Implementasi Algoritma Rabin Karp Untuk Mendeteksi Persamaan Kata Berbasis Website

Fanny Ruzaini Rahmanna Ari
Universitas PGRI Madiun
email: 727fani01@gmail.com

Abstract: *Plagiarism detection is an important aspect in maintaining academic integrity in educational institutions. This study aims to implement the Rabin-Karp algorithm which can detect synonyms and identify cases of plagiarism in a web-based environment. The Rabin-Karp algorithm, known for its efficiency in string matching, was modified to compare word patterns and identify similarities between existing documents. By using this algorithm in a web-based system, educators can effectively detect and deal with cases of plagiarism, increase fair evaluation and maintain academic integrity. This study explores the practical implementation of the Rabin-Karp algorithm and its significance in overcoming plagiarism by working with extreme programming methods that do not require much detailed data, only using UML and class diagrams for storage using JSON with a PHP backend interface. The results of this study are the implementation of the Rabin-Karp algorithm to detect web-based word similarities. In testing this system using black box testing to determine the function of each existing menu.*

Keywords*: Algoritma Rabin Karp, Selling words, Rolling Hashing, Black Box*

Abstrak: Deteksi plagiasi merupakan aspek penting dalam menjaga integritas akademik di lembaga pendidikan. Penelitian ini bertujuan untuk implementasi algoritma *Rabin-Karp* yang dapat mendeteksi persamaan kata dan mengidentifikasi kasus plagiasi dalam lingkungan berbasis web. Algoritma Rabin-Karp, yang dikenal karena efisiensinya dalam pencocokan *string*, dimodifikasi untuk membandingkan pola kata dan mengidentifikasi kemiripan antara dokumen-dokumen yang ada. Dengan menggunakan algoritma ini dalam sistem berbasis web, para pendidik dapat secara efektif mendeteksi dan mengatasi kasus plagiasi, meningkatkan evaluasi yang adil dan menjaga integritas akademik. Penelitian ini mengeksplorasi implementasi praktis algoritma *Rabin-Karp* dan signifikansinya dalam mengatasi plagiasi dengan metode pengerjaan dengan cara extreme programming tidak memperlukan banyak rincian data hanya saja menggunakan UML dan diagram class untuk penyimpanan menggunakan JSON dengan penghubung backend nya PHP. Hasil dari penelitian ini implementasi algoritma *Rabin-Karp* untuk mendeteksi persamaan kata berbasis web. Pada pengujiannya sistem ini menggunakan *black box testing* untuk mengetahui fungsi dari setiap menu yang ada.

Kata kunci*: Algoritma Rabin Karp, Persamaan Kata, Rolling Hashing, Black Box*

Pendahuluan

Pesatnya kemajuan teknologi telah membawa banyak manfaat positif. Namun, dampak negatifnya tidak dapat diabaikan, salah satunya adalah menyalin. Praktik ini terjadi tidak hanya di kalangan mahasiswa sarjana, tetapi juga di kalangan peneliti berpengalaman yang harus dihargai tingkat pengetahuannya. Metode menyalin dan menempel ini tampaknya umum dan dapat dimengerti. Untuk mengatasi masalah tersebut, dikembangkan aplikasi pendeteksi *plagiarisme* menggunakan *document string matching* (Putra & Sularno, 2019).

Dalam dunia pendidikan, perkembangan teknologi yang pesat menawarkan banyak keuntungan, terutama bagi pendidikan tinggi. Sebagai agen perubahan, siswa harus menyesuaikan teknologi dan menggunakannya untuk pengembangan lebih lanjut mereka sendiri. Kementerian Pendidikan dan Kebudayaan membuat program Kampus Merdeka yang bertujuan untuk keleluasaan bagi mahasiswa untuk belajar di luar studinya. Ini harus

memungkinkan siswa untuk lebih memenuhi tuntutan dunia saat ini, yang ditandai dengan perubahan sosial, budaya dan teknologi yang signifikan. Program Kampus Mandiri adalah praktik yang menawarkan kesempatan kepada mahasiswa untuk lebih meningkatkan keterampilannya sesuai dengan bakat dan minatnya (Rozaq et al., 2022:30).

Sebelum mendefinisikan suatu karya siswa yang mengandung *plagiarism* atau tidak mengandung *plagiarisme*, sangat penting untuk mengetahui apa pengertian *plagiarisme* atau *plagiarisme*. *Plagiarisme* adalah adopsi esai, opini, dan lain-lain orang lain dan penyajiannya sebagai karangan sendiri, misalnya dengan menerbitkan tulisan orang lain dengan biaya sendiri (Khaled & Sabeeh, 2021). Orang yang menjiplak disebut penjiplak atau penjiplak.

Berdasarkan argumentasi tersebut, kesamaan atau persamaan isi tugas antar mahasiswa merupakan *plagiarisme*. Saat ini sering terjadi di perguruan tinggi bahwa dokumen studi mahasiswa diperiksa *plagiarism* secara langsung dengan manual dan dibandingkan dengan dokumen studi lainnya. Dengan cara ini, kontrol *plagiarisme* memakan waktu lama. Salah satu metode yang dapat digunakan adalah metode pencocokan *string*, salah satunya menggunakan algoritma Rabin-karp. Menurut Charas dan Lecroq dalam (Siswanto & Giap, 2020), penggunaan fungsi hashing memberikan sebuah metode yang simpel untuk menghindari perbandingan jumlah karakter yang bersifat kuadratik dalam banyak kasus atau situasi. Dari pada memeriksa setiap posisi dalam teks saat pencocokan pola terjadi, lebih baik dan efisien untuk memeriksa hanya jika teks yang sedang diproses memiliki kemiripan dengan pola yang dicari (Sari & Utomo, 2020:44). Algoritma ini juga dapat digunakan untuk mencocokkan beberapa pola *string* sekaligus dalam rangka menemukan kesamaan atau kecocokan data. Algoritma *Rabin-Karp* bergerak dari kiri ke kanan, fungsi algoritma *Rabin-Karp* menghasilkan efisiensi waktu yang baik ketika mencari *string* dengan pola lebih dari satu (Filcha & Hayaty, 2019:26).

Pada penelitian yang dilakukan oleh Supriyadi & Asih, (2020), Dengan berjudul "Aplikasi Deteksi Kemiripan Kata Menggunakan Algoritma *Rabin-Karp"* dengan bertujuan penelitian ini dengan penelitian ini adalah dokumen dengan menggunakan metode algoritma *Rabin-Karp* berbasis *website* 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

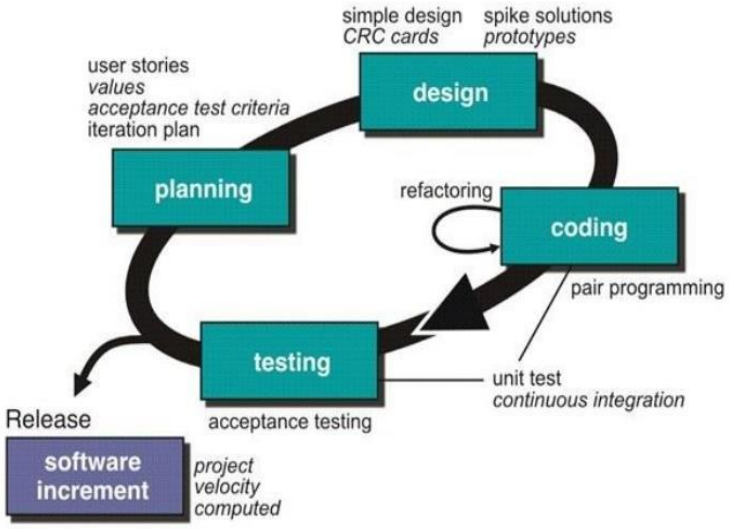
Penelitian Selanjutnya yang dilakukan oleh Sari & Utomo, (2020), yang berjudul "Implementasi Algoritma *Rabin-Karp* Pada Pencarian Quotes Tokoh Terkenal" penelitian yang dilakukan oleh peneliti penulis tentang pencarian quotes tokoh terkenal, maka dari bab-bab sebelumnya dapat diambil kesimpulan yang merupakan hasil akhir dari penulisan yaitu Proses pencarian quotes tokoh terkenal dapat dilakukan dengan menginputkan beberapa kata yang ingin dicari. Kemudian aplikasi akan menampilkan beberapa quotes yang sesuai dengan kata yang telah diinputkan. Penerapan algoritma *Rabin-Karp* ini dilakukan pemeriksaan terhadap setiap posisi dari teks ketika terjadi pencocokan pola, akan lebih efisien melakukan pemeriksaan hanya jika teks yang sedang diproses memiliki kemiripan.

Berdasarkan penelitian terdahalu terdapat kemiripan dan perbedaan yang akan dilakukan oleh peneliti. Kemiripan yang ada pada penelitian terdahulu adalah pada alogritma yang digunakan yaitu algoritma *Rabin-Karp* dan berbasis *website*.

e-ISSN: 2685-5615
p-ISSN: 2715-5315

Metode

Pengembangan sistem dalam penelitian ini menggunakan metode Extreme Programming (XP). Metode Extreme Programming (XP) merupakan metode pengembangan perangkat lunak yang tergolong dalam pendekatan metode Agile (Utama, 2023).Metode pengembangan sistem Extreme Programming merupakan acuan dasar dalam perancangan dan pembangunan sistem. Berikut adalah tahapan pengembangan sistem: Gambar metode extreme programming dapat dilihat pada Gambar 1.



Gambar 1. Metode *Extreme Programing*

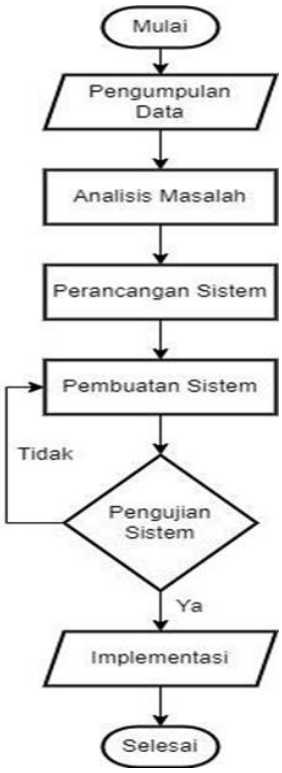
Tahap planning merupakan pembuatan user story yang diperoleh melalui pengumpulan data. User story yang dibuat akan dijadikan gambaran dasar dari pengembangan sistem yang akan dilakukan. Tahap design dilakukan perancangan terhadap alur kerja sistem dan pemodelan basis data berdasarkan user story yang ada. Pada tahap coding, pemodelan yang telah dibuat diimplementasikan dalam baris-baris kode pemrograman. Pada pembuatan sistem dilakukan oleh 2 orang oleh seorang programmer dan tester dengan tujuan jika ada koreksi tahap ini akan dilakukan secara berulang-ulang refactoring agar menghasilkan sistem yang baik. Tahap testing merupakan tahap pengujian akhir sistem untuk mengecek masalah-masalah yang kemungkinan ada pada setiap modul dalam sistem. Apabila pengujian belum sesuai dengan permintaan, akan dilakukan perbaikan pada bagian yang dikoreksi.

Rancangan penelitian yang digunakan dapat dilihat pada Gambar 2. Pada tahap pengumpulan data dilakukan pengambilan data dengan metode wawancara, observasi langsung dan studi pustaka. Pelaksanaan wawancara dan observasi dilakukan secara langsung.

Tahapan selanjtunya analisa data yang telah terkumpul kemudian dibuat user story untuk dijadikan dasar dalam pembangunan sistem yang ramah pada pengguna. Pada tahap ini dilakukan perancangan alur kerja sistem, pemodelan alur kerja sistem menggunakan Unified Modelling Language (UML), perhitungan Rabin karp serta perancangan User Interface (antarmuka) sistem. Pada tahap ini dilakukan penulisan kode program sesuai rancangan yang telah dilakukan.

Selanjutnya, modul-modul yang telah jadi dilakukan pengujian apakah tedapat ketidak sesuaian dalam pengerjaan. Pada tahap ini sistem diuji untuk memastikan seluruh alur kerja sistem bekerja sesuai dengan fungsinya. Pada tahap ini sistem yang sudah dilakukan

pengujian dapat diimplementasikan dengan menambah hosting supaya mendapat akses terhadap internet dan dapat diakses secara terbuka oleh dalam mengidentifikasi pattern dan *text*



Gambar 2. Flowchart Rancangan Penelitian

Hasil

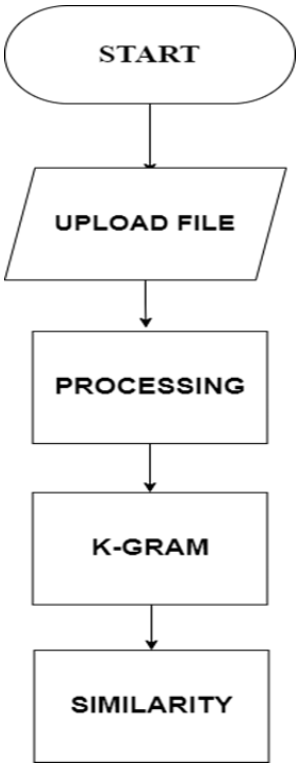
Analisis Sistem

Secara garis besar tentang perancangan system deteksi kesamaan dokumen dengan menggunakan algoritma RabinKarb. Pada umumnya algoritma pendeteksi kesamaan dokumen atau *plagiarisme* memiliki tahapan yang sama dalam pemrosesan dokumen teks. Tahapan-tahapan tersebut terdiri dari input, proses, output. Berawal dari tahap input dokumen teks yangakan diuji kemudian dokumen diproses melalui tahapan prepocessing (Case Folding, Tokenizing, *Filtering*) dan hashing kemudian hitungan *similarity* dokumen. Setelah diproses maka selanjutnya aplikasi akan menghasilkan informasi dokumen.Tujuan dari analisis sistem ini adalah untuk memberikan pemahaman yang komprehensif tentang implementasi algoritma *Rabin-Karp* dalam konteks sistem berbasis *website.*

Dengan menentukan nilai k-gram dan basis bilangan primanya. Hasil dari hashing asli dan hashing uji kemudian dibandingkan untuk dicari hashing yang sama. Jika hashing yang sama ditemukan, maka dari hasil hashing yang sama tersebut dihitung tingkat persentase kesamaannya (similarity). Adapun langkah-langkah algoritma Rabin karp yang dilakukan dalam perancangan sistem ini adalah menghilangkan tanda baca, membagi teks kedalam bentuk k-gram, dimana nilai k merupakan nilai parameter yang dipilih oelh pengguna, menghitung nilai hash dari setiap k-gram, memilih nilai hash yang sama, menghitung nilai tingkat kesamaan dua dokumen.

e-ISSN: 2685-5615
p-ISSN: 2715-5315

K-gram merupakan kumpulan terms dengan panjang K dan yang umum dipakai oleh termsadalah kata (Widiatry & Sari, 2019). Tokenizing adalah proses untuk memecah kalimat menjadi kata-kata atau token (Herwinsyah & Witanti, 2022). Sedangkan hashsing adalah proses pengkonversian *string* yang membentuk nilai numerik yang disebut nilai hash (Rahim et al., 2022). Untuk lebih jelasnya dapat dilihat pada Gambar 3.



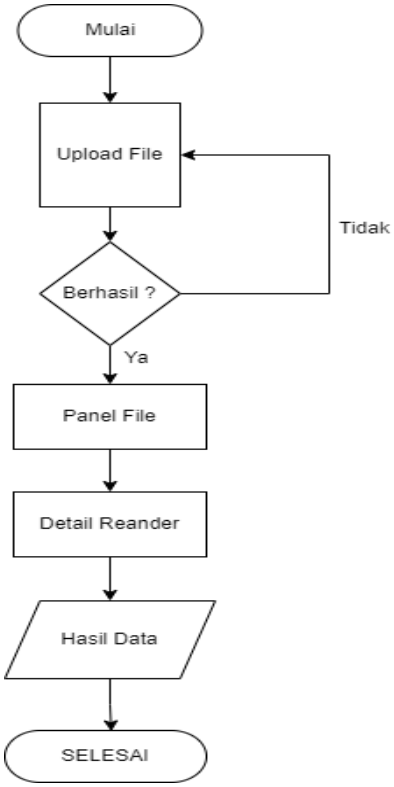
Gambar 3 Flowchart Metode Rabin Karp

Perancangan Sistem

Pada perancangan sistem peneliti menggunakan perancangan sistem *Unified Modeling Language* atau UML. UML merupakan penyatuan dari pemodelan Booch Method, Object Modeling Technique (OMT) dan Object Oriented Software Engineering (OOSE). Dimulai pada tahun 1994 tiga pakar metodologi pemodelan berorientasi objek yaitu Booch, Rumbaugh dan Jacobson mempelopori proses penyatuan metodologi (Anardani, 2019). Dalam UML terdapat beberapa diagram sebagai cara mendefinisian alur kerja dari suatu sistem atau bagian tertentu dalam sebuah sistem, beberapa diagram dalam UML yaitu *use case diagram, sequence diagram, activity diagram,* dan *class diagram*. Selain itu peneliti menggunakan *flowchart,* dan perhitungan *Rolling Hash.*

Flowchart merupakan papan cerita yang berisi antarmuka, lokasi menu, integrasi antar menu dalam bentuk frame dan sebagainya (Rabiman et al., 2020) seperti pada Gambar 4.

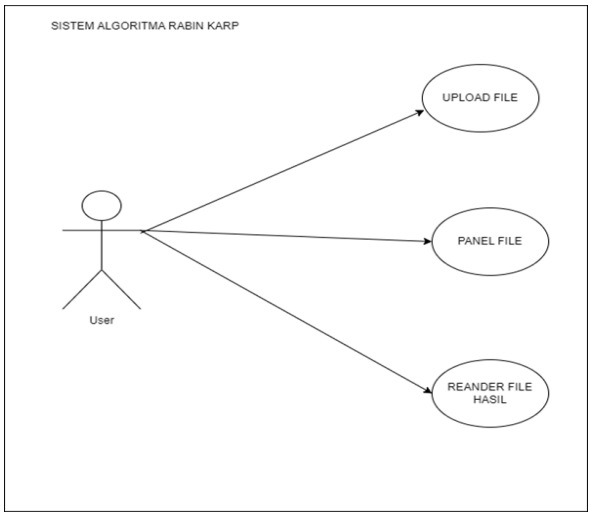
e-ISSN: 2685-5615
p-ISSN: 2715-5315



Gambar 4 *Flowchart* Alur Kerja Sistem

Pemodelan data pertama-tama dilakukan dengan membagi sekelompok data dalam tiga bagian yaitu processing, Rolling hash dan *similarity*. processing yang telah dipisahkan kemudian dilakukan setelah melakukan processing membagi bagian dengan ada k-gram yang akan diarahkan kedalam *Rolling hash* dan hasil akhir yang akan keluar berupa *similarity.*

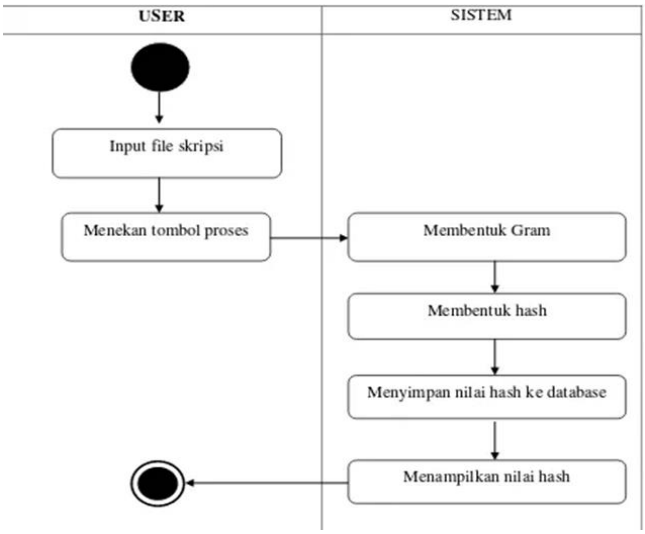
Use case diagram adalah serangkaian tindakan yang dilakukan olehsistem, pengguna diwakili oleh aktor, atau sistem lain yang berinteraksi dengan sistem yang dimodelkan (Zulfa et al., 2020). Dapat dilihat pada Gambar 5.



Gambar 5 *Use Case Diagram*

e-ISSN: 2685-5615
p-ISSN: 2715-5315

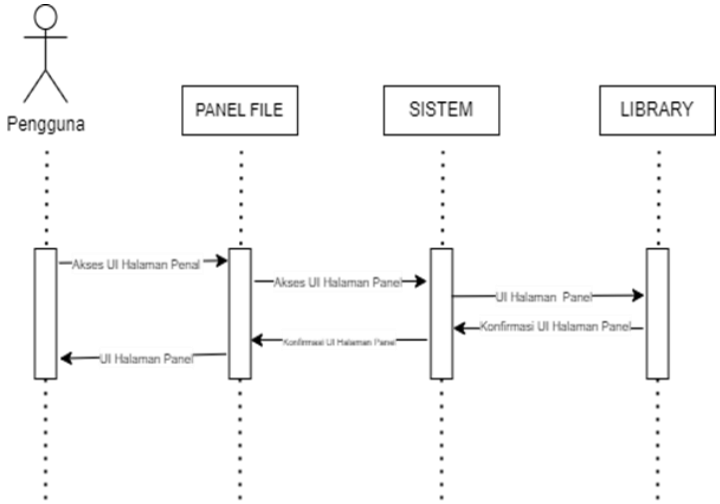
Activity Diagram adalah representasi yang disederhanakan dari diagram aktivitas (Fan et al., 2021). Dapat dilihat Gambar 6.



Gambar 6 *Activity Diagram*

Activity diagram diatas menjelaskan alur kerja sistem *string match*. Dalam diagram tersebut pengguna dapat mengakses sistem dengan 3 menu navigasi yaitu halaman uploads, halaman panel file dan halaman reander. Halaman uploads menampilkan form upload yang akan mereander file akan di prosses dan masuk dalam panel file setelah masuk dalam panel file akan masuk kedetail reander untuk melihat detail hasil file *string match.*

Sequence diagram proses akses halaman panel file dilakukan oleh pengguna untuk mengakses halaman uploads. Pengguna melakukan aksi pada setelah tekan button reander ke halaman panel file, system melanjutkan perintah dari pengguna ke konfigurasi UI untuk mengambil akses tampilan pada library yang kemudian diteruskan untuk ditampilkan kembali kepada pengguna. Sequence diagram proses akses halaman panel file dapat dilihat pada Gambar 7.



Gambar 1 Sequence Diagram Akses Halaman Panel File

Processing adalah perubahan suatu *text* seperti contoh menghilangkan symbol dan spasi lalu *text* dijadikan huruf kecil sama rata sebegai contoh dapat dilihat table 1.

Tabel 1 Desain Sturktur *Text Processing*

Dokumen	Dokumen Pertama	Dokumen	Dokumen Kedua
Case Folding	dalam strategi pemasaran, langkah pertama yang harus dilakukan adalah identifikasi pasar.	*Case Folding*	dalam strategi pemasaran, langkah pertama yangharus dilakukan adalah identifikasi pasar.
Tokenizing	{dalam} {strategi}{pemasaran} {langkah} {pertama} {yang} {harus} {dilakukan} {adalah} {identifikasi} {pasar}	*Tokenizing*	{dalam} {strategi} {pemasaran} {langkah} {pertama} {yang harus} {dilakukan} {adalah} {identifikasi}{pasar}
Punctuation Removal	dalamstrategipemasaranlangkahpertamayang harusdilakukanadalahidentifikasipasar	*Punctuatin Removal*	Dalam strategi pemasaran langkah pertama yang
Stopword Removal	Dalam strategi pemasaran	*Stopword Removal*	dalam strategi pemasaran
Stemming	Dalam strategi pemasaran, langkah pertama yang harus dilakukan adalah identifikasi pasar.	*Stemming*	dalam strategi pemasaran, langkah pertama yang harus dilakukan adalah identifikasi pasar.

Hashing merupakan salah satu cara untuk mengubah karakter *string* menjadi integer yang disebut nilai hash. Proses pengubahan menjadi nilai hash menggunakan fungsi Rolling hash. Persamaan Rolling hash dapat dilihat pada persamaan untuk mengetahui cara kerja pada Rolling hash dapat dilihat pada table dapat dilihat pada table 2.

Tabel 2 Desain Struktur table *Rolling Hash*

Atribut	Nilai Array
Rolling 1	[0] => maka
Hash Pertama	m=109, a = 97, k=107, a=97, basis=11, mod = 10007
	H=c_m*b^(k-1)+c_a*b^(k-2)+c_k*b^(k-3)+c_a*b^(k-4)
	H=109*11^3+97*11^2+107*11^1+97*11^0
	H=145079+11737+1177+97
	H=158090 Mod 10007
	H= 7985
Rolling 2	[1] => akan
Hash Kedua	a = 97, k=107, a=97, n=110, basis=11, mod =10007

e-ISSN: 2685-5615
p-ISSN: 2715-5315

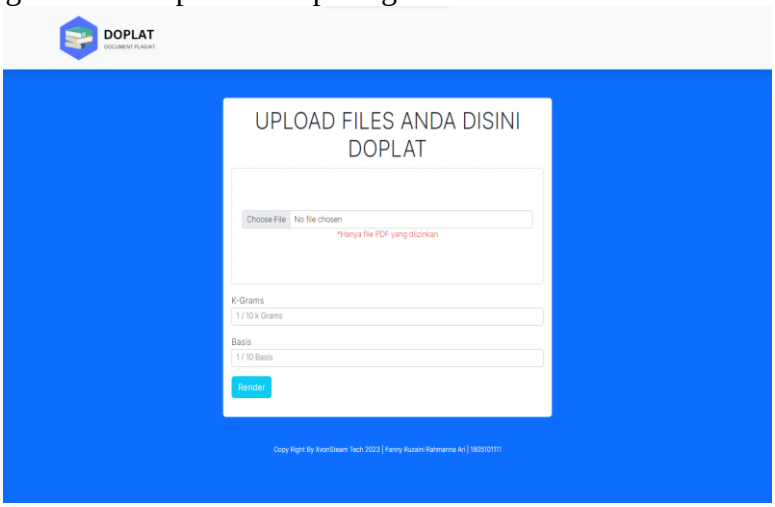
H=c_a*b^(k-1)+c_k*b^(k-2)+c_a*b^(k-3)+c_n*b^(k-4)
H=97*11^3+107*11^2+97*11^1+110*11^0
H=129107+12947+1067+110
H=143231 Mod 10007
H= 3133

Berikut adalah contoh untuk mendapatkan hasil *similarity* dari perhitungan pada table bisa menggunakan rumus sebagai berikut:
S = ((2*C)/(A+B)) * 100
C = Jumlah Hashing yang sama dari A dan B
A = Jumlah Hashing pada dokumen A
B = Jumlah Hashing pada dokumen B
S = ((2*32)/(42+40)) * 100
S = 78.05%

Berdasarkan perhitungan diatas, dapat diketahui persentase *similarity* antara dokumen pertama dan kedua adalah 78.05%.

Implementasi

Halaman upload berfungsi form type file berfungsi sebagai membaca file yang akan di upload jenis yang akan di setujui hanyalah pdf saja jika file tidak berupa pdf akan ditolak. Form type number dengan nama k-grams berfungsi sebagai penentu tokenizing pada huruf yang akan dibuatkan grub berdasarkan nilai k-gram yang diberikan. Form type number dengan nama basis berfungsi sebagai penentu perhitungan pada *similarity* yang akan dihitung dengan tokenizing bisa kita dapat dilihat pada gambar 8.



Gambar 8 Halaman *Upload File*

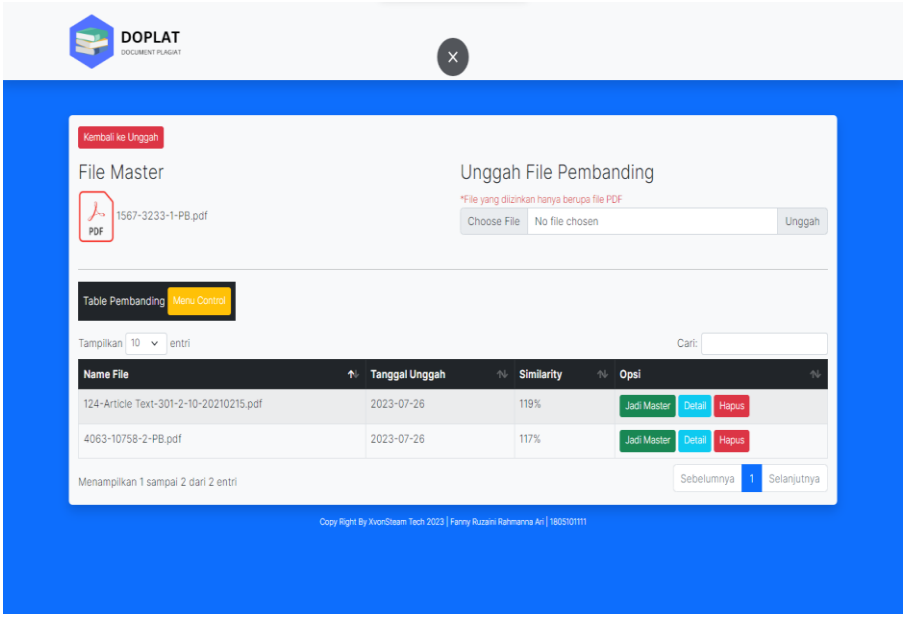
Halaman panel berfungsi mereander file yang sebagai pembanding pada aplikasi doplat sebagai berikut fungsi dari setiap menu nya. Button kembali berfungsi sebagai kembalinya ke menu upload. Preview PDF master Sebagai fungsi menandakan master yang di gunakan saat ini dengan nama pdf yang di upload sebelum nya. Button upload pembanding. Sebagai untuk menambahkan pembanding didalam aplikasi.

Menu control sebagai merubah k-gram dan basis dengan *button* tambahan supaya pengguna dapat menggunakan secara instan dan juga terdapat hapus semua file berfungsi sebagai hapus semua isi file yang ada pada aplikasi. Tabel pembanding sebagai pemberitahu file yang sebagai pembanding dan tanggal upload juga disertakan pada table tersebut ada juga

e-ISSN: 2685-5615
p-ISSN: 2715-5315

nilai *similarity* yang sudah terinclude didalam nya dengan berdasarkan file yang berbeda didalam table juga terdapat opsi.

Button jadi *Master a*kan merubah *Master* secara instan dengan cara satu klik akan merubah master dari pembanding lalu jadi master. *Button Detail a*kan merender file berdarakan master pembanding yang telah di pilih jadi file to file. Button Hapus berfungsi menghapus file pembanding bisa kita lihat gambar 9.



Gambar 9 Halaman Panel

Pengujian

Pengujian sistem Black Box Testing dikenal juga sebagai pengujian fungsional adalah sebuah metode pengujian sistem yang menitikberatkan pengujian pada detail sistem, seperti user interface (antarmuka), fungsi-fungsi yang ada dan kesesuaian sistem berdasarkan perancangan yang sebelumnya telah dilakukan. Metode pengujian Black Box Testing dilakukan tanpa mengetahui struktur kode pada sistem. Hasil pengujian menggunakan metode black box testing ditunjukkan pada tabel 3.

Tabel 3 Desain struktur tabel pengujian black box testing

No	Menu	Hasil		Kesimpulan
		Normal	Error	
1.	Uploads	√		Normal
2.	Panel File	√		Normal
3.	Delete All	√		Normal
4.	Detail	√		Normal
5	Delete File	√		Normal

Pembahasan

Dalam konteks implementasi algoritma *Rabin-Karp* untuk mendeteksi persamaan kata dengan berbasis *website*, salah satu masalah yang relevan adalah masalah plagiatisme. Plagiatisme adalah tindakan mengambil atau menggunakan karya orang lain tanpa izin atau

pengakuan, dan hal ini dapat terjadi dalam konteks teks yang ada di *website*. Dengan adanya algoritma *Rabin-karp*, dapat dilakukan pendekatan untuk mendeteksi kemungkinan adanya plagiatisme dalam teks yang diperoleh dari *website*. Dalam hal ini, setiap teks yang diambil dari *website* dapat dipecah menjadi token atau kata-kata menggunakan proses tokenizing. Kemudian, algoritma *Rabin-Karp* akan membandingkan token-token dari teks yang ada dengan token-token dari sumber teks lainnya, seperti sumber teks asli atau database referensi.

Selanjutnya penelitian ini melakukan pengujian menggunakan pengujian algortima *rabin karp* berbasis *website* dengan cara user meng-uploads file pada halaman utama dan user juga akan memasukan nilai k-grams pada form setelah itu user akan mendapatkan nilai *similarity* dari setiap perbandingan file dan juga dapat melihat processing pada disetiap file dengan tekan button reander. Hasil dari sistem ini diharapkan dapat membantu proses penulisan karya ilmiah untuk mengetahui persamaan yang mendasar pada karya ilmiah.

Simpulan

Dalam penulisan laporan skripsi ini, saya telah mengimplementasikan algoritma *Rabin-Karp* untuk melakukan pencocokan kata berbasis *website*. Berikut adalah beberapa kesimpulan yang dapat diambil dari implementasi ini adalah algoritma *Rabin-Karp* dapat dirancang dengan metode processing, k-grams, tokenizing, fingerprint, nilai ascii dan akan menghasilkan nilai *similarity*. Sistem ini dibangun menggunakan bahasa pemrograman PHP dan CSS. Hasil dari penelitian ini adalah aplikasi algoritma rabin karp untuk mendekteksi persamaan kata berbasis *website*. Hasil dari algoritma rabin karp berbasis *website* yaitu user akan menerima hasil *similarity* dari table panel file dan juga dapat melihat secara detail didalam file reander. Pengujian algortima *rabin karp* berbasis *website* dengan cara user meng-uploads file pada halaman utama dan user juga akan memasukan nilai k-grams pada form setelah itu user akan mendapatkan nilai *similarity* dari setiap perbandingan file dan juga dapat melihat processing pada disetiap file dengan tekan button reander. Hasil dari pengujian sistem yang menggunakan metode *black box* menunjukkan bahwa semua menu pada sistem berjalan secara normal.

Daftar Pustaka

Anardani, S. (2019). *Perancangan Sistem Berorientasi Objek Dengan Pemodelan Uml (Unified Modeling Language) Tools*.

Asmara, J. (2019). Rancang Bangun Sistem Informasi Desa Berbasis Website (Studi Kasus Desa Netpala). *Jurnal Pendidikan Teknologi Informasi (JUKANTI)*, *2*(1), 1–7.

Fan, L., Wang, Y., & Liu, T. (2021). Automatic Test Path Generation and Prioritization using UML Activity Diagram. *Proceedings - 2021 8th International Conference on Dependable Systems and Their Applications, DSA 2021*, 484–490. https://doi.org/10.1109/DSA52907.2021.00072

Filcha, A., & Hayaty, M. (2019). Implementasi Algoritma Rabin-Karp untuk Pendeteksi Plagiarisme pada Dokumen Tugas Mahasiswa. *JUITA : Jurnal Informatika*, *7*(1), 25. https://doi.org/10.30595/juita.v7i1.4063

Herwinsyah, & Witanti, A. (2022). Analisis Sentimen Masyarakat Terhadap Vaksinasi Covid-19 Pada Media Sosial Twitter Menggunakan Algoritma Support Vector Machine (Svm). *Jurnal Sistem Informasi Dan Informatika (Simika)*, *5*(1), 59–67. https://doi.org/10.47080/simika.v5i1.1411

Khaled, F., & Sabeeh, M. (2021). Plagiarism Detection Methods and Tools: An Overview. *Iraqi Journal of Science*, *62*(8), 2771–2783. https://doi.org/10.24996/ijs.2021.62.8.30

Matusea, A. A. F., & Suprianto, A. (2021). Rancang Bangun Aplikasipendaftaran Pasien

Online Danpemeriksaandokterdiklinikpengobatanberbasisweb. *Jurnal Rekayasa Informasi*, *10*(2), 1–14.

Putra, P. N., & Sularno, S. (2019). Penerapan Algoritma Rabin-Karp Dengan Pendekatan Synonym Recognition Sebagai Antisipasi Plagiarisme Pada Penulisan Skripsi. *Jurnal Teknologi Dan Sistem Informasi Bisnis*, *1*(2), 48–58. https://doi.org/10.47233/jteksis.v1i2.52

Rabiman, R., Nurtanto, M., & Kholifah, N. (2020). Design and development E-learning system by learning management system (Lms) in vocational education. *International Journal of Scientific and Technology Research*, *9*(1), 1059–1063.

Rahim, I., Anwar, N., Widodo, A. M., Karsono Juman, K., & Setiawan, I. (2022). Komparasi Fungsi Hash Md5 Dan Sha256 Dalam Keamanan Gambar Dan Teks. *Ikraith-Informatika*, *7*(2), 41–48. https://doi.org/10.37817/ikraith-informatika.v7i2.2249

Rozaq, A., Yunitasari, Y., Sussolaikah, K., & Sari, E. R. N. (2022). Sentiment Analysis of Kampus Mengajar 2 Toward the Implementation of Merdeka Belajar Kampus Merdeka Using Naïve Bayes and Euclidean Distence Methods. *International Journal of Advances in Data and Information Systems*, *3*(1), 30–37. https://doi.org/10.25008/ijadis.v3i1.1233

Sari, D. N., & Utomo, D. P. (2020). Implementasi Algoritma Rabin-Karp Pada Pencarian Quotes Tokoh Terkenal. *Pelita Informatika : Informasi Dan Informatika*, *9*(1), 43–55.

Setiawansyah, Sulistiani, H., Yuliani, A., & Hamidy, F. (2021). Perancangan Sistem Informasi Akuntansi Upah Lembur Karyawan Menggunakan Extreme Programming. *Technomedia Journal*, *6*(1), 1–14. https://doi.org/10.33050/tmj.v6i1.1421

Siswanto, E., & Giap, Y. C. (2020). Implementasi Algoritma Rabin-Karpdan Cosine Similarityuntuk Pendeteksi Plagiarisme Pada Dokumen. *Jurnal Algor*, *1*(2), 16–22. https://jurnal.buddhidharma.ac.id/index.php/algor/index

Styawantoro, I., & Komarudin, A. (2021). *PEMROGRAMAN BERBASIS WEB HTML, PHP 7, MySQLi, Dan Bootstrap 4*. Klaten: Lakeisha.

Supriyadi, E. I., & Asih, D. B. (2020). Implementasi Artificial Intelligence (Ai) Di Bidang Administrasi Publik Pada Era Revolusi Industri 4.0. *JURNAL SOSIAL DAN HUMANIORA UNIVERSITAS*, *2*(2), 12–23. https://doi.org/10.1007/978-3-030-55190-2_49

Utama, M. (2023). Pengembangan Aplikasi Kamus Bahasa Daerah Berbasis Mobile Menggunakan Metode Agile. *Teknologipintar.Org*, *3*(3), 1–18. http://103.140.189.167/index.php/cyberarea/article/view/351%0Ahttp://103.140.189.167/index.php/cyberarea/article/download/351/340

Widiatry, W., & Sari, N. N. kamala. (2019). Rancang Bangun Website untuk Memeriksa Plagiat E-Journal Fakultas Kedokteran Universitas Palangka Raya. *Jurnal CoreIT: Jurnal Hasil Penelitian Ilmu Komputer Dan Teknologi Informasi*, *5*(2), 36. https://doi.org/10.24014/coreit.v5i2.8142

Zulfa, F., Siahaan, D. O., Fauzan, R., & Triandini, E. (2020). Inter-Structure and Intra-Structure Similarity of Use Case Diagram using Greedy Graph Edit Distance. *2020 2nd International Conference on Cybernetics and Intelligent System, ICORIS 2020*, 3–8. https://doi.org/10.1109/ICORIS50180.2020.9320840