

Implementasi Algoritma Rabin-Karp pada Pengecekan Plagiarisme dalam Penulisan Laporan Kerja Magang Mahasiswa ITPLN

Firmansyah Apryadhi^{1)}; Abdurrasyid¹; Emil Setiawan¹*

1. Institut Teknologi PLN, Menara PLN, Jl. Lingkar Luar Barat, Duri Kosambi, Cengkareng, Jakarta Barat, DKI Jakarta 11750

**)Email: firmansyah@itpln.ac.id*

Received: 29 Agustus 2023 | Accepted: 18 Desember 2023 | Published: 29 Januari 2024

ABSTRACT

Internship is a mandatory task to be carried out by students of the Institut Teknologi PLN (ITPLN) as a graduation requirement that must be fulfilled. This activity has a minimum duration of 3 months, and students participating in the internship program are required to compose a report as a crucial component of the internship course. Subsequently, this report will serve as a prerequisite for participating in a presentation after the internship period concludes. During the process of composing internship reports, there are instances where several students are involved in academic violations such as plagiarism. One common form of plagiarism is copying text from other students and incorporating it into their reports with little to no alteration. Therefore, this research aims to perform plagiarism detection in student internship reports using the Rabin-Karp algorithm. The development of this application will follow the waterfall development method and will be implemented in the form of a website platform. These steps are expected to enhance the ability to analyze the level of similarity among student internship report documents.

Keywords: *Plagiarism checker, Rabin-Karp Algorithm, Internship, Plagiarism*

ABSTRAK

Kerja magang merupakan suatu tugas yang wajib dijalankan oleh mahasiswa Institut Teknologi PLN (ITPLN) sebagai persyaratan kelulusan yang harus dipenuhi. Kegiatan ini memiliki durasi minimal selama 3 bulan, dan mahasiswa yang mengikuti program magang diwajibkan untuk menyusun sebuah laporan sebagai komponen penting dalam mata kuliah magang. Selanjutnya, laporan tersebut akan menjadi prasyarat untuk mengikuti sidang setelah periode magang berakhir. Dalam proses penyusunan laporan magang, terdapat kasus dimana beberapa mahasiswa terlibat dalam pelanggaran akademik seperti plagiarisme. Salah satu bentuk plagiarisme yang sering terjadi adalah menyalin tulisan dari mahasiswa lain dan memasukkannya ke dalam laporan mereka dengan tanpa perubahan atau perubahan yang sangat minim. Oleh sebab itu, penelitian ini bertujuan untuk melakukan deteksi plagiarisme pada laporan magang mahasiswa dengan memanfaatkan algoritma Rabin-Karp. Pengembangan aplikasi ini akan mengikuti metodologi pengembangan waterfall dan akan diimplementasikan dalam bentuk platform website. Langkah-langkah ini diharapkan akan meningkatkan kemampuan program studi dalam menganalisis tingkat kemiripan antara dokumen-dokumen laporan magang mahasiswa

Kata kunci: *Pengecekan plagiarisme, Algoritma Rabin-Karp, Magang, Plagiarisme*

1. PENDAHULUAN

Kerja magang adalah suatu proses pembelajaran di mana seseorang memperoleh dan menguasai keterampilan baru, baik dengan atau tanpa bimbingan dari mereka yang sudah terampil di bidang tersebut [1]. Lebih lanjut lagi, kegiatan ini merupakan syarat wajib bagi mahasiswa di kampus ITPLN untuk menyelesaikan studi S1 atau D3. Kerja magang umumnya berlangsung selama 3 hingga 5 bulan, periode ini memungkinkan mahasiswa untuk mempelajari beberapa hal yang bermanfaat, seperti menambah ilmu di luar kelas selama kuliah, mengenal lingkungan pekerjaan yang relevan dengan jurusan atau studi yang diambil, memperkuat pola pikir dan mental bagi peserta, serta merasakan sejumlah manfaat lainnya dari pengalaman magang.

Setelah dilaksanakannya kerja magang selama minimal 3 bulan, mahasiswa diwajibkan menyusun sebuah laporan mengenai aktivitas yang mereka lakukan selama masa magang tersebut. Hal ini ditempuh sebagai langkah validasi atas pelaksanaan kerja magang yang berlangsung selama 3 hingga 5 bulan, dengan tujuan memastikan bahwa proses tersebut berjalan dengan baik. Dalam penulisan laporan, penting untuk merinci dengan mendalam dan sejelas mungkin. Laporan tersebut harus memuat profil perusahaan atau instansi tempat magang, ruang lingkup kerja, tujuan dari kerja tersebut, serta rangkaian program atau kegiatan magang yang harus diuraikan secara rinci dalam log book harian. Tidak hanya itu, laporan juga harus disertai dengan dokumentasi dari setiap kegiatan harian, termasuk hasil dari program-program yang telah dikerjakan. Setelah tahap tersebut selesai, mahasiswa akan menyampaikan laporan magang kepada fakultas terkait dan melanjutkan dengan sidang magang.

Laporan kerja magang yang telah disusun secara cermat perlu diajukan kepada jurusan atau pihak kampus untuk dinilai kelengkapan penulisan serta dokumen yang dicantumkan ataupun dikumpulkan terpisah dari laporan tersebut. Dalam proses penulisan laporan, pihak kampus tidak menyelenggarakan pengecekan plagiarisme. Walaupun demikian, ketika dilakukan pengecekan secara cermat dan manual, terkadang pola penulisan yang serupa, bahkan identik, dapat teridentifikasi tanpa adanya perubahan signifikan hingga laporan ini bisa sampai pada tahap sidang kerja magang.

Dalam menghasilkan laporan kerja magang, tidak bisa dihindari adanya fenomena plagiarisme karena praktek ini kerap kali dilakukan oleh mahasiswa dengan alasan-alasan tertentu. Salah satu bentuk plagiarisme yang sering terjadi adalah penjiplakan laporan mahasiswa lain. Melakukan kecurangan plagiarisme merupakan pelanggaran serius terhadap kode etik yang berlaku dalam lingkungan perguruan tinggi dan program studi. Tindakan ini tidak hanya merugikan mahasiswa yang terkait, tetapi juga merusak integritas sistem akademik. Dalam mengatasi situasi ini, jika didapati mahasiswa yang melakukan plagiarisme pada penulisan laporan kerja magang, maka akan diberikan sanksi yang merugikan pelaku tindakan tersebut.

Seperti yang telah dijelaskan sebelumnya, penelitian ini dilakukan untuk mengkaji sejauh mana tingkat plagiarisme pada laporan kerja magang mahasiswa. Hal ini dilakukan untuk mencegah adanya kecurangan atau kesamaan dalam penulisan laporan kerja magang sehingga mahasiswa bisa lolos hingga tahap sidang magang. Dalam upaya mendeteksi plagiarisme, dapat digunakan sebuah sistem dan metode seperti software pendeteksi plagiarisme. Sistem ini akan memproses dokumen yang ada dan memberikan nilai dan merefleksikan sejauh mana tingkat kesamaan antara dokumen tersebut [13]. Sejalan dengan hal tersebut, penelitian ini akan menerapkan algoritma Rabin-Karp sebagai metode pendeteksian plagiarisme. Pemilihan algoritma Rabin-Karp didasarkan pada fakta bahwa algoritma ini menggunakan metode hash dalam proses pencarian kata. Kendati jarang digunakan untuk pencarian kata tunggal, algoritma ini sangat efisien dalam mengidentifikasi pola tertentu. Fungsi hash mengubah string menjadi enumerasi, sehingga memberikan nilai unik untuk

setiap string tertentu [8]. Dalam rangka menerapkan metode pembuatan aplikasi pengecekan plagiarisme pada laporan kerja magang, penerapan ini akan dilakukan melalui sebuah website.

2. METODE/PERANCANGAN PENELITIAN

2.1. Hashing

Hashing adalah suatu cara untuk mentransformasi sebuah string menjadi suatu nilai yang unik dengan panjang tertentu (fixed-length) yang berfungsi sebagai penanda [4]. Hashing biasa digunakan untuk menentukan atau menghasilkan sebuah nilai yang dikenal sebagai nilai hash. Penentuan nilai hash ini diperlukan untuk menghitung bobot dari sebuah string dan selanjutnya nilai tersebut akan dimasukkan ke dalam metode Rabin-Karp. Metode Rolling Hash adalah salah satu metode hashing yang memberikan kemampuan untuk menghitung nilai hash tanpa mengulangi seluruh string [9].

2.2. K-Gram

K-Gram merupakan kumpulan karakter, kata, atau frasa yang ditemukan dalam suatu dokumen dan penerapannya dilakukan dengan menggeser satu string maju ke depan [5]. Penentuan K-Gram ini kemudian digabungkan dengan perhitungan nilai hash dan dimasukkan ke dalam metode Rabin-Karp. Penggunaan K-Gram berbasis kata bertujuan untuk memprediksi kata-kata berikutnya dalam urutan kata tertentu. Model ini diterapkan pada kata itu sendiri atau sebagai model prediktif yang menetapkan kemungkinan tertentu [10]. Lebih lanjut lagi, K-Gram itu sendiri berfungsi sebagai wadah untuk sekelompok kata yang dipotong menjadi set dengan panjang K tertentu [11].

2.3. Regex

Regex merupakan sebuah teks (string) yang mendefinisikan sebuah pola pencarian sehingga dapat membantu melakukan matching (pencocokan), locate (pencarian), dan manipulasi teks [7]. Regex merupakan fondasi penting yang digunakan untuk melakukan pemrosesan pada sebuah string [15]. Dalam penelitian ini, penggunaan regex diperlukan untuk memastikan bahwa sistem hanya melakukan pendeteksian plagiarisme pada laporan kerja magang dari Bab 1 hingga Bab 5. Pola regex yang diterapkan dalam penelitian ini adalah sebagai berikut:

`/(PENDAHULUAN\sLatar)([s\S]*)(?=\sDAFTAR\sPUSAKA)/g`

Dengan demikian, nilai *hash* tidak akan banyak mempengaruhi isi karena banyaknya kata-kata yang serupa disebabkan oleh format laporan kerja magang itu sendiri.

2.4. Algoritma Rabin-Karp

Algoritma Rabin-Karp adalah salah satu algoritma pencarian string yang dikembangkan oleh Michael O. Rabin dan Richard M. Karp pada tahun 1987 yang menggunakan fungsi hash untuk menemukan pola di dalam string teks [2]. Fungsi hash sendiri akan dipadukan dengan K-Gram sehingga memberikan pola kalimat dengan batasan tertentu kemudian menggerakkan satu string dan mengambil pola lainnya agar dihitung kembali.

Dalam pencarian pola kalimat pada algoritma Rabin-Karp, dibutuhkan rumus dari rolling hash untuk mencari nilai hash pada sebuah pola atau teks. Adapun rumus yang digunakan dalam pengembangan algoritma ini yaitu [6]:

$$t = \sum((v * d^{n-1}) \% 5807) \quad \text{atau} \quad p = \sum((v * d^{n-1}) \% 5807)$$

Rumus *hash* untuk teks

Rumus *hash* untuk pola

Keterangan:

- t = Hasil dari *hash* yang akan didapatkan dari teks
- p = Hasil dari *hash* yang akan didapatkan dari pola
- v = Nilai *ASCII* - 96
- d = Total dari alfabet (a-z)

- n = Panjang karakter pola dan berbentuk *decrement*

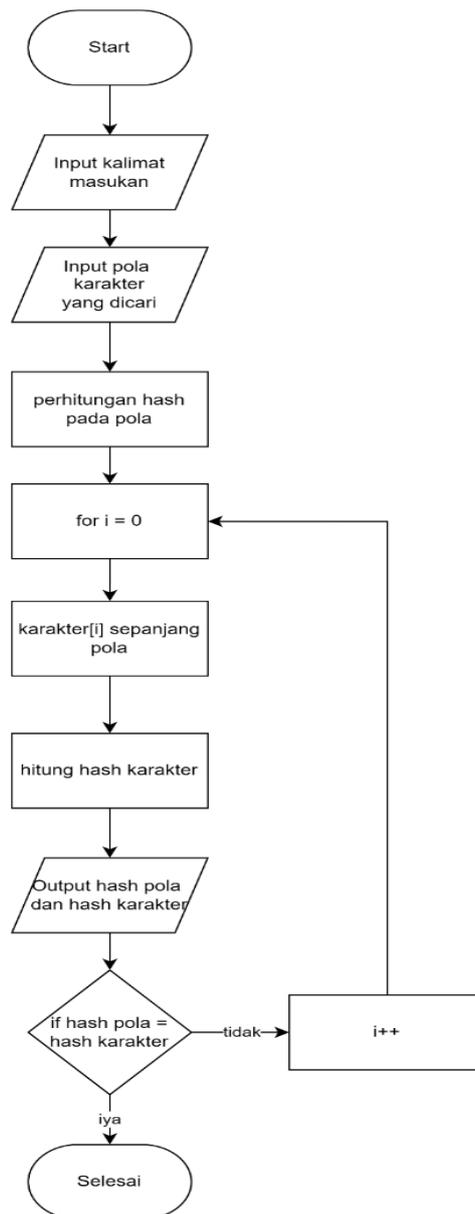
Setelah menghitung nilai *hash* seperti dijelaskan diatas, algoritma akan membandingkan pola atau kata yang sama dengan menghitung dari nilai *hash* dengan rumus sebagai berikut:

$$P = \frac{2 \times n}{a + b} \times 100\%$$

Keterangan:

- n = Jumlah alfabet
- a = pola yang dibuat dengan k-gram pada kalimat/dokumen a
- b = pola yang dibuat dengan k-gram pada kalimat/dokumen b

Berikut adalah langkah-langkah menjelaskan bagaimana algoritma *Rabin-Karp* berfungsi sebagai metode pendeteksian plagiarisme dalam penelitian ini.



Gambar 1. Diagram alir tahapan Rabin-Karp

3. HASIL DAN PEMBAHASAN

Penelitian ini akan menyajikan hasil kajian mengenai persentase kesamaan antara dokumen mahasiswa. Namun, penting untuk dicatat bahwa penelitian ini tidak mengungkapkan kalimat atau karakter spesifik yang dapat diidentifikasi. Sebagai gantinya, bagian ini akan menampilkan tabel hasil pengujian yang mencakup dokumen laporan kerja magang sebagai berikut:

Tabel 1. Tabel hasil pengecekan

Dokumen A	Dokumen B	Presentase
M Irvan	Cendekia Luthfieta	14,34%
	Ika Christine	15,12%
	Kevin Rizki Perdana	16,85%
	M Luthfi Arsyad	14,86%
	A S Suci	16,09%
	Arfan	16,69%
	Cantika Belia	16,43%
	Emil Setiawan	13,40%
	Rasyidana Sulthan F	17,06%

Dengan menerapkan regex sebelum melakukan pendeteksian, tidak hanya memudahkan pengguna dalam membelah sebuah dokumen tetapi juga dapat melakukan perhitungan atau pendeteksian plagiarisme dengan lebih optimal. Selanjutnya, dengan menggunakan nilai $K=5$ pada K -Gram dan menerapkan rolling hash, persentase yang dihasilkan tidak begitu besar maupun kecil.

4. KESIMPULAN DAN SARAN

4.1. Kesimpulan

Berdasarkan hasil dari penelitian yang dilakukan, dapat diambil kesimpulan sebagai berikut:

- Algoritma *Rabin-Karp* mampu mendeteksi plagiarisme dengan cara mencari kesamaan pola. Pendekatan ini melibatkan pencarian *token* terlebih dahulu yang dihasilkan dari jumlah *K-Gram*. Setelah token ditemukan, algoritma *Rabin-Karp* melaksanakan perhitungan nilai *hash* dan mencari pola dengan menghitung jumlah pola yang sama. Dari jumlah tersebut, dapat dihitung persentase kesamaan antar dokumen.
- Mengimplementasikan aplikasi pengecekan plagiarisme dengan menggunakan sebuah *framework* dan menciptakan API membantu dalam pengembangan aplikasi menjadi efisien. Dalam pengembangan dengan memanfaatkan suatu *framework*, terdapat banyak sumber dokumentasi yang berkaitan dengan *framework* yang digunakan dalam penelitian ini. Selain itu, komunitas yang menggunakan *framework* tersebut juga cukup besar. Selanjutnya, pengembangan aplikasi dengan pembuatan API membantu dalam mengatasi segala jenis kesalahan secara terpisah tanpa berdampak pada yang lainnya. Pada penerapan algoritma *Rabin-Karp* yang dilakukan dalam pengujian dokumen laporan kerja magang mahasiswa, ditemukan bahwa rata-rata persentase kesamaan dokumen adalah sebesar 16.51%.

- c. Sebelum melakukan pengecekan dan pengunggahan pada aplikasi, perlu dipastikan bahwa dokumen yang akan digunakan sebagai *input* berada dalam format *.zip* dan di dalamnya terdapat koleksi laporan kerja magang mahasiswa dalam format *.docx*. Selanjutnya, setiap dokumen laporan kerja magang harus mencakup seluruh bagian mulai dari BAB 1 hingga DAFTAR PUSTAKA.

4.2. Saran

Dalam upaya untuk terus menyempurnakan dan mengembangkan penelitian ini di masa depan, terdapat sejumlah saran yang dapat dijadikan panduan. Dengan demikian, dapat dikembangkan lebih maksimal dari penelitian yang telah dilaksanakan ini. Adapun saran pada penelitian ini yaitu sebagai berikut:

- a. Memaksimalkan penggunaan fitur-fitur yang disediakan dalam *framework*, sehingga dapat memaksimalkan performa dari aplikasi berbasis *website* menjadi lebih baik.
- b. Mengembangkan sistem dengan penambahan peran/*role*, yang bertujuan untuk menjadikan aplikasi pengecekan plagiarisme ini menjadi bukti dan syarat dalam pengajuan pengumpulan dokumen kerja magang.
- c. Menambahkan fitur *responsive* atau aplikasi berbasis *mobile*, sehingga pengguna dapat menggunakan aplikasi pada telepon genggam ataupun tablet.
- d. Apabila dibutuhkan, tambahkan fitur untuk pengunduhan hasil dari pengecekan plagiarisme sehingga dapat dijadikan dasar laporan atau dokumen yang disimpan untuk jurusan.
- e. Menambahkan fitur untuk memberikan atau menentukan nilai batasan pada presentasi untuk dijadikan indikasi tingkat plagiarisme pada aplikasi.
- f. Menambahkan jenis-jenis ekstensi dokumen yang dapat diunggah ke dalam aplikasi, lalu memberikan sebuah pesan apabila dokumen tidak memiliki kriteria laporan kerja magang seperti terdapat BAB 1 hingga BAB 5 atau kata DAFTAR PUSTAKA pada dokumen DAFTAR PUSTAKA.
- g. Penerapan dengan menggunakan metode lain yang digabungkan dengan *Rabin-Karp* dapat meningkatkan pendeteksian plagiarisme, contohnya dengan menambahkan metode seperti *Cosine Similarity*. Selain pendeteksian plagiarisme jauh lebih akurat, pendeteksian dengan kedua metode ini akan memberikan waktu pemrosesan yang lebih singkat walau dengan jumlah karakter yang lebih banyak [12].
- h. Menerapkan konsep pemrosesan paralel untuk mempercepat pendeteksian plagiarisme dengan menggunakan kartu grafik (GPU), sehingga memaksimalkan performa dari *runtime* pada saat melakukan pendeteksian plagiarisme dengan skalabilitas yang lebih luas [14].

DAFTAR PUSTAKA

- [1] Anwar Prabu Mangkunegara. Sumber Daya Perusahaan Cetakan kedua belas, 2015.
- [2] Asvarizal Filcha, Mardhiya Hayaty. Implementasi Algoritma Rabin-Karp untuk Pendeteksi Plagiarisme pada Dokumen Tugas Mahasiswa, 2019.
- [3] Nurdin, Rizal, & Rizwan. Pendeteksian Dokumen Plagiarisme Dengan Menggunakan Metode Weight Tree, 2019.
- [4] Hari Bagus Firdaus. Deteksi Plagiat Dokumen Menggunakan Algoritma Rabin-Karp, 2003.
- [5] Ida Widaningrum, Dyah Mustikasari, Rizal Arifin, Erika Diyah Cahyani. Analisa Penggunaan K-Gram pada Karakter, Kata dan kalimat untuk Mendeteksi Kesamaan Dokumen, 2020.
- [6] Ravi, S. Rabin-Karp Algorithm. Programiz, 2020.

- <https://www.programiz.com/dsa/rabin-karp-algorithm>
- [7] Muhardian, A. 2020. Regex. <https://www.petanikode.com/regex/>
 - [8] Ade Ambarwati br ginting, Dito Putro Utomo. Perancangan Aplikasi *Catalog* Wisata Di Sumatera Utara Menggunakan Algoritma Rabin-Karp, 2019.
 - [9] Reynald Karisma Wibowo, Khafiizh Hastuti. Penerapan Algoritma Winnowing Untuk Mendeteksi Kemiripan Teks Pada Tugas Akhir Mahasiswa, 2016.
 - [10] Zudha Pratama , Ema Utami , M. Rudyanto Arief. Analisa Perbandingan Jenis N-Gram Dalam Penentuan Similarity Text pada Deteksi Plagiat, 2017.
 - [11] Sugianto, S. A., Liliana., Rostianingsih, S. Pembuatan Aplikasi Predictive Text Menggunakan Metode N-Gram-Based, 2013.
 - [12] Eric Siswanto, Yo Ceng Giap. Implementasi Algoritma Rabin-Karp Dan Cosine Similarity Untuk Pendeteksi Plagiarisme Pada Dokumen, 2020.
 - [13] G. Srinivasa Raghavan, S. Sowmiya, & P. D. Sathya. A Survey of Plagiarism Detection Methods, 2014.
 - [14] Khandelwal, A., & Sharma, V. Plagiarism Detection using Improved Rabin-Karp Algorithm, 2015.
 - [15] Friedl, J. E. F. Mastering Regular Expressions (3rd ed), 2006.