Jurnal Teknologi dan Informasi (JATI)
Volume 12 Nomor 2 Edisi September 2022
P-ISSN 2088-2270, E-ISSN 2655-6839
DOI 10.34010/jati.v12i2

Aplikasi Deteksi Kemiripan Kata Menggunakan Algoritma Rabin-Karp

S L B Ginting[1], Y R Ginting[2], Sutono[3], W A Sirait[4]
Program Studi Sistem Komputer, Fakultas Teknik dan Ilmu Komputer, Universitas Komputer Indonesia[134]
Jalan Dipatiukur No. 112-116, Bandung, 40132, Indonesia[134]
Program Studi Teknik Mesin, Fakultas Teknik, Universitas Riau[2]
Kampus Bina Widya KM 12,5, Simpang Baru, Pekanbaru 28293, Indonesia[2]
selvia.lorena@email.unikom.ac.id*[1], yogierinaldy@gmail.com[2], sutono@email.unikom.ac.id[3], waldy.sirait@gmail.com[4]

diterima: 18 April 2022
direvisi: 23 Oktober 2022
dipublikasi: 29 Oktober 2022

Abstrak

Meningkatnya tindakan plagiarisme di universitas dan perguruan tinggi merupakan masalah yang cukup serius dan perlu ditangani. Pada zaman sekarang ini, mendapatkan suatu informasi memang sangatlah mudah sehingga memberikan kesempatan kepada kalangan tertentu untuk mendapatkan keuntungan pribadi. Salah satunya adalah dengan mengambil atau menggunakan karya orang lain tanpa izin seolah-olah itu adalah hasil karyanya sendiri dan ini disebut juga plagiarisme. Berdasarkan keadaan ini peneliti bertujuan untuk merancang sebuah aplikasi pendeteksi kemiripan kata pada dokumen. Metode yang diterapkan adalah algoritma Rabin-Karp dengan menggunakan pola string matching. Pencocokan pada kata sudah ditentukan sebesar 5 k-gram dan dikonversi ke nilai hashing dan nilai hashing yang serupa akan membentuk kesamaan kata pada kedua dokumen yang akan dilakukan percobaan. Dari hasil percobaan yang telah dilakukan menggunakan 10 data dokumen abstrak pada skripsi menghasilkan tingkat akurasi sebesar 95.08% dan waktu untuk memproses dokumen yang diuji rata-rata 11,8 detik. Sistem pendeteksi kesamaan kata ini dapat membantu mengidentifikasi kesamaan antara dua dokumen yang dibandingkan.
Kata kunci: Plagiarisme; Aplikasi Deteksi Plagiat; Algoritma Rabin-Karp; Kemiripan

Abstract

The increase in deliberate acts of plagiarism in universities and colleges is a serious problem and needs to be addressed. In this day and age, getting information is easy so that it provides opportunities for certain circles to gain personal gain. One of them is to take or use other people's work without permission as if it was your own work and this is called plagiarism. Based on the circumstances, this researcher aims to design a word detection application in documents. The method applied is the Rabin-Karp algorithm using string pattern matching. Matches on words have been determined at 5 k-grams and similar hashing values will form similarities in the two documents that will be tested. Based on the findings of the tests that have been performed using 10 abstract document data in the thesis, it produces an accuracy rate of 95.08% and the time to process the tested documents is an average of 11.8 seconds. This word detection system can help identify similarities between the two documents being compared.
Keywords: *Plagiarism; Plagiarism Detection Application; Rabin-Karp Algorithm; Similarity*

1. Pendahuluan

Teknologi yang semakin berkembang pesat membawa banyak dampak baik itu positif ataupun negatif. [1-4]. Dari sekian banyak pengaruh negatif dari teknologi, salah satunya adalah masyarakat lebih rentan terhadap penipuan dan dapat berujung pada kejahatan [1,5]. Penipuan yang kerap terjadi disebabkan kemajuan teknologi ialah penjiplkan terhadap karya kepunyaan orang lain dan mengklaim sebagai milik sendiri [1,5]. Untuk mencegah tindakan

penjiplakan karya ini, maka dibutuhkan suatu aplikasi dalam melakukan pengecekan kesamaan terhadap dua buah dokumen. Dalam penelitian ini, Aplikasi dibangun dengan menerapkan algoritma Rabin-Karp yang dapat mendeteksi kesamaan pada dua buah dokumen. Cara kerja algoritma ini adalah menghitung kata yang sama dari dokumen dibandingkan dengan dokumen yang lain untuk menentukan apakah dokumen yang diperiksa ini merupakan tiruan dari penulis lain atau bukan [4].

Plagiarisme yang marak terjadi di Indonesia adalah di sejumlah perguruan tinggi, baik negeri maupun swasta [5]. Melihat hal ini mantan menteri pendidikan nasional Mohammad Nuh mengatakan, "Meningkatnya plagiarisme menunjukkan kelemahan dalam pendidikan karakter, budaya dan moralitas manusia di dunia akademis." Ini adalah contoh buruk pada dunia pendidikan tidak terkecuali di Indonesia sendiri begitu banyak tindakan meniru karya seseorang baik tanpa disadari maupun disadari [5, 6]. Mendeteksi kesamaan dalam dokumen adalah metode yang efisien untuk memperbaiki mutu pada dunia pendidikan dan mengurangi kesamaan dokumen. Namun pengenalan secara manual sulit dilakukan, sehingga dibutuhkan sebuah aplikasi yang bisa mengenali kesamaan kata pada dokumen. Percobaan yang sudah dilakukan oleh Surahman yang mana mengusulkan pembuatan pengembangan algoritma Rabin-Karp dengan berbasis web dengan tujuan untuk memudahkan pengguna melakukan pengecekan karya internet [1]. Penelitian yang lain, Astuti B., metode yang digunakan adalah algoritma Rabin-Karp untuk menemukan kesamaan dokumen atau menganalisis kesamaan dokumen seperti *substring matching*, kemiripan dalam dokumen dan metode *cosine similarity* untuk menghasilkan nilai kemiripan yang serupa pada aplikasi pemeriksa plagiarisme [7]. Siswanto mengklaim, hasil perhitungan akurasi dengan menggunakan *confusion matrix* makalah penelitian dari 20 makalah penelitian perbandingan diperoleh kemiripan 90% [8]. Filcha dan Hayaty melakukan percobaan pertama memiliki nilai kemiripan tertinggi 5 k-gram , dengan nilai kemiripan 98,24%. Sedangkan percobaan kedua plagiarisme nilai kemiripan 2 k-gram dengan nilai kemiripan 98,98% [9]. Di samping itu Algoritma Rabin-Karp juga digunakan oleh Jauhari A., yang menyatakan bahwa Algoritma Rabin-Karp lebih efisien diterapkan pada kasus pengecekan kemiripan dokumen dibandingkan dengan algoritma *Jaro-Winkler*. Ini dikarenakan algoritma Rabin-Karp menampilkan persentase kemiripan kata yang lebih besar dibandingkan dengan algoritma *Jaro-Winkler* di beberapa pengujian yang dilakukan. Rabin-Karp mempunyai kemiripan sebanyak 51% sedangkan *Jaro-Winkler* mempunyai rata-rata sebanyak 35% [10].

Berdasarkan penelitian-penelitian tersebut, algoritma Rabin-Karp dapat dianggap sebagai metode yang efektif dan akurat dalam proses pendeteksian kemiripan dokumen, karena dibandingkan dengan algoritma yang lain algoritma Rabin-Karp lebih tinggi akurasinya pada tingkat pencocokan kata sehingga persentase kemiripan lebih tinggi. Oleh karena itu, metode Rabin-Karp cocok digunakan dalam studi kasus ini. Lebih lanjut, pada penelitian-penelitian sebelumnya hanya menerapkan sistem *upload file* sedangkan sistem yang penulis terapkan adalah menggunakan dua cara yaitu *upload file* dan memasukkan teks langsung sehingga mempermudah pengguna hanya dengan melakukan *copy-paste*.

2. Kajian Pustaka

2.1 Plagiarisme

Plagiarisme atau pelanggaran hak cipta adalah perbuatan yang disengaja untuk memperoleh nilai dan juga karya ilmiah kepentingan sendiri tanpa mencantumkan asal muasal karya tersebut. Berdasarkan penelitian-penelitian sebelumnya, hal-hal yang dapat dianggap sebagai tindakan plagiarisme, diantaranya [11]:

1. Salin dan tempel. Salin semua kata tanpa melakukan modifikasi/perubahan.
2. Pelanggaran hak cipta, tindakan pendelegasian menyembunyikan bagian yang direplikasi,

dibedakan menjadi empat strategi, yaitu *shake* and *paste*, pencurian sastra luas, pemalsuan kontraktif, dan plagiat *mosaic*.

3. Penyamaran teknis. Untuk menutupi konten yang dicuri dari pendeteksi otomatis, menggunakan kekurangan sistem analisis teks dasar, seperti memperbarui alfabet pada simbol bahasa asing.
4. Parafrase yang tidak tepat, parafrase yang disengaja dari ide-ide orang lain melalui pilihan istilah, dan gaya plagiarisme sambil menyembunyikan asal aslinya.
5. Plagiarisme yang diterjemahkan. Mengonversi dari satu bahasa ke bahasa asing lain.
6. Ide plagiarisme, memanfaatkan pikiran orang lain tanpa mencantumkan sumbernya.

2.2 Algoritma Rabin-Karp

Perhitungan Rabin-Karp adalah salah satu dari banyak pendekatan pencocokan string yang dirancang dan ditemukan oleh Michael O. Rabin dan Richard M. Karp. Standar perhitungan ini adalah mengamati desain teks dengan membandingkan nilai hash dari setiap teks. Untuk teks dengan panjang n dan contoh pencarian dengan panjang m, biasanya kompleksitas waktu terbaik adalah O(n), sedangkan kompleksitas waktu terburuk adalah O(mn). Dasar Rabin-Karp adalah memisahkan nilai *hashing* dari string input dengan *substring* teks. Jika sama, perbandingan dengan karakter dilakukan lagi dan jika tidak sama, menggeser sub string ke kanan. Bagian terpenting dari kinerja algoritma ini adalah perhitungan yang efektif pada nilai *hashing substring* ketika diterapkan [12].

2.3 Tokenizing

Pengertian dari tokenizing adalah untuk membuat pecahan kalimat membentuk sebuah kata atau *token*. *Tokenizing* digunakan dengan cara membagi kata tersebut dan menetapkan struktur sintaksis dari setiap kata. Proses ini akan menghasilkan kata yang membentuk *string*/teks atau *tokenizing* [13]. Berikut ini adalah contoh proses *tokenizing*.

Input:
Kalimat: Tokenizing adalah fase di mana *string* input dipotong berdasarkan pembatas yang diberikan. Karakter selain itu dianggap sebagai pembatas dan dihapus oleh proses untuk menampung kata-kata yang membentuk teks. Proses ini menghasilkan kata yang membentuk apa yang sering disebut sebagai *string*/teks atau *token/term*.

Output:
{token} {okeni} {keniz} {enizi} {nizin} {izing} {zinga} {ingad} {ngada} {gadal} {adala} {dalah} {alahf} {lahfa} {ahfas} {hfase} {fased} {asedi} {sedim} {edima} {diman} {imana} {manas} {anast} {nastr} {astri} {strin} {tring} {ringi} {ingin} {nginp} {ginpu} {input} {nputd} {putdi} {utdip} {tdipo} {dipot} {ipoto} {poton} {otong} {tongb} {ongbe} {ngber} {gberd} {berda} {erdas} {rdasa} {dasar} {asark} {sarka} {arkan} {rkanp} {kanpe} {anpem} {npemb} {pemba} {embat} {mbata} {batas} {atasy} {tasya} {asyan} {syang} {yangd} {angdi} {ngdib} {gdibe} {diber} {iberi} {berik} {erika} {rikan} {ikank} {kanka} {ankar} {nkara} {karak} {arakt} {rakte} {akter} {kters} {terse} {ersel} {rsela} {selai} {elain} {laini} {ainit} {initu} {nitud} {itudi} {input} {nputd} {putdi} {utdip} {tdipo} {dipot} {ipoto} {poton} {otong} {tongb} {ongbe} {ngber} {gberd} {berda} {erdas} {rdasa} {dasar} {asark} {sarka} {arkan} {rkanp} {kanpe} {anpem} {npemb} {pemba} {embat} {mbata} {batas} {atasy} {tasya} {asyan} {syang} {yangd} {angdi} {ngdib} {gdibe} {diber} {iberi} {berik} {erika} {rikan} {ikank} {kanka} {ankar} {nkara} {karak} {arakt} {rakte} {akter} {kters} {terse} {ersel} {rsela} {selai} {elain} {laini} {ainit} {initu} {nitud} {itudi} {tudia} {udian} {diang} {iangg} {tudia} {udian}

Jurnal Teknologi dan Informasi (JATI)
Volume 12 Nomor 2 Edisi September 2022
P-ISSN 2088-2270, E-ISSN 2655-6839
DOI 10.34010/jati.v12i2

{diang}token} {okeni} {keniz} {enizi} {nizin} {izing} {zinga} {ingad} {ngada} {gadal} {adala} {dalah} {alahf} {lahfa} {ahfas} {hfase} {fased} {asedi} {sedim} {edima} {diman} {imana} {manas} {anast} {nastr} {astri} {strin} {tring} {ringi} {ingin} {nginp} {ginpu} {input} {nputd} {putdi} {utdip} {tdipo} {dipot} {ipoto} {poton} {otong} {tongb} {ongbe} {ngber} {gberd} {berda} {erdas} {rdasa} {dasar} {asark} {sarka} {arkan} {rkanp} {kanpe} {anpem} {npemb} {pemba} {embat} {mbata} {batas} {atasy} {tasya} {asyan} {syang} {yangd} {angdi} {ngdib} {gdibe} {diber} {iberi} {berik}{iangg} {angga} {nggap} {ggaps} {gapse} {apseb} {pseba} {sebag} {ebaga} {bagai} {agaip} {gaipe} {aipem} {ipemb} {pemba} {embat} {mbata} {batas} {atasd} {tasda} {asdan} {sdand} {dandi} {andih} {ndiha} {dihap} {ihapu} {hapus} {apusa} {pusat} {usata} {satau} {ataud} {taudi} {audih} {udiha} {dihap} {ihapu} {hapus} {apuso} {pusol} {usole} {soleh} {olehp} {lehpr} {ehpro} {hpros} {prose} {roses} {osesu} {sesun} {esunt} {suntu} {untuk} {ntukm} {tukme} {ukmen} {kmena} {menam} {enamp} {nampu} {ampun} {mpung} {pungk} {ungka} {ngkat} {gkata} {katak} {ataka} {takat} {akata} {katay} {ataya} {tayan} {ayang} {yangm} {angme} {ngmem} {gmemb} {membe} {emben} {mbent} {bentu} {entuk} {ntukt} {tukte} {uktek} {kteks} prose}{roses} {osesi} {sesin} {esini} {sinim} {inime} {nimen} {imeng} {mengh} {engha} {nghas} {ghasi} {hasil} {asilk} {silka} {ilkan} {lkank} {kanka} {ankat} {nkata} {katay} {ataya} {tayan} {ayang} {yangm} {angme} {ngmem} {gmemb{katay} {ataya} {tayan} {ayang} {yangm} {angme} {ngmem} {gmemb} {membe} {emben} {mbent} {bentu} {entuk} {ntukt} {tukte} {uktek} {kteks} prose}{roses} {osesi} {sesin} {esini} {sinim} {inime} {nimen} {imeng} {mengh} {engha} {nghas} {ghasi} {hasil} {asilk} {silka} {ilkan} {lkank} {kanka} {ankat} {nkata} {katay } {membe} {emben} {mbent} {bentu} {entuk} {ntuka} {tukap} {ukapa} {kapay} {apaya} {payan} {ayang} {yangs} {angse} {ngser} {gseri} {serin} {ering} {ringd} {ingdi} {ngdis} {gdise} {diseb} {isebu} {sebut} {ebuts} {butse} {utseb} {tseba} {sebag} {ebaga} {bagai} {agais} {gaist} {aistr} {istri} {strin} {tring} {ringt} {ingte} {ngtek} {gteks} {teksa} {eksat} {ksata} {satau} {ataut} {tauto} {autok} {utoke} {token} {okent} {kente} {enter} {nterm}

2.4 Filtering

Filtering adalah proses untuk menghapus kata-kata yang tidak berpengaruh pada dokumen. Kata yang tidak penting akan disebut sebagai *stoplist*. *Stoplist* (*stopword)* merupakan kata-kata yang tidak deskriptif yang dapat dihapus dalam proses kumpulan kata. Contohnya adalah “what”, “and”, “at”, “of” [14].

2.5 K-gram

K-gram merupakan kumpulan *terms* dengan panjang K dan yang umum dipakai oleh *terms* adalah kata. K-gram adalah sebuah metode yang digunakan untuk menghasilkan kata atau karakter. Dalam metode k-gram ini dilakukan pengambilan potongan-potongan karakter huruf sejumlah k dari sebuah kata yang secara kontinuitas dibaca dari teks sumber hingga akhir dari dokumen [10]. Pada tabel 1 ditunjukkan bahwa perbedaan antara proses *preprocessing* dan proses k-gram (5) terletak pada suku kata dipecah menjadi 5 huruf per kalimat, tujuannya adalah untuk mempermudah proses *similarity*.

Tabel 1. Contoh 5 K-gram

Kalimat:	**Plagiat tindakan tidak terpuji**
Preprocessing:	plagiattindakantidakterpuji

k-gram(5):	{plagi} {lagia} {agiat} {giatt} {iatti} {attin} {ttind} {tinda} {indak} {ndaka} {dakan} {akant} {kanti} {antid} {ntida} {tidak} {idakt} {dakte} {akter} {kterp} {terpu} {erpuj} {rpuji}

2.6 Hashing

Hash adalah proses pengkonversian *string* yang membentuk nilai numerik yang disebut nilai hash. *String* dikonversi ke nilai tunggal menggunakan panjang yang tetap, berguna untuk indikator *string*. Dalam sistem ini, proses *hashing* menggunakan tabel *hash* ASCII [1].

$$Ip = p1 * d(m-1) + p2 * d(m-2) + \cdots + p(m-1) * d1 + p(m-1) * d0 \quad (1)$$

Dimana:
I = nilai *hash*
p = nilai karakter *ASCII* (desimal)
m = jumlah karakter (indeks karakter)
d = radix (nilai radix harus bilangan prima)

Contohnya, apabila *substring* yang akan dicari "*sistem*", karakter (k) yang ditetapkan adalah 5 maka *substring sistem* dibagi dua yaitu "*siste*" & *istem*. Sementara itu, basis (b) yang dipakai adalah 3, rumus perhitungan nilai *hashing* adalah:

$H(siste) = ascii(m) * 3\ (4) + ascii(a) * 3\ (3) + ascii(c) * 3\ (2) + ascii(h) * 3\ (1) + ascii(i) * 3\ (0)$

$H(siste) = 109 * 81 + 97 * 27 + 99 * 9 + 104 * 3 + 105 * 1 = 12756$

Nilai *hash* pada *substring* "*siste*" bernilai 12756. Untuk *substring* berikutnya, misal: "*achio*" k = 5 dan b = 3.

$H(istem) = (H(siste) - ascii(m) * 3\ (4)\ ) * 3 + ascii\ (o)$

$H(istem) = (12756 - 109 * 81) * 3 + 111 = 11892$

Nilai *hash* pada *substring* "*istem*" bernilai 11892.

Input:
Kalimat: *Tokenizing* adalah fase di mana string input dipotong berdasarkan pembatas yang diberikan. Karakter selain itu dianggap sebagai pembatas dan dihapus oleh proses untuk menampung kata-kata yang membentuk teks. Proses ini menghasilkan kata yang membentuk apa yang sering disebut sebagai *string*/teks atau *token/term* [13].

Output:
{185} {20} {35} {135} {480} {200} {365} {430} {400} {90} {65} {315} {510} {360} {155} {240} {355} {495} {430} {195} {440} {25} {195} {80} {355} {105} {130} {425} {425} {265} {20} {200} {275} {330} {310} {355} {500} {430} {290} {370} {270} {125} {40} {170} {15} {115} {335} {470} {255} {10} {505} {210} {70} {80} {125} {245} {515} {50} {95} {330} {410} {50} {135} {75} {70} {465} {395} {435} {485} {35} {525}

{160} {275} {350} {265} {215} {180} {420} {310} {325} {540} {430} {40} {320} {480} {15} {160} {250} {45} {205} {0} {530} {145} {260} {335} {125} {415} {310} {185} {210} {555} {330} {485} {535} {0} {20} {330} {530} {540} {515} {50} {95} {330} {305} {120} {465} {405} {440} {60} {95} {125} {335} {365} {445} {310} {280} {15} {110} {290} {510} {360} {125} {335} {365} {515} {35} {295} {505} {220} {50} {530} {430} {45} {170} {470} {240} {40} {390} {460} {420} {230} {15} {280} {180} {70} {370} {120} {85} {485} {540} {25} {285} {550} {505} {270} {300} {55} {270} {20} {120} {115} {415} {295} {110} {260} {195} {470} {90} {500} {455} {395} {455} {265} {45} {170} {410} {280} {335} {530} {525} {25} {315} {135} {25} {390} {270} {0} {550} {145} {450} {60} {265} {225} {550} {55} {270} {20} {120} {115} {415} {295} {110} {260} {195} {470} {90} {500} {360} {325} {75} {360} {255} {30} {120} {145} {395} {145} {245} {215} {180} {400} {445} {480} {190} {380} {165} {295} {375} {275} {520} {320} {485} {535} {0} {35} {395} {135} {85} {130} {425} {480} {560} {175} {275} {130} {300} {305} {15} {190} {455} {415} {255} {185} {75} {270} {400} {110} {110} {290} {510} {360} {125} {335} {365} {515} {35} {295} {505} {220} {50} {530} {430} {45} {170} {470} {240} {40} {390} {460} {420} {230} {15} {280} {180} {70} {370} {120} {85} {485} {540} {25} {285} {550} {505} {270} {300} {55} {270} {20}{185} {20} {35} {135} {480} {200} {365} {430} {400} {90} {65} {315} {510} {360} {155} {240} {355} {495} {430} {195} {440} {25} {195} {80} {355} {105} {130} {425} {425} {265} {20} {200} {275} {330} {310} {355} {500} {430} {290} {370} {270} {125} {40} {170} {15} {115} {335} {470} {255} {10} {505} {210} {70} {80} {125} {245} {515} {50} {95} {330} {410} {50} {135} {75} {70} {465} {395} {435} {485} {35} {525} {160} {275} {350} {265} {215} {180} {420} {310} {325} {540} {430} {40} {320}

2.7 Perhitungan Similarity

Perhitungan kesamaan (*similarity*) dan jarak antara dua entitas informasi merupakan persyaratan inti untuk seluruh temuan informasi, termasuk pengambilan informasi dan data yang penting. Ditingkatkan dalam sebuah bentuk aplikasi, salah satunya merupakan sistem pendeteksi kemiripan [16,17]. Langkah-langkah kesamaan yang tepat tidak hanya mengembangkan kapasitas pemilihan bahan, namun juga memperpendek *the price and time* pemrosesan. Saat menghitung nilai kesamaan menggunakan pendekatan k-gram. Berikut adalah perhitungan yang digunakan.

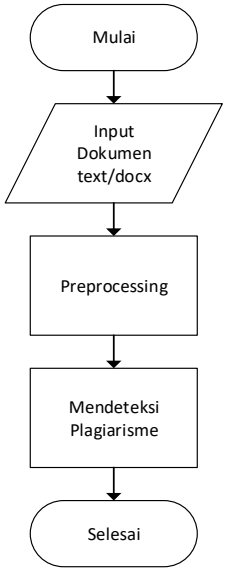
$$S = (IH / THA + THB) \times 100\% \quad (2)$$

Dimana:
S = *Similarity* (kesamaan)
IH = Jumlah pola yang sama
THA = Jumlah hashing dokumen uji
THB = Jumlah hashing dokumen penguji

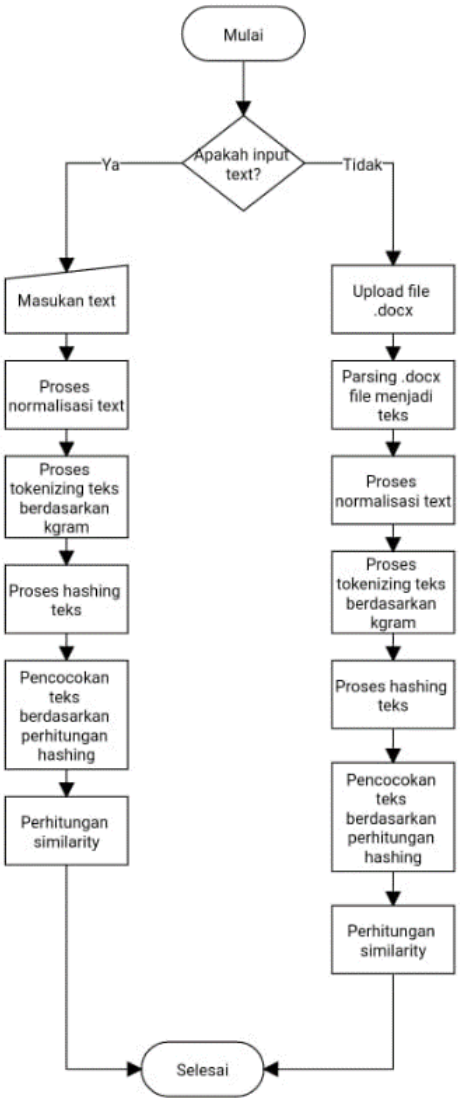
2.8 Flowchart Aplikasi Pendeteksi Kemiripan Kata dan Flowchart Proses Algoritma Rabin-Karp

Flowchart adalah grafik yang menunjukkan aliran program sistem atau prosedur dengan cara yang logis. Flowchart terutama digunakan untuk mendukung komunikasi dan dokumentasi [11]. Berikut gambar 1 yang mengemukakan sistem Pendeteksi Kemiripan Kata dan gambar 2 mengenai algoritma Rabin-Karp.

Jurnal Teknologi dan Informasi (JATI)
Volume 12 Nomor 2 Edisi September 2022
P-ISSN 2088-2270, E-ISSN 2655-6839
DOI 10.34010/jati.v12i2



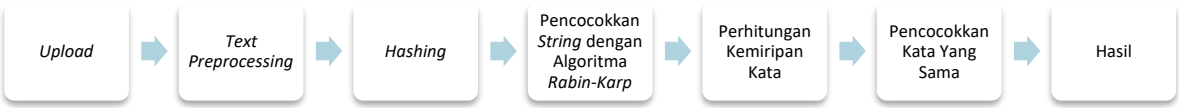
Gambar 1. Flowchart Sistem Plagiarisme



Gambar 2. Flowchart Proses Rabin-Karp

3. Metode Penelitian

Metode pengolahan sistem pendeteksi kesamaan dokumen dengan menerapkan algoritma Rabin-Karp, yang merupakan deskripsi kebutuhan perangkat lunak sebelum pembuatan kode dimulai. Perancangan meliputi diagram skema dari sistem pendeteksi kemiripan kata. Langkah-langkah proses utama dari sistem pengenalan kesamaan kata adalah:



Gambar 3. Model Proses Rabin-Karp

Gambar 3 di atas adalah model proses Rabin-Karp yang merupakan tahapan umum dari algoritma Rabin-Karp. Tahap pertama adalah melakukan upload dokumen yang akan di proses lalu kemudian di tahap kedua melakukan *text preprocessing* yang bertujuan untuk memecah dokumen menjadi potongan *token/term*/kata untuk memudahkan proses komputasi. Kemudian dokumen ini akan diproses lebih lanjut. Tahap ketiga melakukan

Jurnal Teknologi dan Informasi (JATI)
Volume 12 Nomor 2 Edisi September 2022
P-ISSN 2088-2270, E-ISSN 2655-6839
DOI 10.34010/jati.v12i2

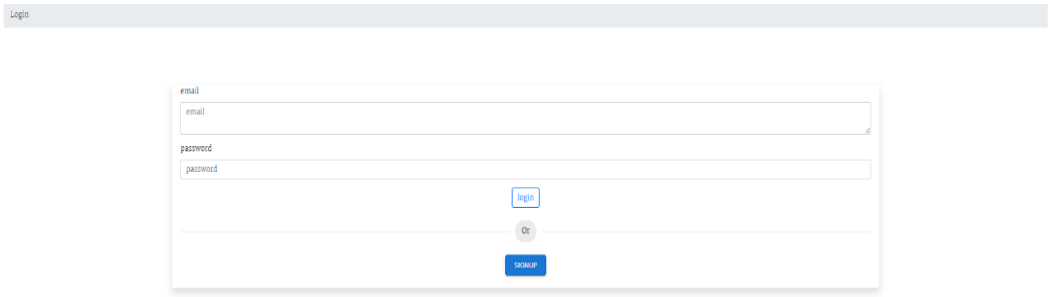
proses *hashing* yaitu proses pengkonversian string yang akan membentuk nilai numerik. Tahap berikutnya yaitu tahap keempat adalah tahap pencocokan *string* dengan algoritma Rabin-Karp dilanjutkan dengan tahap perhitungan kemiripan kata. Tahap keenam adalah pencocokan kata yang sama yang akan menghasilkan persentase kemiripan dua dokumen yang diproses. Selanjutnya ditampilkan hasil perhitungan yang merupakan persentase kemiripan kata.

4. Hasil dan Pembahasan

Sistem yang dikembangkan bertujuan untuk membantu pengguna mengidentifikasi kesamaan antar dokumen. Jika materinya sama akan dianggap melakukan plagiarisme atau peniruan. Aplikasi ini ialah aplikasi berbasis web yang dapat digunakan siapa saja. Maka dari itu, tampilan aplikasi dibuat sesederhana mungkin untuk kemudahan penggunaannya.

1. Tampilan Halaman Utama

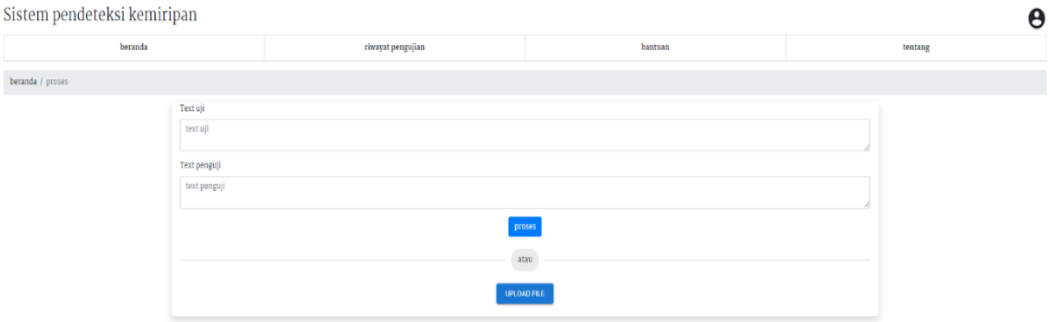
Pada saat melakukan pengujian *similarity*, diperlukan *login* terlebih dahulu sebelum melakukan akses ke *website*. Setelah melakukan *login*, *website* akan menampilkan halaman utama. Di mana terdapat menu beranda, riwayat pengujian, bantuan dan tentang. Pada menu profil menampilkan *account* dan *change password*. Menu *account* menunjukkan data *user* seperti *username*, *email*, nama, NIK. Menu *edit account*, dapat mengubah *password*-nya dengan cara memasukkan *password* lama kemudian memasukkan *password* baru.



Gambar 4. Tampilan Halaman Utama

2. Tampilan Input Teks Langsung

Pada halaman input dua file dapat dilihat pada gambar 4, pengguna dapat memasukkan teks langsung dengan cara *copy & paste* teks uji dan teks penguji serta upload dokumen berekstensi.docx. Ini tujuannya adalah agar memudahkan pengguna dan selain itu ukuran file juga tidak terlalu besar atau kata yang diproses hanya sedikit, maka dari itu diterapkan sistem melalui proses input teks langsung.

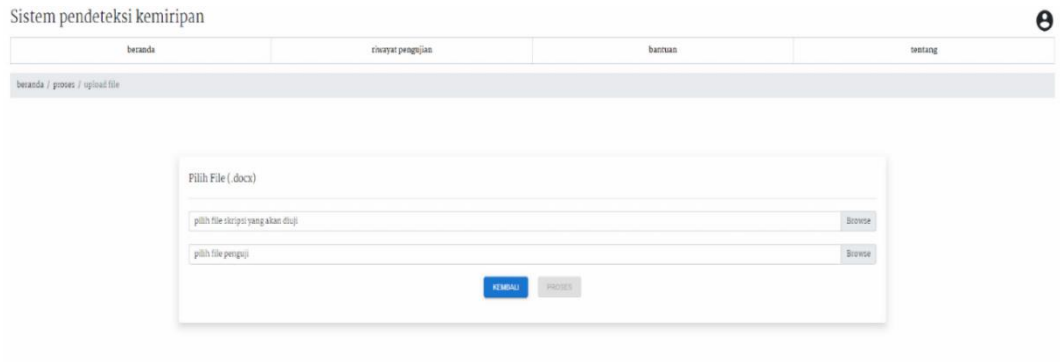


Gambar 5. Tampilan Pada Masukan Teks

Jurnal Teknologi dan Informasi (JATI)
Volume 12 Nomor 2 Edisi September 2022
P-ISSN 2088-2270, E-ISSN 2655-6839
DOI 10.34010/jati.v12i2

3. Tampilan Halaman Upload File

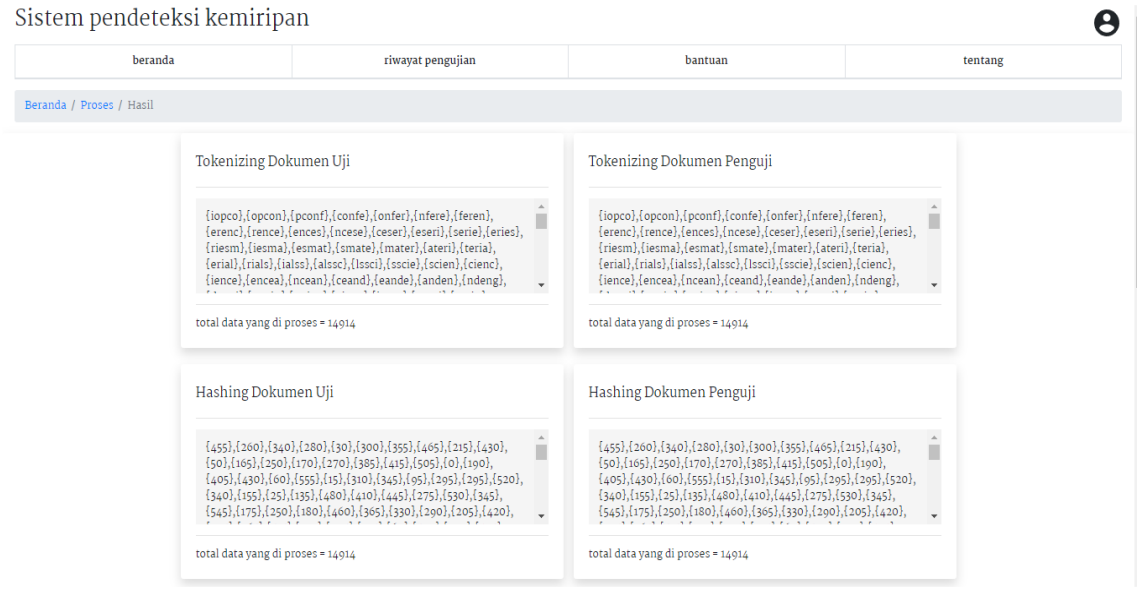
Dapat dilihat pada gambar 5 ada menu Pilih file uji dan file penguji yang akan mengakses *directory*/penyimpanan di dalam komputer *user*. Lalu kemudian akan diproses kemiripan kata pada dokumen dan setelah input file *user* dapat pilih proses sehingga sistem akan memproses dokumen dan akan menghasilkan persentase kemiripan dokumen.



Gambar 6. Tampilan Halaman Pada Masukan Teks atau Upload Dokumen

4. Tampilan Hasil Proses Dokumen Pada

Pada halaman ini menampilkan proses hasil dari *Tokenizing* dan *Hashing* pada dua dokumen yang dibandingkan.

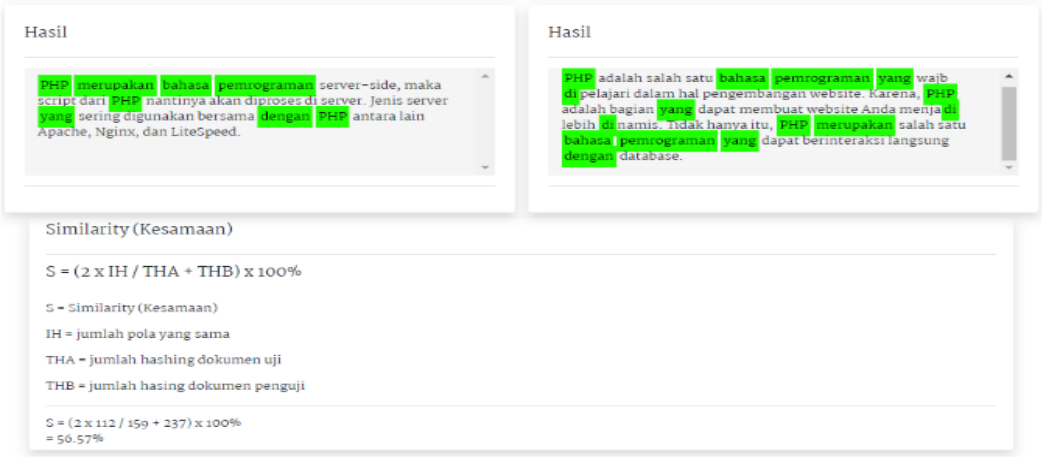


Gambar 7. Tampilan Halaman Proses Dokumen

5. Tampilan Hasil Proses Dokumen

Menampilkan hasil dari proses Rabin-Karp, setiap kata yang memiliki kesamaan yang diuji sistem akan menampilkan tanda hijau, di mana tanda tersebut merupakan kata yang sama dari antara kedua dokumen yang diuji.

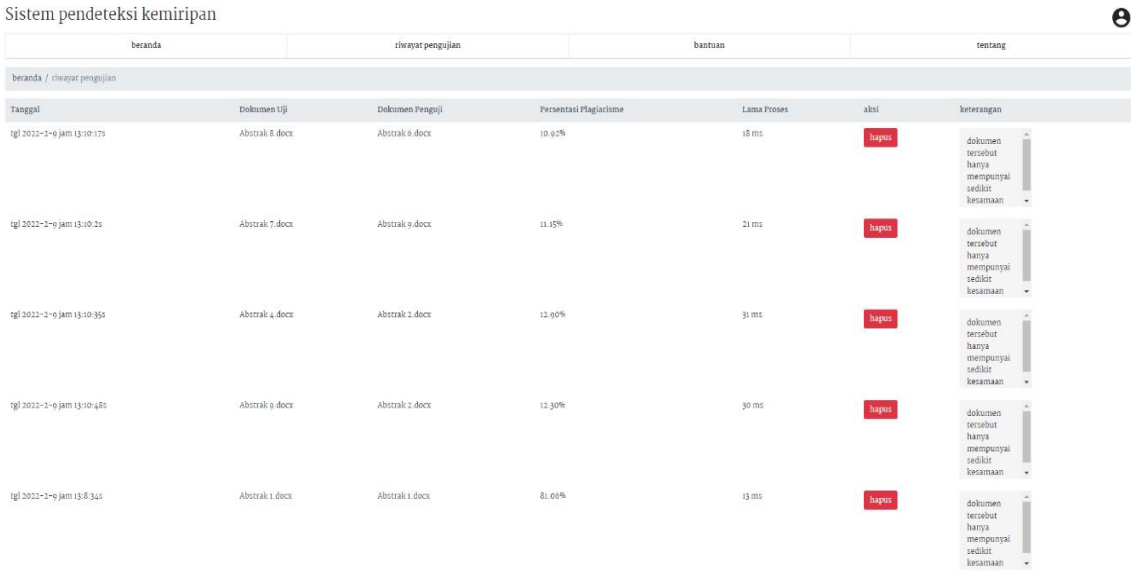
Jurnal Teknologi dan Informasi (JATI)
Volume 12 Nomor 2 Edisi September 2022
P-ISSN 2088-2270, E-ISSN 2655-6839
DOI 10.34010/jati.v12i2



Gambar 8. Tampilan Halaman Hasil Similarity

6. Tampilan Halaman Riwayat Pengujian

Pada tampilan ini, pengguna dapat melihat Riwayat Pengujian dokumen yang disimpan oleh pengguna setelah melakukan proses pendeteksi dokumen dan akan menampilkan tanggal yang akan sebagai id, nama dokumen uji dan penguji, persentase Kemiripan Kata, lama proses, aksi dan keterangan.



Gambar 9. Tampilan Riwayat Pengujian Sistem

Pada Tabel 2 menunjukkan perbedaan antara teks uji dan teks penguji setelah melalui hasil *text processing*.

Tabel 2. Hasil *Text Preprocessing*

Teks Uji:	Plagiat merupakan tindakan tidak terpuji
Teks Penguji:	Tindakan plagiarisme adalah tidak baik untuk di contoh

Jurnal Teknologi dan Informasi (JATI)
Volume 12 Nomor 2 Edisi September 2022
P-ISSN 2088-2270, E-ISSN 2655-6839
DOI 10.34010/jati.v12i2

Tabel 3**.** Tahap Proses Algoritma Rabin-Karp

Tahap-Tahap Algoritma Rabin-Karp	Teks Uji	Teks Penguji
Tokenizing	{plagi} {lagia} {agiat} {giatm} {iatme} {atmer}{tmeru} {merup} {erupa} {rupak} {upaka} {pakan} {akant} {kanti} {antin} {ntind} {tinda} {indak} {ndaka} {dakan} {akant} {kanti} {antid} {ntida} {tidak} {idakt} {dakte} {akter} {kterp} {terpu} {erpuj} {rpuji}	{tinda} {indak} {ndaka} {dakan} {akanp} {kanpl} {anpla} {nplag} {plagi} {lagia} {agiar} {giari} {iaris} {arism} {risme} {ismea} {smead} {meada} {eadal} {adala} {dalah} {alaht} {lahti} {ahtid} {htida} {tidak} {idakb} {dakba} {akbai} {kbaik} {baiku} {aikun} {ikunt} {kuntu} {untuk} {ntukd} {tukdi} {ukdic} {kdico} {dicon} {icont} {conto} {ontoh}
Hashing	{20} {285} {210} {55} {485} {260} {250} {200} {170} {65}{440} {250} {415} {275} {0} {150} {390} {270} {130} {280} {415} {275} {515} {545} {365} {520} {360} {540} {415} {130} {250} {190}	{390} {270} {130} {280} {395} {115} {230} {200} {20} {285} {200} {230} {250} {15} {145} {10} {130} {395} {95} {65} {315} {15} {165} {210} {560} {365} {430} {400} {280} {375} {345} {345} {0} {410} {460} {375} {280} {115} {95} { 120} {145} {95} {480}

Pada tabel 3 menampilkan hasil dari proses *tokenizing* dan hasil dari proses setiap kata yang diujikan yang kemjudian akan diperoleh persentase kemiripan melalui proses yang terdapat di bawah ini yang merupakan perhitungan *similarity* yang dilakukan:

$$S = (IH / THA + THB) \times 100\% \quad (3)$$

Di mana:
S = Similarity (Kesamaan)
IH = jumlah pola yang sama
THA = jumlah hashing dokumen uji
THB = jumlah hashing dokumen penguji
S = (7 / 32 + 43) x 100%
= 9.33%

Tabel 4. Hasil Perbandingan K-gram pada Dokumen

File Dokumen k-gram sama				File Dokumen k-gram Berbeda		
No	Nama Dokumen Uji	Nama Dokumen Penguji	Hasil Kemirpan	Nama Dokumen Uji	Nama Dokumen Penguji	Hasil Kemirpan
1	Abstrak 1.docx	Abstrak 1.docx	90.06%	Abstrak 1.docx	Abstrak 10.docx	6.60%
2	Abstrak 2.docx	Abstrak 2.docx	100%	Abstrak 7.docx	Abstrak 9.docx	11.15%

3	Abstrak 3.docx	Abstrak 3.docx	100%	Abstrak 8.docx	Abstrak 6.docx	10.92%
4	Abstrak 4.docx	Abstrak 4.docx	100%	Abstrak 4.docx	Abstrak 2.docx	12.90%
5	Abstrak 5.docx	Abstrak 5.docx	88.04%	Abstrak 9.docx	Abstrak 2.docx	12.30%
		Rata-Rata Persentase	95,06 %		Rata-Rata Persentase	11%

Berdasarkan hasil pengujian yang disajikan pada Tabel 4, output pengujian telah dihasilkan dan dapat dicermati bahwa seluruh pengujian sudah terima. Oleh karena itu, sistem identifikasi pencocokan kata memenuhi semua spesifikasi persyaratan fungsional yang ditentukan sebelumnya. Dalam hasil proses uji validasi, terdapat 10 dokumen yang diuji. Terlihat sistem pendeteksi kemiripan kata memperoleh persentase berdasarkan 10 dokumen uji. Pengujian menggunakan arsip k-gram yang sama dengan setiap dokumen yang diproses memberikan tingkat kesamaan rata-rata 95,06% karena isi dan jumlah data yang diproses sama. Sedangkan pengujian yang dilakukan berdasarkan k-gram yang berbeda persentase yang diperoleh adalah rata yang 11% isi dan data tidak sama, karena isi dari dokumen yang diuji benar-benar berbeda sehingga mempengaruhi pencocokan pada kalimat. Dilihat dari sisi waktu proses sistem, jika dokumen dengan ukuran lebih besar yang di proses maka akan membutuhkan waktu lebih lama untuk diselesaikan.

5. Kesimpulan

Kesimpulan dari penelitian ini adalah dokumen dengan menggunakan metode algoritma Rabin-Karp berbasis werb berhasil diterapkan. Aplikasi dapat mengidentifikasi kata yang sama dalam dua dokumen dan akan ditampilkan hasil berupa persentase *similarity* atau kesamaan kata. Aplikasi berhasil memproses dokumen melalui input teks langsung dan juga proses upload dokumen. Hasil perhitungan mempunyai nilai akurasi sebesar 95,06% yang diperoleh dari 10 dokumen asbtrak pada skripsi dan memerlukan waktu proses eksekusi rata-rata 11,8 detik. Semakin besar ukuran file yang diproses oleh aplikasi maka semakin lama waktu proses eksekusi yang dibutuhkan. Namun, penelitian ini dapat lebih ditingkatkan di masa depan. Penelitian mendatang dapat berupa peningkatan dalam hal dapat membedakan banyak dokumen artinya lebih dari dua dokumen. Kemudian penting untuk menampilkan teknik pemecahan kalimat yang dapat mengidentifikasi judul/kependekan dan konstruksi arsip yang kompleks seperti angka, tabel, dan lain-lain. Selain itu sistem yang dibangun di masa depan (penelitian selanjutnya) diharapkan dapat memproses berbagai jenis dokumen seperti (*.pdf), (*.odt) dan (*jpg).

Daftar Pustaka

[1] Surahman, A.M., "Perancangan Sistem Penentuan Similarity Kode Program Pada Bahasa C Dan Pascal Dengan Menggunakan Algoritma Rabin-Karp" JUSTIN (Jurnal Sistem dan Teknologi Informasi), 2015, Tersedia: https://jurnal.untan.ac.id/index.php/justin/article/view/1097 [Diakses: 7 Maret 2022]

[2] Kristiyono, J., Budaya Internet: "Perkembangan Teknologi Informasi Dan Komunikasi Dalam Mendukung Penggunaan Media Di Masyarakat" Scriptura, 2015. 5(1): p. 23-30. Tersedia: https://ojs.petra.ac.id/ojsnew/index.php/iko/article/view/19386. [Diakses: 10 Maret 2022]

[3] Ameliola, Syifa, and Hanggara Dwiyudha Nugraha. "Perkembangan media informasi dan teknologi terhadap anak dalam era globalisasi." *Prosiding In International Conference On Indonesian Studies" Ethnicity And Globalization*. 2013. Tersedia: https://www.academia.edu/7996998/PERKEMBANGAN_MEDIA_INFORMASI_DAN_TEKNOLOGI_TERHADAP_ANAK_DALAM_ERA_GLOBALISASI.[Diakses: 5-6 Maret 2022]

[4] Lorena Br Ginting, Selvia, Wendi Zarman, and Astrid Darmawan. "Teknik Data Mining Untuk Memprediksi Masa Studi Mahasiswa Menggunakan Algoritma K-Nearest Neighborhood." *KOMPUTIKA-Jurnal Sistem Komputer UNIKOM* 3.2 (2015). Tersedia: https://repository.unikom.ac.id/30342/. [Diakses: 13 Agustus 2022]

[5] Pakan, S., "Pencocokan String Untuk Mencari Kemiripan Pada Dokumen Teks Dengan Algoritma Rabin-Karp" Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Kristen Duta Wacana, 2013Tersedia: https://katalog.ukdw.ac.id/3394/. [Diakses: 13 Maret 2022]

[6] Astuti, B., "Identifikasi Perilaku Plagiat Pada Mahasiswa Fakultas Ilmu Pendidikan" Universitas Negeri Yogyakarta, Artikel Penelitian, Yogyakarta: Fakultas Ilmu Pendidikan UNY, 2012.

[7] Djafar, F.B., "Penerapan Algoritma Smith-Waterman Dalam Sistem Pendeteksi Kesamaan Dokumen. Skripsi" 2013. 1(531408030). Tersedia: https://repository.ung.ac.id/skripsi/show/531408030/penerapan-algoritma-smith-waterman-dalam-sistem-pendeteksi-kesamaan-dokumen.html. [Diakses: 7 Maret 2022]

[8] Siswanto, Eric, and Yo Ceng Giap. "A Implementasi algoritma rabin-karp dan cosine similarity untuk pendeteksi plagiarisme pada dokumen." *ALGOR* 1.2 (2020): 16-22. Tersedia: https://jurnal.ubd.ac.id/index.php/algor/article/view/299. [Diakses: 21 Maret 2022]

[9] Filcha, A. and M. Hayaty, "Implementasi Algoritma Rabin-Karp untuk Pendeteksi Plagiarisme pada Dokumen Tugas Mahasiswa" JUITA: Jurnal Informatika, 2019. **7**(1): p.25-32.Tersedia: http://jurnalnasional.ump.ac.id/index.php/JUITA/article/view/4063. [Diakses: 7 Maret 2022]

[10] Jauhari, A., "Pendeteksian Plagiarisme Dengan Menggunakan Algoritma Rabin Karp" Universitas Trunojoyo,2015 Tersedia:http://digilib.mercubuana.ac.id/manager/t!@file_artikel_abstrak/Isi_Artikel_140319122078.pdf. [Diakses: 7 Maret 2022]

[11] Leonardo, B. and S. Hansun, "Text documents plagiarism detection using Rabin-Karp and Jaro-Winkler distance algorithms" Indonesian Journal of Electrical Engineering and Computer Science", 2017. 5(2): p. 462-471. Tersedia: https://www.researchgate.net/profile/Seng-Hansun/publication/316681173_Text_Documents_Plagiarism_Detection_using_Rabin-Karp_and_Jaro-Winkler_Distance_Algorithms/links/59ad1e16458515d09cdfdc4f/Text-Documents-Plagiarism-Detection-using-Rabin-Karp-and-Jaro-Winkler-Distance-Algorithms.pdf. [Diakses: 7 Maret 2022]

[12] Jody, J., A.T. Wibowo, and A. Arifianto, "Analisis Dan Implementasi Algoritma Winnowing Dengan Synonym Recognition Pada Deteksi Plagiarisme Untuk Dokumen Teks Berbahasa Indonesia" e Proceedings of Engineering, 2015. 2(3). Tersedia: https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/1361. [Diakses: 7 Maret 2022]

[13] Ginting, Selvia Lorena Br, and Hayi Akbar. "Pembangunan Perangkat Lunak Menggunakan Algoritma Ant Colony Optimization Untuk Optimasi Penjadwalan Kuliah (Studi Kasus Penjadwalan Kuliah Jurusan Teknik Komputer Unikom)." *Jurnal Manajemen Informatika (JAMIKA)* 3.1 (2013). Tersedia: https://ojs.unikom.ac.id/index.php/jamika/article/view/665. [Diakses: 7 Agustus 2022]

[14] Putra, Doddi Aria, Herry Sujaini, and Helen Sasty Pratiwi. "Implementasi Algoritma Rabin-Karp untuk Membantu Pendeteksian Plagiat pada Karya Ilmiah." *Jurnal Sistem dan Teknologi Informasi (JUSTIN)* 1.1 (2015): 1-9. Tersedia: https://core.ac.uk/download/pdf/296441603.pdf. [Diakses: 14 Agustus 2022]