string s[1..n], string pattern[1..m])
2   hpattern := hash(pattern[1..m]);
3   for i from 1 to n-m+1
4     hs := hash(s[i..i+m-1])
5     if hs = hpattern
6       if s[i..i+m-1] = pattern[1..m]
7         return i
8   return not found
```

Gambar 1

Pseudocode* dari Algoritma *Rabin Karp

c. *Hashing*

Hashing adalah suatu cara untuk mentransformasi sebuah *string* menjadi suatu nilai yang unik dengan panjang tertentu yang berfungsi sebagai penanda *string* tersebut. Fungsi untuk menghasilkan nilai ini disebut fungsi *hash*. Lalu dari fungsi tersebut akan mengeluarkan sebuah nilai yang dapat disebut dengan nilai *hash*. *Hashing* sendiri berguna untuk medapatkan *string* pada pencarian database yang sudah disediakan. Apabila tidak di-*hash*, pencarian akan dilakukan karakter per karakter pada setiap *string*. Namun nilai *hashing* ini akan lebih baik jika ditambahkan karakter *N-Gram* pada sistem yang dibuat. Nilai *hash* pada umumnya digambarkan sebagai *fingerprint* yaitu suatu string pendek yang terdiri atas huruf dan angka yang terlihat acak (data biner yang ditulis dalam heksadesimal).

Untuk mentranformasi dari rangkaian *string* menjadi rangkaian nilai hash menggunakan *rolling hash*. *Rolling hash* memungkinkan untuk menghitung nilai *hash* tanpa melakukan *rehashing* kembali dari iterasi pertama.

$$H(C_1.....C_n) = C_1 * B^{k-1} + C_2 * B^{k-2} * ..... + C_{n-1} * B + C_k \quad [8]$$

Untuk menghitung nilai *hash* pertama dilakukan perhitungan menggunakan rumus diatas. $H(C_1.....C_n)$ Merupakan nilai hash pertama, dimana b adalah konstan bilangan prima, k adalah nilai kgram dan C_n merupakan nilai *ascii* dari karakter. Pada perhitungan nilai *hash* kedua dan seterusnya tidak perlu dilakukan perhitungan lagi dari iterasi pertama namun dengan melakukan perhitungan rumus dibawah.

$$H(C_1.....C_{n+1}) = (H(C_{1.....}C_n) - C_1 * B^{(n-1)})* B + C_{(n+1)} [8]$$

Keterangan:
c : nilai ASCII karakter
b : basis (bilangan prima)
l : banyak karakter atau panjang *string*

d. Jaccard Similarity Coeffisient

Jaccard Similarity Coefficient merupakan salah satu dari banyak *measure dissimilarity* yang dapat digunakan untuk menghasilkan suatu nilai dari perbandingan 2 dokumen. Indeks *Jaccard*, juga dikenal sebagai *Intersection over Union* dan koefisien kesamaan *Jaccard* (koefisien *coautical coined* yang diciptakan oleh Paul *Jaccard*), adalah statistik yang digunakan untuk membandingkan kemiripan dan keragaman set sampel. Koefisien *Jaccard* mengukur kesamaan antara set sampel terbatas, dan didefinisikan sebagai ukuran perpotongan dibagi dengan ukuran penyatuan set sampel [9]. Berikut merupakan rumus yang dimiliki oleh *Jaccard Similarity Coefficient*.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \quad [10]$$

Keterangan:
A : Kumpulan data pada array A
B : Kumpulan data pada array B

e. Rumus *Precision, Recall,* Akurasi, dan *F-Measure*

Didalam perhitungan untuk mencari nilai akurasi, banyak cara yang dapat dilakukan untuk mendapatkan performansi dari suatu sistem. Pada Tugas Akhir ini, sistem menggunakan 4 jenis perhitungannya. Antara lain adalah *Precision*, *Recall*, Akurasi, dan *F-Measure*. Berikut adalah beberapa rumus umum dengan analogi sistem yang dibuat.

$$Precision = \frac{TP}{TP + FP} \quad [11]$$

$$Recall = \frac{TP}{TP+FN} \quad [11]$$

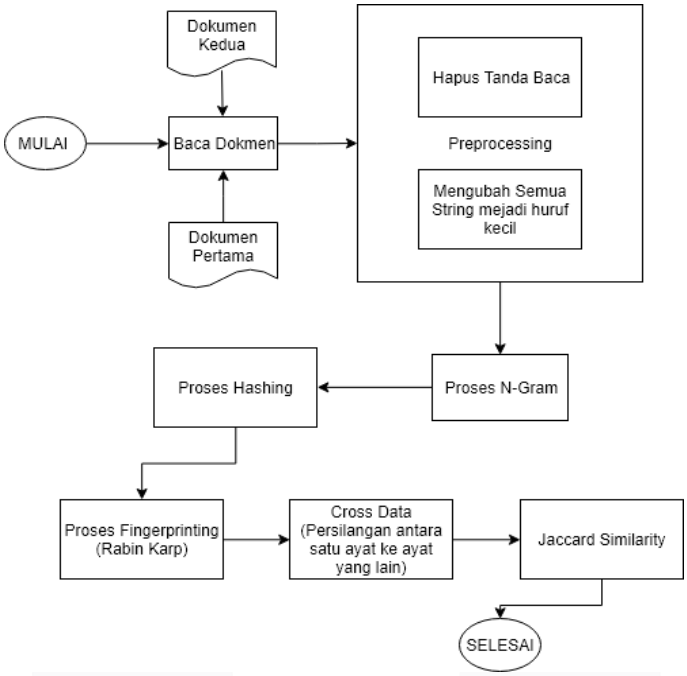
$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad [11]$$

Dimana:
Tp = *true positive*, yaitu jumlah data yang diprediksi oleh sistem dan dalam kenyataannya bernilai *positive*.
Tn = *true negative*, yaitu jumlah data yang diprediksi oleh sistem dan dalam kenyataannya bernilai *negative*.
Fp = *false positive*, yaitu jumlah data yang diprediksi *positive* oleh sistem namun kenyataanya bernilai *negative*.
Fn = *false negative*, yaitu jumlah data yang diprediksi *negative* oleh sistem namun dalam kenyataannya bernilai *positive*.

$$F\text{-}Measure = 2\frac{Precision\ .\ Recall}{Precision+Recall} \quad [12]$$

3. Sistem yang Dibangun

Untuk membangun sistem pada tugas akhir ini, memerlukan beberapa tahapan untuk penyelesaiannya. Gambaran secara umum dalam perancangan sistem dapat dilihat pada Gambar 1. Dengan menggunakan data pada halaman Al-Qur'an dan dibandingkan dengan halaman yang lainnya. Lalu data akan dicari nilai kemiripannnya dengan menggunakan *Jaccard Similarity*. Nilai yang dihasilkan akan digunakan untuk mencari akurasi dari sistem yang telah dibuat dengan keluaran berupa nilai akurasi. Berikut merupakan tahapan – tahapan yang tersusun dalam *flowchart* sebagai berikut.

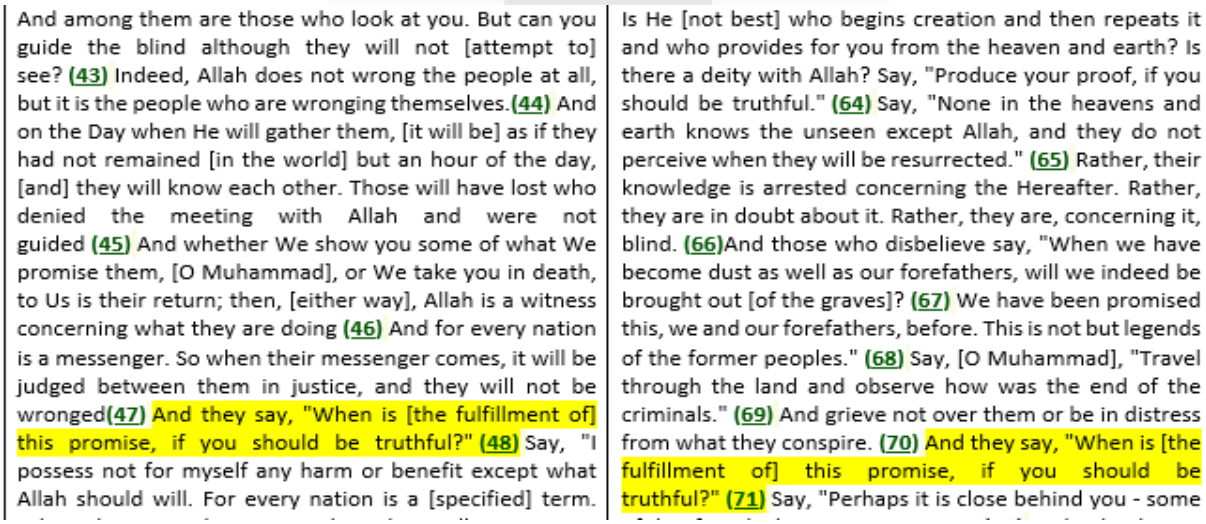


Gambar 2

Gambaran umum perancangan sistem

a. Dataset

Pada tugas akhir ini mengunakan data antar halaman – halaman Al – Qur'an yang sudah dipilih terlebih dahulu. *Dataset* pada proses pembentukan sistem menggunakan terjemahan Bahasa Inggris yang diambil pada situs *website* http://tanzil.net. Data halaman pada Al-Qur'an yang telah dikumpulkan, di gabung menjadi beberapa format file Microsoft Excel yang disebut dengan csv. Terdapat 3 jenis dataset yang digunaka pada Tugas Akhir ini. Jenis *dataset* tersebut antara lain, data mirip sekali, data kurang mirip, dan data yang tidak mirip.



And among them are those who look at you. But can you guide the blind although they will not [attempt to] see? (43) Indeed, Allah does not wrong the people at all, but it is the people who are wronging themselves.(44) And on the Day when He will gather them, [it will be] as if they had not remained [in the world] but an hour of the day, [and] they will know each other. Those will have lost who denied the meeting with Allah and were not guided (45) And whether We show you some of what We promise them, [O Muhammad], or We take you in death, to Us is their return; then, [either way], Allah is a witness concerning what they are doing (46) And for every nation is a messenger. So when their messenger comes, it will be judged between them in justice, and they will not be wronged(47) And they say, "When is [the fulfillment of] this promise, if you should be truthful?" (48) Say, "I possess not for myself any harm or benefit except what Allah should will. For every nation is a [specified] term.

Is He [not best] who begins creation and then repeats it and who provides for you from the heaven and earth? Is there a deity with Allah? Say, "Produce your proof, if you should be truthful." (64) Say, "None in the heavens and earth knows the unseen except Allah, and they do not perceive when they will be resurrected." (65) Rather, their knowledge is arrested concerning the Hereafter. Rather, they are in doubt about it. Rather, they are, concerning it, blind. (66)And those who disbelieve say, "When we have become dust as well as our forefathers, will we indeed be brought out [of the graves]? (67) We have been promised this, we and our forefathers, before. This is not but legends of the former peoples." (68) Say, [O Muhammad], "Travel through the land and observe how was the end of the criminals." (69) And grieve not over them or be in distress from what they conspire. (70) And they say, "When is [the fulfillment of] this promise, if you should be truthful?" (71) Say, "Perhaps it is close behind you - some

Gambar 3

Contoh *Dataset* yang di ambil oleh sistem

b. Preprocessing

Sebelum pengolahan data, *dataset* yang sudah diambil sebelumnya akan melewati tahapan ini. Ada beberapa tahapan pemersihan data pada *dataset* sebelum masuk ke proses utama. Pertama adalah proses *case folding*. *Case Folding* merupakan sebuah proses praproses yang mengubah semua *string* atau kalimat menjadi huruf kecil [13].

Sebelum proses *case folding*	Setelah proses *case folding*
And among them are those who look at you. **But** can you guide the blind although they will not [attempt to] see? **(43)**	**and** among them are those who look at you. **but** can you guide the blind although they will not [attempt to] see? **(43)**

Tabel 1

Perbedaan sebelum dan sesudah proses *case folding*

Selanjutnya adalah proses *Punctuation Removal*. *Punctuation* Removal adalah proses menghilangkan tanda baca pada setiap kata. Pada kalimat contoh didapatkan bahwa kata "asrama:" memiliki titik dua (:) yang menempel dengan kalimat tersebut sehingga perlu dihilangkan, sedangkan jika sebuah kata tidak terdapat tanda baca maka tidak akan ada perbuahannya. *List* tanda baca yang di hapus adalah (: , ; [ ] ! ? 1 2 3 4 5 6 7 8 9 0)

Sebelum proses *punctuation removal*	Setelah proses *punctuation removal*
And among them are those who look at you. But can you guide the blind although they will not [attempt to] see? (43)	and among them are those who look at you but can you guide the blind although they will not attempt to see

Tabel 2

Perbedaan sebelum dan sesudah proses *Punctuation Removal*

Lalu tahap akhir dari proses *Preprocessing* adalah *Whitespace Insensitivity*. Proses *Whitespacce* ini merupakan suatu proses pembersihan spasi antar kalimat. Jadi hasil akhir dari tahapan *Preprocessing* merupakan sekumpulan *string* yang ada pada satu penampung *array*.

Sebelum proses *Whitespace Insentivity*	Sesudah proses *Whitespace Insenttivity*
and among them are those who look at you.	'andamongthemarethosewholookatyou'

c. N- *Gram*

Teknik N-*gram* merupakan sebuah teknik untuk memotong *term* atau karakter. Dalam hal ini adalah huruf. Pemotongan huruf ini dilakukan pada kumpulan kata sebanyak N. Pengambilan karakter sebanyak N ini adalah dari teks yang dibaca dari awal hingga akhir dokumen. Sedangkan pada Tugas Akhir ini menggunakan beberapa variasi jumlah N pada sistem. N-*Gram* yang digunakan pada sistem antara lain adalah 2,4,6,8,10. Berikut merupakan contoh dari beberapa proses N-*Gram*. Set N- *Gram* pada aplikasi menggunakan N-*Gram* kurang dari 12. Hal ini diakukan dikarenakan keterbatasan panjang karakter dari kalimat yang ada pada ayat Al-Qur'an.

Sebelum proses N-Gram	Setelah proses N-Gram
andamongthemarethosewholookatyou	'andamongthemarethose', 'ndamongthemarethosew', 'damongthemarethosewh', 'amongthemarethosewho', 'mongthemarethosewhol', 'ongthemarethosewholo', 'ngthemarethosewholoo', 'gthemarethosewholook', 'themarethosewholooka', 'hemarethosewholookat', 'emarethosewholookaty', 'marethosewholookatyo', 'arethosewholookatyou'

Tabel 3

Contoh dari proses N-*Gram* dengan jumlah N = 20 pada sistem

d. Rolling Hashing

Pada tahap ini, sistem akan mengubah semua string yang sudah melewati proses *Preprocessing* dan *N-Gram* ke dalam bentuk nilai atau diubah ke dalam bentuk angka. Berikut merupakan contoh dari perhitungan *Rolling Hashing*.

Contoh diambil dari string "anda" dengan jumlah n = 3
Perhitungan pada iterasi pertama

Nilai Ascii a = 97, n = 110, d = 100 dan menggunakan bilangan prima 2
H ("and") = (97 x $2^{(3-1)}$) + (110 x $2^{(3-2)}$) + (100 x $2^{(3-3)}$)
= (97 x 8) + (110 x 2) + (97)
= 776 + 220 + 97
= 1093
Perhitungan pada iterasi kedua
H ("nda") = (1093 - (97 x $2^{(3-1)}$)) x $2^{(3-2)}$ + 97
= (1093 - 776) x (2) + (97)
= 317 x 2 + 97
= 731

e. Algoritma *Rabin Karp* (*Fingerprint*)

Pada Proses ini, Algoritma *Rabin Karp* akan mencari hasil dari perhitungan *rolling hash*. Algoritma ini akan membadingkan hasil nilai *rolling hash* dari satu karakter ke semua karakter yang ada pada salah satu dokumen.

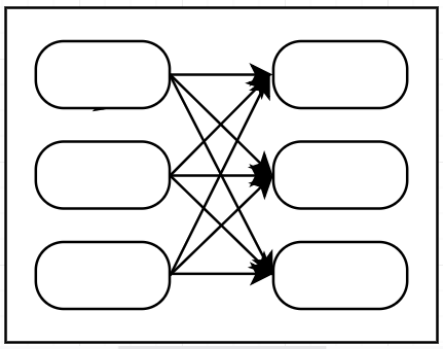
hash	172404 154262 159609 145290 134197 56087 148477 154621 134341 57683 166112 143617 159724 161192	172404 154262 159609 145290 134197 56087 148477 154621 134341 57683
fingerprint	172404 154262 159609 145290 134197 56087 148477 154621 134341 57683 166112 143617 159724 161192	172404 154262 159609 145290 134197 56087 148477 154621 134341 57683
	172404 154262 159609 145290 134197 56087 148477 154621 134341 57683	

Gambar 4

Contoh dari penggunaan *Rabin Karp* pada sistem

***f. Cross* data**

Pada proses ini sistem akan melakukan persilangan data antar ayat ayat pada halaman pertama dan halaman kedua. Pada proses persilangan ini, sistem akan sekalius mencari nilai similaritynya dengan menggunakan rumus yang disediakan oleh *Jaccard Similarity Coeffisien.* Dari proses persilangan yang dilakukan oleh sistem, maka hasil akan dimasukkan ke dalam sebuah *array*. Berikut contoh ilustrasi dalam kinerja yang dilakukan oleh sistem.



Gambar 5

Contoh Cross Data pada yang dilakukan sistem

g. Jaccard Similarity Coefficient

Dari proses sebelumnya, sistem akan menghitung hasil nilai yang dihasilkan oleh *fingerprinting*. Lalu sistem akan menghitung nilai tersebut dengan menggunakan *Jaccard Similarity Coeffisient*. Pada tahap ini nilai yang dikeluarkan oleh sistem merupakan jumlah nilai 0 dan 1. Dimana nilai 1 mengartikan sebuah ayat tersebut memiliki kemiripan tinggi dengan ayat yang lain. Begitu juga degan nilai 0 yang mengartikan sebuah ayat tersebut memiliki kemiripan rendah dengan ayat yang lain.

4. Evaluasi

4.1 Hasil Pengujian

Untuk melakukan uji pada sistem, digunakan 39 pasangan antar halaman *query* dengan pasangan antar halaman *corpus*. Di antara pasangan ayat tersebut dilakukan uji data pada ayat - ayat yang ada di dalam halaman *query* dan ayat – ayat pada halaman *corpus*. Pada pengujian ini, digunakan beberapa hasil data yang mengunakan jumlah N-*Gram*. Jumlah N yang dipakai antara lain 2,4,6,8,10. Jika nilai kemiripan yang dikeluarkan adalah 1 maka data dianggap sama atau mirip. Namun jika data menghasilkan nilai 0 maka data dianggap tidak sama atau tidak mirip. Dari Uji data tersebut maka dihasilkan nilai *precison, recall, f1- measure,* dan akurasinya.

Jumlah Label Kemiripan (Manual) dengan nilai sama dengan 1	Jumlah Label Kemiripan (Sistem) dengan nilai sama dengan 1			
39	19	19	19	19
Hasil Perhitungan				
Jenis *Meassurement*	N = 4	N = 6	N = 8	N = 10
Precision	32.76 %	32.76 %	32.76 %	32.76 %
Recall	48.72 %	48.72 %	48.72 %	48.72 %
Accuracy	9.93 %	9.93 %	9.93%	9.93 %
F-Measure	39.18 %	39.18 %	39.18 %	39.18 %

Tabel 4

Hasil dari perhitungan *precision, recall, f1-measure,* dan akurasi

4.2 Analisis Hasil Pengujian

Dari hasil uji di atas, dapat disimpulkan bahwa Algoritma *Rabin Karp* ini tidak terlalu berpengaruh dengan jumlah N yang diset kurang dari jumlah karakter pada dataset. Hal ini dibuktikan dengan nilai prosentase yang dihasilkan oleh sistem yang dimana sistem mendapatkan hasil yang sama pada jumlah N = 4, N = 6, N = 8, N = 10. Namun nilai yang dihasilkan sedikit berbeda ketika nilai N yang di set kurang dari 4.

5. Kesimpulan

Berdasarkan percobaan yang telah dilakukan sebelumnya, dapat disimpulkan bahwa Algoritma *Rabin karp* meghasilkan rata – rata nilai akurasi nya yang kurang maksimal. Tercatat dari ke-4 nilai *precsion* yang dilakukan, prosentase nilai maksimumnya hanya sampai pada angka 77,42 %. Hal ini dikarenakan jumlah data manual nya yang kurang lengkap pada kasus ini.

Kedepannya, dapat dilakukan pengembangan Tugas Akhir ini dengan menambahkan data manual antar ayat agar sistem tidak hanya mengeluarkan nilai 0 dan 1 saja. Selain itu untuk menigkatkan nilai akurasinya, dapat ditambahkan metode ataupun algoritma yang dapat mendeteksi kemiripan kata satu dengan kata yang lainnya.

Daftar Pustaka

[1] M. H. T. G. M. T. J. Martin Potthast, "Overview of the 5th International Competition on Plagiarism Detection," *CLEF 2013 Evaluation Labs and Workshop–Working Notes Papers*, p. 1–31, 2013.

[2] Firdaus, "Algoritma Rabin Karp," 2008.

[3] H. B. Firdaus, "DETEKSI PLAGIAT DOKUMEN MENGGUNAKAN," p. 2, 2008.

[4] X. Yao, "Feature-Driven Question Answering with Natural Language," p. 1–353, 2014.

[5] B. S. a. E. Handrie Noprisson, "Implementasi Algoritma RabinKarp untuk menentukan Keterkaitan antar Publikasi Penelitian Dosen Tahun 2013," *Teknologi Informasi Volume 9 Nomor 2*, p. 1–15, 2013.

[6] J. Cosma, "Plagiarisme," 2008.

[7] "Wikipedia Plagiarism," [Online]. Available: www.wikipedia.com. [Accessed 14 August 2018].

[8] K. H. Reynald Karisma Wibowo, "Penerapan Algoritma Winnowing Untuk Deteksi Plagiarisme Tugas Akhir Mahasiswa Universitas Dian Nuswantoro," pp. 4-5.

[9] "Wikipedia Jaccard Similarity Coeffisient," [Online]. Available: www.wikipedia.com. [Accessed 2018 August 14].

[10] J. S. E. N. a. S. W. Suphakit Niwattanakul, "the International MultiConference of Engineers and Computer Scientists 2013 Vol I," in *IMECS 2013*, Hong Kong, March 13 - 15, 2013.

[11] J. Euzenat, "Semantic Precision and Recall for Ontology Alignment Evaluation," in *In Proceedings of IJCAI-2007*, 2007.

[12] S. H. P. S. a. t. Marghescu D., "Performance Evaluation," in *IGI Global*, 2008.

[13] S. K. Vairaprakash Gurusamy, "Preprocessing Techniques for Text Mining," pp. 4-6, 2014.

[14] "Python Documentation," Python Software Foundation, [Online]. Available: https://docs.python.org/3/tutorial/datastructures.html#list-comprehensions. [Accessed 14 August 2018].

[15] bangdavid, "bangdavid.blogspot.com," [Online]. Available: https://bangdavid.blogspot.com/2017/10/pengertian-precision-recall-accuracy.html. [Accessed 2018 August 14].