

Pengaruh Sudut Sdu Turbinu Crossflow Terhadap Daya Dan Efisien Pada Pembangkit C

Zukron Alviandy Rahmadhan¹⁾, Indri Rahamyuni²⁾, Fazrol Rozi³⁾, Ikhsan Yusda Prima Putra⁴⁾
^{1,2,3,4}Politeknik Negeri Padang, Padang, Indonesia

E-mail: *²⁾ indri@pnp.ac.id

Abstract: By requiring students to find solutions to problems that occur in society in different ways from one another, even with the same problem cases. With the same case but different solutions, students look for references to articles or journals with topics that are not much different. Such as searching on the internet, books, as well as from thesis, final assignments and journals belonging to seniors. Therefore, if the final assignment raised by a student has also been appointed by a senior with the same thing, make the reference not as a basic reference for research but more as a cheat material by playing with words so they don't look the same. And for supervisors who do not have time to look for senior final assignments with similar titles to check one by one. Therefore, this makes students uncritical of the topics raised, so that the writing of the final project seems to just make an ordinary assignment paper.

In this study, to measure the final project document of Padang State Polytechnic students, 4 methods of similarity processing were used. That is using the Rabin-Karp, Jaro-Winkler, Hybrid (Rabin-Karp and Jaro-Winkler Combination) and Hybrid algorithms with text mining normalization. Of the four methods compared to which method is suitable for processing final project documents that have more than 10000 words and a relatively short processing time. By examining various aspects, it was found that the Rabin-Karp algorithm is suitable for this case with a shorter processing time and a higher percentage than the Jaro-Winkler algorithm. Therefore, the Jaro-Winkler algorithm is more suitable for systems to check for typo sentences, online essay exams, auto correct typo sentences or suggestions in the search column because the algorithm compares each character.

Keywords: Document similarity, hybrid algorithm, jaro-winkler, rabin-karp, string searching, thesis

Abstrak: Dengan mewajibkan mahasiswa untuk mendapatkan solusi dari permasalahan yang terjadi di masyarakat dengan cara berbeda satu sama lain, walaupun dengan kasus permasalahan yang sama. Dengan kasus yang sama tapi solusi yang berbeda membuat mahasiswa mencari referensi artikel atau jurnal dengan topik yang tidak jauh berbeda. Seperti mencari di internet, buku, maupun dari skripsi, tugas akhir dan jurnal milik senior. Oleh karena itu jika tugas akhir yang diangkat oleh mahasiswa telah diangkat juga oleh seniornya dengan hal serupa menjadikan referensi tersebut bukan sebagai acuan dasar penelitian tapi lebih sebagai bahan contekan dengan memainkan kata-kata agar tidak terlihat sama. Dan untuk dosen pembimbing tidak mempunyai waktu untuk mencari tugas akhir senior dengan judul yang serupa untuk memeriksa satu persatu. Oleh karena itu hal tersebut membuat mahasiswa menjadi tidak kritis terhadap topik yang diangkat, sehingga penulisan tugas akhir terkesan hanya membuat makalah tugas biasa.

Dalam penelitian ini untuk mengukur dokumen tugas akhir mahasiswa Politeknik Negeri Padang menggunakan 4 metode pemrosesan kemiripan. Yaitu menggunakan algoritma Rabin-Karp, Jaro-Winkler, Hybrid (Kombinasi Rabin-Karp dan Jaro-Winkler) dan Hybrid dengan normalisasi *text mining*. Dari keempat metode tersebut dibandingkan metode yang mana yang cocok untuk memproses dokumen tugas akhir yang mempunyai lebih dari 10000 kata dan waktu proses yang relatif singkat. Dengan menguji berbagai aspek didapati algoritma Rabin-Karp cocok untuk kasus tersebut dengan waktu proses lebih singkat dan persentase lebih tinggi ketimbang algoritma Jaro-Winkler. Oleh karena itu algoritma Jaro-Winkler lebih cocok untuk sistem memeriksa kalimat *typo*, ujian *essay online*, auto *correct* kalimat yang *typo* atau *suggestion* pada kolom pencarian karena algoritma tersebut membandingkan tiap karakter.

Kata kunci: Jaro-Winkler, kombinasi algoritma, kemiripan pada dokumen, pencarian string, rabin-karp, tugas akhir

1. Pendahuluan

Pada setiap mahasiswa semester akhir yang sedang mengikuti jenjang pendidikan perkuliahan pasti memiliki kewajiban untuk membuat sebuah tulisan berdasarkan dengan menganalisis hal-hal yang terjadi pada

lingkup masyarakat. Dengan melihat fakta yang ada pada masyarakat dan membuat sebuah tulisan untuk menciptakan bagaimana masalah tersebut bisa terselesaikan dengan solusi yang dituangkan oleh penulis. Tulisan tersebut dinamakan skripsi atau tugas akhir yang menjadi landasan seseorang mahasiswa untuk bisa lulus di jenjang pendidikan perkuliahan.

Dengan mewajibkan mahasiswa untuk mendapatkan solusi dari permasalahan yang terjadi di masyarakat dengan cara berbeda satu sama lain, walaupun dengan kasus permasalahan yang sama. Dengan kasus yang sama tapi solusi yang berbeda membuat mahasiswa mencari referensi artikel atau jurnal dengan topik yang tidak jauh berbeda. Seperti mencari di internet, buku, maupun dari skripsi, tugas akhir dan jurnal milik senior. Oleh karena itu jika tugas akhir yang diangkat oleh mahasiswa telah diangkat juga oleh seniornya dengan hal serupa menjadikan referensi tersebut bukan sebagai acuan dasar penelitian tapi lebih sebagai bahan *contekan* dengan memainkan kata-kata agar tidak terlihat sama. Dan untuk dosen pembimbing tidak mempunyai waktu untuk mencari tugas akhir senior dengan judul yang serupa untuk memeriksa satu persatu. Oleh karena itu hal tersebut membuat mahasiswa menjadi tidak kritis terhadap topik yang diangkat, sehingga penulisan tugas akhir terkesan hanya membuat makalah tugas biasa.

Sejak kemajuan teknologi yang semakin cepat menjadikan penulisan tugas akhir menjadi lebih mudah karena bisa mengetik dan menghapus jika ada kesalahan atau revisi. Dengan bentuk dokumen tugas akhir yang berbentuk digital, maka untuk memeriksa setiap dokumen tugas akhir lebih praktis serta menghemat perlu dibangun sebuah sistem untuk melakukan hal tersebut. Karena tugas akhir berbentuk digital, bisa membuat siswa ataupun mahasiswa melakukan *copy paste* terhadap tugas akhir yang sedang dikerjakan. Ini menyebabkan dosen yang menilai memakan waktu untuk memvalidasi setiap tugas akhir yang ada terhadap tugas akhir senior. Oleh karena itu, dengan menekankan tindakan *copy paste* yang dilakukan oleh mahasiswa perlu adanya proses validasi antar dokumen tugas akhir yang dikumpulkan unik dan diharapkan tidak terjadinya kemiripan.

Pada sistem diharapkan terdapat formulir yang bisa mengunggah kumpulan tugas yang sudah dikerjakan oleh mahasiswa yang berformat PDF dan sekaligus mengelompoknya ke dalam sebuah label. Label bisa berupa nama yang berkaitan dengan tugas akhir mahasiswa atau hal yang mudah diingat. Dosen juga dapat memasukkan daftar kalimat yang mau dihapus, hal ini dilakukan agar sistem menghapus kalimat yang tidak berkaitan dengan isi dari tugas akhir seperti adanya kalimat "Kata Pengantar", kalimat pada lembar pengujian, kalimat yang mengandung jurusan dan lain-lain. Dengan cara mengelompokkan kumpulan tugas akhir yang telah diunggah oleh dosen untuk bisa dicek kemiripan di antara tugas tersebut berdasarkan nama label yang diinginkan. Dan hasil dari pengecekan akan diarsipkan oleh sistem agar mudah di cek kembali atau diunduh oleh dosen yang memberikan tugas.

Dalam mengecek kemiripan antar tugas akhir, sistem akan memanfaatkan kombinasi algoritma Rabin-Karp dan Jaro-Winkler dalam pencarian *string* tiap tugas. Sudah banyak penelitian tentang algoritma ini memiliki efektivitas yang tinggi dalam pengujian kemiripan teks. Sistem pengecekan kemiripan tugas mahasiswa berbasis web akan dikembangkan menggunakan ReactJS yang didukung dengan NodeJS yang berbasis bahasa pemrograman JavaScript dan TypeScript sehingga menghasilkan sistem yang berbasis Web serta menggunakan basis data PostgreSQL sebagai media penyimpanan data.

Penelitian ini diharapkan persentase plagiarisme terhadap tugas akhir yang diangkat oleh mahasiswa tingkat akhir di Politeknik Negeri Padang bisa di minimalisir. Sehingga diharapkan dapat memberi waktu untuk dosen agar memeriksa tugas akhir dengan cepat dan seksama sehingga diharapkan mahasiswa tersebut cocok untuk lulus dari perkuliahan. Dengan kombinasi antar algoritma Rabin-Karp dan Jaro-Winkler diharapkan dapat meningkatkan efektivitas dalam memeriksa tugas yang memiliki banyak *string* serta kalimat *typo* sekalipun. Karena pada algoritma Jaro-Winkler mampu menghitung transposisi antar kata, yang artinya membandingkan kata yang berbeda urutan di kalimat yang dianggap sama[1].

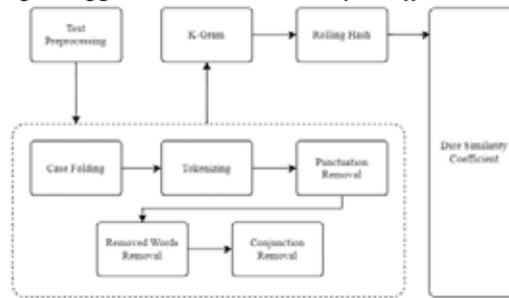
2. Metode

Pada penelitian ini menggunakan 4 metode untuk mengecek kemiripan antar dokumen tugas akhir, yaitu menggunakan algoritma Rabin-Karp, Jaro-Winkler, Hybrid (Kombinasi Algoritma Rabin-Karp dan Jaro-Winkler) dan Hybrid dengan normalisasi. Setiap metode memiliki tahapan-tahapan yang perlu dilalui untuk mendapatkan persentase kemiripan tiap dokumen tugas akhir.

1. Algoritma Rabin-Karp

Rabin-Karp adalah salah satu algoritma pencarian *string* yang dikembangkan oleh Michael O. Rabin dan Richard M. Karp pada tahun 1987 yang menggunakan fungsi *hashing* untuk menemukan *pattern* di dalam *string* teks. Fungsi *hash* sebagai pembanding antara *string* yang dicari (*m*) dengan *substring* pada teks (*n*). Apabila nilai hash keduanya sama maka akan dilakukan perbandingan sekali lagi

terhadap karakter-karakternya[2]. Adapun 3 tahap yang perlu dilalui sebelum mendapatkan persentase kemiripan yang akan dihitung menggunakan *Dice Similarity Coefficient*.



Gambar 1. Alur proses algoritma Rabin-Karp

- *Text Preprocessing*

Pada tahap *preprocessing* sebuah *string* akan masuk ke sub tahap *case folding*, *tokenizing*, *punctuation removal*, *removed words removal* dan *conjunction removal*.

Tabel 1. *Preprocessing* Kalimat

Kalimat	Sedangkan rembulan dan bintang dimalam hari
Preprocessing	rembulanbintangdimalamhari

- K-Gram

K-Gram adalah rangkaian *terms* dengan panjang K. Kebanyakan yang digunakan sebagai *terms* adalah kata. K-Gram merupakan sebuah metode yang diaplikasikan untuk pembangkitan kata atau karakter. Metode K-Gram ini digunakan untuk mengambil potongan-potongan karakter huruf sejumlah K dari sebuah kata yang secara kontinuitas dibaca dari teks sumber hingga akhir dari dokumen.

Tabel 2. Memecah Kalimat Sesuai Nilai K-Gram

K-Gram	5
	(rembu) (embul) (mbula) (bulan) (ulanb) (lanbin) (anbint) (binta) (intan) (ntang) (tangd) (angdi) (ngdim) (gdima) (dimal) (imala) (malam) (alamh) (lamha) (amhar) (mhari)

- *Rolling Hash*

Rolling Hash adalah suatu cara untuk mentransformasikan sebuah string menjadi suatu nilai yang unik dengan panjang tertentu yang berfungsi sebagai penanda string tersebut. Nilai unik bisa berasal dari nilai ASCII pada setiap huruf dan dikalkulasi pada setiap Rolling yang didapatkan menggunakan K-Gram. Untuk nilai Basis serta Modulo didapatkan secara acak dengan ketentuan nilai Basis harus lebih besar dari Modulo dan kedua nilai tersebut tidak boleh terlalu besar untuk mendapatkan akurasi yang tinggi[3].

Tabel 3. *Hashing* Tiap Kalimat K-Gram

Basis	7
Modulo	5001
Kata	rembu
ASCII	r = 114, e = 101, m = 109, b = 98, u = 117
	$H = ((114 \times 7^4) + (101 \times 7^3) + (109 \times 7^2) + (98 \times 7^1) + (117 \times 7^0)) \text{ mod } 5001$
	$H = (273714 + 34643 + 5341 + 686 + 117) \text{ mod } 5001$
	$H = 314501 \text{ mod } 5001$
	$H = 4439$

- *Dice Similarity Coefficient*

Dice similarity coefficient merupakan metode pengukuran yang paling umum digunakan untuk mencari nilai persentase kemiripan pada kedua *string* dengan pendekatan K-Gram dan *Rolling Hash*.

$$S = \frac{2C}{A+B} \quad (1)$$

Keterangan

S : Nilai kemiripan

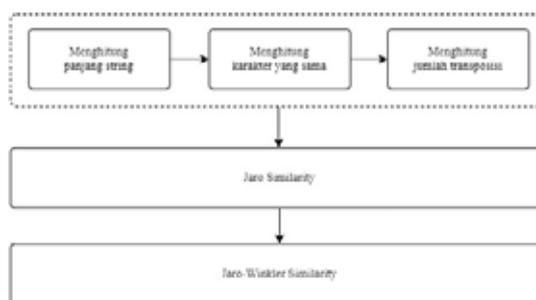
A : Jumlah *fingerprint hash* pada kalimat pertama

B : Jumlah *fingerprint hash* pada kalimat kedua

C : Jumlah *hash* yang sama pada kedua *fingerprint hash*

2. Algoritma Jaro-Winkler

Jaro-Winkler Distance merupakan algoritma untuk mengukur kemiripan antara 2 *string*. Algoritma ini biasanya digunakan untuk pendeteksian duplikat, jika semakin mirip kedua *string* yang dibandingkan maka semakin tinggi pula nilai dari Jaro-Winkler Distance-nya. Nilai normal untuk menandakan tidak adanya kemiripan yaitu 0, dan 1 untuk menandakan adanya kesamaan. Algoritma Jaro-Winkler mempunyai 3 komponen dasar yaitu, menghitung panjang string, menemukan jumlah karakter yang sama di antara 2 *string* dan mencari jumlah transposisi[4]. Pada metode mencari persentase kemiripan antar kalimat menggunakan algoritma Jaro-Winkler mempunyai 3 komponen utama dalam memproses sebuah kalimat, yaitu menghitung panjang kalimat, menghitung huruf yang sama dan menghitung jumlah transposisi.



Gambar 2. Alur proses algoritma Jaro-Winkler

Untuk mendapati nilai transposisi dengan menghitung jumlah huruf yang sama tapi berada di posisi yang berbeda dan dibagi 2 seperti persamaan 1.

$$\frac{d}{2} \quad (2)$$

Keterangan

d : Jumlah huruf yang sama berbeda posisi

Setelah 3 komponen tersebut didapatkan, maka akan dilanjutkan pada tahap mencari nilai *Jaro Similarity*. Menggunakan rumus pada persamaan 3.

$$d_j = \frac{1}{3} \times \left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m} \right) \quad (3)$$

Keterangan

m : Jumlah huruf yang sama persis

|s1| : Panjang *string* pertama

|s2| : Panjang *string* kedua

t : Nilai transposisi

Setelah mencari nilai *Jaro Similarity* maka dilanjutkan dengan melakukan *Jaro-Winkler Similarity* dengan mengukur jarak antar kedua *string* dengan menggunakan *prefix scale* (p) yang mana memberikan penilaian antar 2 *string* yang dibandingkan dan *prefix length* (l) adalah jumlah karakter yang sama dengan dibandingkan per karakter di setiap *string* sampai tidak ditemukannya kesamaan lagi dengan jumlah maksimal adalah 4[5].

$$Sw = Sj + p \times l \times (1 - Sj) \quad (4)$$

Keterangan

Sw : Jaro-Winkler Similarity

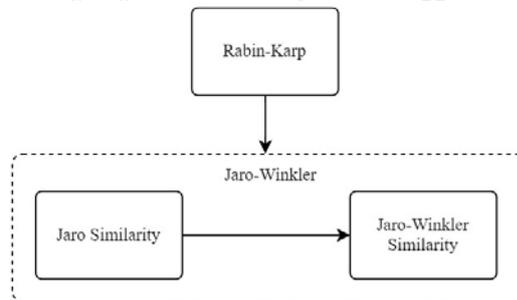
Sj : Jaro Similarity

p : scale factor dengan nilai 0.1

l : karakter yang sama dengan jumlah maksimal 4

3. Hybrid (Kombinasi algoritma Rabin-Karp dan Jaro-Winkler)

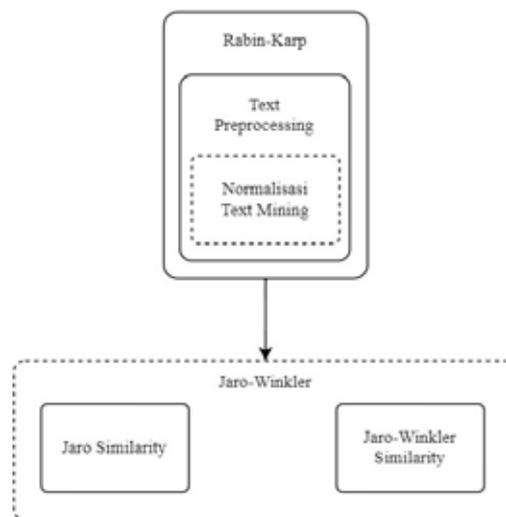
Pada metode ini proses menghitung persentase kemiripan dengan cara mengombinasikan beberapa tahap yang ada di algoritma Rabin-Karp dan dilanjutkan dengan algoritma Jaro-Winkler[6]. Tahapannya yaitu *text preprocessing*, *k-gram*, serta dilanjutkan menggunakan algoritma Jaro-Winkler.



Gambar 3. Alur proses kombinasi algoritma Rabin-Karp dan Jaro-Winkler

4. Hybrid dengan normalisasi

Hampir sama dengan proses sebelumnya yaitu dengan kombinasi algoritma Rabin-Karp dan Jaro-Winkler. Yang membedakan adalah pada tahap Text Preprocessing dengan tambahan sub-proses normalisasi text mining. Normalisasi dilakukan agar jarak jumlah kalimat pada sebuah dokumen tidak terlalu jauh yang mengakibatkan pada dokumen dengan jumlah kalimat yang sedikit akan memiliki jumlah persentase kemiripan tinggi.



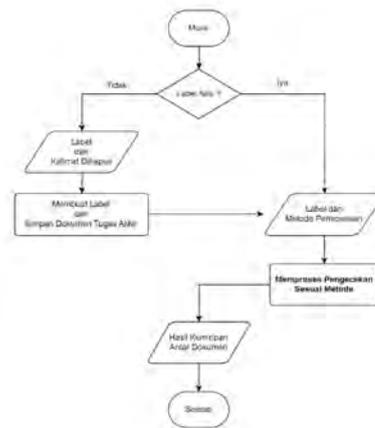
Gambar 4. Alur proses kombinasi algoritma dengan normalisasi

3. Hasil Dan Pembahasan

Pada tahap ini akan dilakukan perancangan serta hasil implementasi sistem yang akan berjalan. Seperti flowchart pemrosesan dokumen tugas akhir, flowchart pemrosesan kalimat, implementasi user interface serta pengujian kemiripan dokumen tugas akhir dan kalimat.

1. Flowchart pemrosesan dokumen tugas akhir

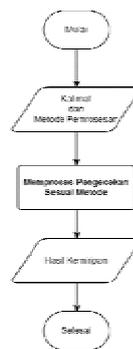
Berikut flowchart bagaimana alur dari pemrosesan dokumen tugas akhir mahasiswa Politeknik Negeri Padang yang akan dicek kemiripannya. Pada proses **Memproses Pengecekan Sesuai Metode** adalah proses dokumen-dokumen tersebut dicek menggunakan metode algoritma Rabin-Karp, Jaro-Winkler, Hybrid (Kombinasi Rabin-Karp dan Jaro-Winkler) dan Hybrid dengan normalisasi text mining yang dipilih oleh user.



Gambar 5. Flowchart pemrosesan dokumen tugas akhir

2. Flowchart pemrosesan kalimat

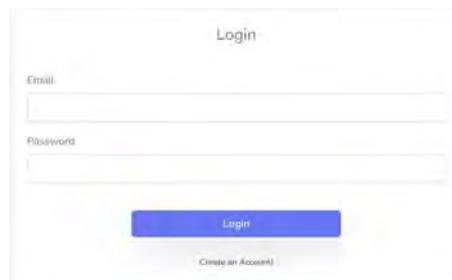
Berikut flowchart bagaimana alur dari pemrosesan membandingkan 2 kalimat yang akan dicek kemiripannya. Pada proses **Memproses Pengecekan Sesuai Metode** adalah proses dokumen-dokumen tersebut dicek menggunakan metode algoritma Rabin-Karp, Jaro-Winkler, Hybrid (Kombinasi Rabin-Karp dan Jaro-Winkler) dan Hybrid dengan normalisasi text mining yang dipilih oleh user.



Gambar 6. Flowchart pemrosesan kalimat

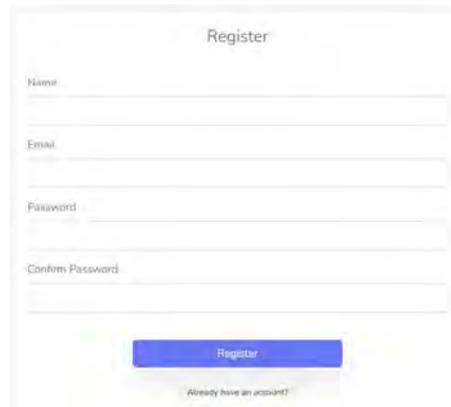
3. Implementasi user interface

Pada gambar 7 merupakan halaman login pada sistem ini, yang menerima 2 input data. Data pertama adalah email dari user yang terdaftar dan yang kedua adalah password atau kata sandi sesuai email yang bersangkutan. Sebelum user bisa masuk ke halaman Dashboard, data yang telah dimasukkan akan melewati validasi dengan mengirim data email dan password ke backend melewati API untuk dicek apakah email dan password tersebut telah terdaftar di basis data atau tidak.



Gambar 7. Halaman login

Pada gambar 8 halaman ini user bisa mendaftar ke sistem dengan memasukkan beberapa data seperti nama, email serta password. Data-data tersebut akan dikirim ke backend untuk divalidasi apakah sudah terdaftar atau belum, sehingga data email yang disimpan pada basis data bersifat unik. Ketika proses registrasi berhasil, maka user akan otomatis ke halaman login dan login menggunakan email dan password yang telah terdaftar.



Gambar 8. Halaman register

Pada gambar 9 terdapat halaman yang menampilkan nama dari aplikasi yaitu “Sistem pengecekan kemiripan dokumen menggunakan algoritma Rabin-Karp dan Jaro-Winkler”. Dengan tampilan bar bagian atas yang menampilkan nama user yang sedang login serta menu untuk logout (keluar) dan mengganti bahasa aplikasi.



Gambar 9. Halaman dashboard

Pada halaman yang terdapat pada gambar 10 menampilkan diagram proses bagaimana proses dari metode pemrosesan kemiripan dalam memproses dokumen sampai bisa mendapatkan persentase kemiripan yang akan digunakan oleh user.



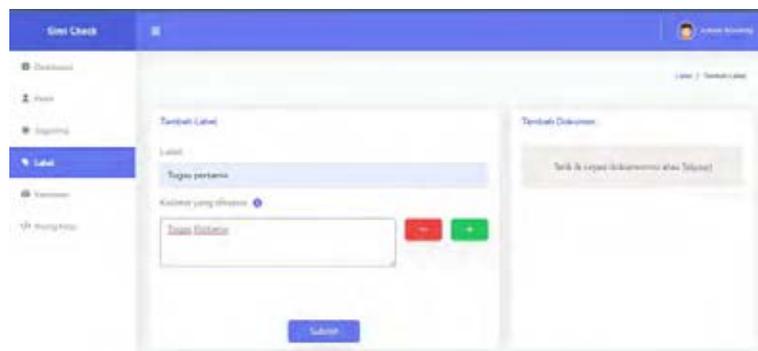
Gambar 10. Halaman algoritma

Pada halaman yang terdapat pada gambar 11 hanya menampilkan label-label yang sudah dibuat. Setiap label yang telah dibuat akan bisa diubah kembali jika user ingin mengubah detail label atau memodifikasi dokumen-dokumen yang terdapat pada label tersebut. Dan pada halaman ini juga terdapat fitur untuk menghapus label serta dokumen-dokumen yang bersangkutan.



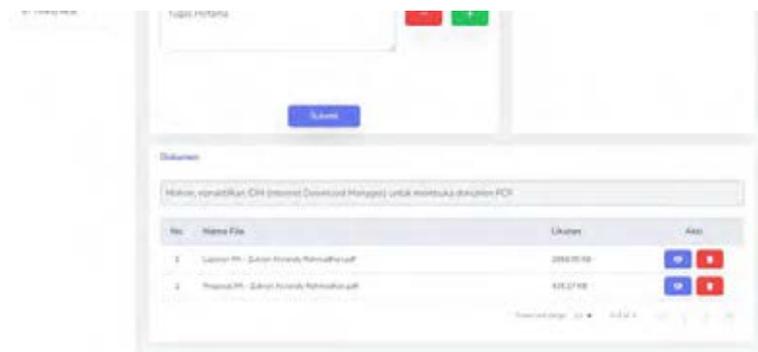
Gambar 11. Halaman label

Setiap dokumen yang ingin dicek kemiripannya harus mempunyai sebuah label. Dan pada halaman yang terdapat pada gambar 12 menampilkan label tersebut dibuat dengan meng-inputkan nama label dan daftar kalimat yang akan dihapus. Formulir kalimat yang dihapus bisa dihilangkan jika tugas akhir ada kalimat yang tidak harus dihilangkan pada text preprocessing. Setelah detail dari sebuah dari label telah diinputkan maka dokumen-dokumen yang ingin dicek harus di unggah.



Gambar 12. Halaman buat label

Setiap halaman yang telah dibuat, maka tetap bisa diedit detail dari label tersebut. Termasuk dokumen-dokumen yang ada, dokumen tersebut bisa dihapus dan dilihat seperti pada gambar 13.



Gambar 13. Halaman edit label

Pada halaman yang terdapat pada gambar 14 menampilkan kemiripan yang telah dibuat dan sudah diproses menggunakan algoritma. Dengan menampilkan nama label dari dokumen-dokumen yang dicek serta nilai minimal dan maksimal kemiripan dari dokumen pada label yang dicek. Algoritma yang dipakai juga akan ditampilkan di halaman daftar kemiripan ini serta status pengecekan.

No	Label	Min	Max	Algorithm	Status	Actions
1	Layanan Pelang (D)	41,47%	42,47%	Hybrid (Platin-Kayu + Lem-Whisker)	Completed	✖
2	Layanan Pelang (D)	30,71%	30,71%	Hybrid with Text Mining	Completed	✖
3	Bank-Bank (D)	11,20%	11,40%	Platin-Kayu	Completed	✖
4	Bank-Bank (D)	28,02%	28,02%	Hybrid (Platin-Kayu + Lem-Whisker)	Completed	✖
5	Layanan Pelang (D)	32,20%	32,20%	Platin-Kayu	Completed	✖
6	Layanan Pelang (D)	30,71%	30,71%	Hybrid (Platin-Kayu + Lem-Whisker)	Completed	✖

Gambar 14. Halaman kemiripan

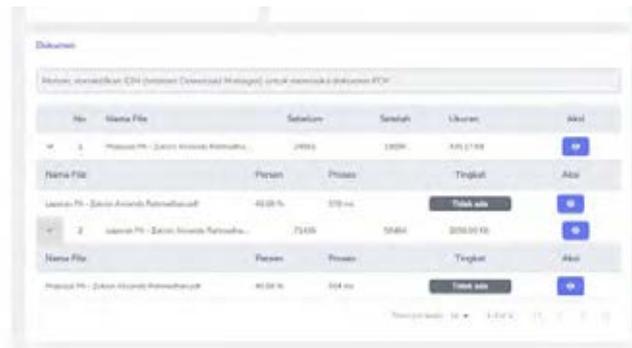
Sebelum membuat proses untuk mengecek kemiripan, terlebih dahulu membuat data label yang mana terdapat dokumen-dokumen yang bersangkutan dengan label tersebut. Dengan demikian label yang telah dibuat akan dipilih menggunakan sebuah dropdown pada halaman buat dan edit kemiripan. Di halaman yang terdapat pada gambar 15 ini juga user akan memilih metode atau proses yang bagaimana untuk mengecek kemiripan dari dokumen dengan label yang telah dipilih.

Gambar 15. Halaman buat kemiripan

Setiap data kemiripan yang telah dibuat dapat diakses untuk melihat detail. Pada halaman yang terdapat pada gambar 16 detail data terhadap label yang sedang di proses akan dideskripsikan secara rinci. Dari minimal dan maksimal persentase yang didapatkan dari dokumen, metode pemrosesan, jumlah kalimat sebelum dan setelah di proses pada tahap text preprocessing, waktu proses dan tingkatannya.

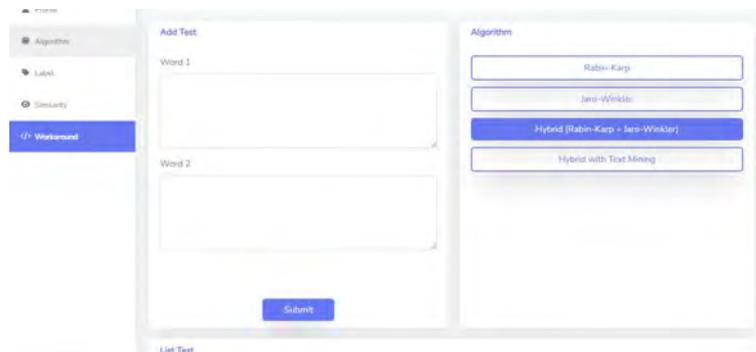
Gambar 16. Halaman detail kemiripan

Di halaman yang terdapat pada gambar 17 detail kemiripan juga terdapat tabel yang menampilkan kemiripan lebih detail antar dokumen yang dibandingkan dan tombol untuk melihat dokumen. Di tabel tersebut menampilkan persentase kemiripan, waktu proses dan level kemiripan dari dokumen pembandingan ke dokumen-dokumen lainnya.



Gambar 17. Daftar dokumen yang diuji kemiripannya

Jika user ingin membandingkan kedua kalimat tanpa perlu mengunggah dokumen. Maka pada halaman yang terdapat pada gambar 18 ini hal tersebut dapat dilakukan. Dengan menerima masukan kalimat pertama dan kedua, serta memilih metode untuk memproses kemiripan maka user dapat mengetahui persentase kemiripan dan tingkatannya seperti gambar 19.

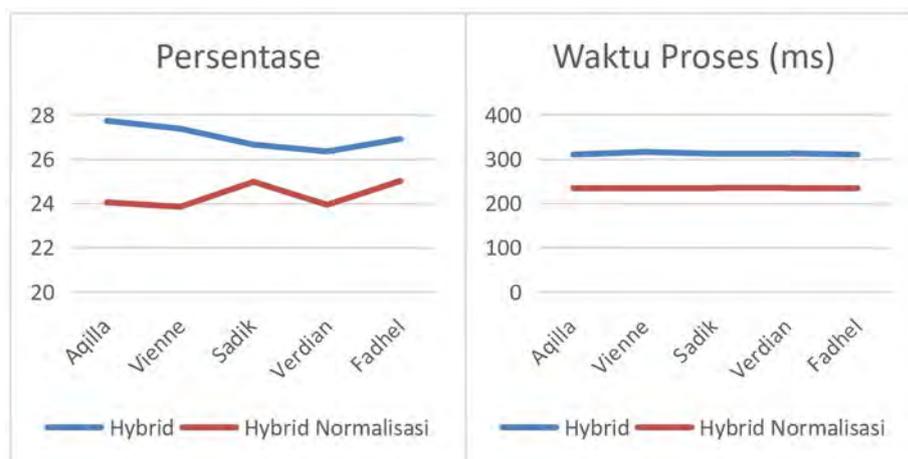


Gambar 18. Halaman ruang kerja



Gambar 19. Daftar tes pada halaman ruang kerja

Masih di halaman ruang kerja yang terlihat pada gambar 20, tapi untuk menampilkan detail pemrosesan yang telah dilakukan pada kedua kalimat dengan cara menampilkan sebuah modal. Pada modal tersebut ditampilkan tahap-tahap serta keluaran yang dihasilkan pada setiap metode yang dilakukan. Dengan demikian user tahu apa yang terjadi di belakang layar



Gambar 22. Grafik persentase dan waktu proses kombinasi algoritma Rabin-Karp dan Jaro-Winkler dan Normalisasi *text mining* pada 6 dokumen tugas akhir

5. Pengujian kemiripan dokumen tugas akhir

Pada pengujian ini mengambil 6 sampel pada dokumen tugas akhir alumni mahasiswa Politeknik Negeri Padang di jurusan Teknologi Informasi dengan berbagai judul yang masih berkaitan dengan membangun sebuah sistem. Dokumen tugas yang diuji memiliki setidaknya lebih dari 80 halaman dan 9000 kata.

1. Metode pemrosesan algoritma Rabin-Karp dan Jaro-Winkler

Berikut hasil pemrosesan dengan 2 dokumen tugas akhir yang menjadi acuan dari 6 dokumen yang ada menggunakan algoritma Rabin-Karp dan Jaro-Winkler. Dengan tugas akhir yang menjadi pembandingan berjudul Implementasi Metode Naive Bayes Clasifier Untuk Optimalisasi Pencarian Dokumen Pada Website Repository Jurusan Teknologi Informasi Politenik Negeri Padang dengan penulis M. Aidil Putra.

Tabel 4. Hasil Perbandingan Menggunakan Algoritma Rabin-Karp dan Jaro-Winkler

Judul	Penulis	Jaro-Winkler		Rabin-Karp	
		Persentase	Waktu (ms)	Persentase	Waktu (ms)
Fraud Detection Pada Aplikasi Akuntansi Pt Tre Jaya Shipping Menggunakan Algoritma K-Means	Aqilla Novia Sukma	9.38	420	2.93	30
Implementasi Algoritma Rabin-Karp Untuk Penilaian Jawaban Essay Otomatis Pada Ujian Online Berbasis Website	Vienne Anggelica Kurnia	27.18	976	84.53	427
Implementasi Cloud Computing Pada Sistem Monitoring Suhu Kelembaban Rumah Wale	Sadik Tulastu	29.08	815	84.89	390
Rancang Bangun Alat Keamanan Pintu Menggunakan E-Ktp, Modul Rfid Dan Aws Ec2	Verdian Ramdhani	28.29	935	85.88	395

Berbasis Nodemcu
Esp8266

Media Interaktif Manual Augmented Reality	Pembelajaran Fabrikasi Berbasis PCE	Fadhel Wedi Pratama	28.81	765	83.69	404
---	-------------------------------------	---------------------	-------	-----	-------	-----

2. Metode pemrosesan kombinasi algoritma Rabin-Karp dan Jaro-Winkler dan Normalisasi text mining

Berikut hasil pemrosesan dengan 2 dokumen tugas akhir yang menjadi acuan dari 6 dokumen yang ada menggunakan kombinasi algoritma Rabin-Karp dan Jaro-Winkler dan Normalisasi text mining. Dengan judul Implementasi Metode Naive Bayes Clasifier Untuk Optimalisasi Pencarian Dokumen Pada Website Repository Jurusan Teknologi Informasi Politenik Negeri Padang oleh M. Aidil Putra

Tabel 5. Hasil Pengujian Menggunakan Hybrid dan Hybrid dengan Normalisasi Text Mining

Judul	Penulis	Hybrid		Hybrid (Normalisasi Text Mining)	
		Persentase	Waktu (ms)	Persentase	Waktu (ms)
Fraud Detection Pada Aplikasi Akuntansi Pt Tre Jaya Menggunakan Shipping Algoritma K-Means	Aqilla Novia Sukma	27.73	310	24.04	234
Implementasi Algoritma Rabin-Karp Untuk Penilaian Jawaban Essay Otomatis Pada Ujian Online Berbasis Website	Vienne Anggelica Kurnia	27.37	316	23.84	234
Implementasi Cloud Computing Pada Monitoring Suhu Kelembaban Rumah Wale	Sadik Tulastu	26.65	312	24.97	235
Rancang Bangun Alat Keamanan Pintu Menggunakan E-Ktp, Modul Rfid Dan Aws Ec2 Berbasis Nodemcu Esp8266	Verdian Ramdhani	26.34	313	23.93	234
Media Interaktif Manual Augmented Reality	Fadhel Wedi Pratama	26.91	310	25.01	233

Pada pengujian dari 6 sampel tugas akhir mahasiswa Politeknik Negeri Padang jurusan Teknologi Informasi mendapati hasil yang tinggi pada saat menggunakan algoritma Rabin-Karp dan sangat rendah dengan menggunakan algoritma Jaro-Winkler. Hal tersebut karena pada sistem pemrosesan yang menggunakan algoritma Jaro-Winkler tidak membandingkan keseluruhan kata antar dokumen melainkan hanya membandingkan kata di posisi yang sama, berbeda dengan algoritma

Rabin-Karp yang membandingkan kata di setiap posisi. Jika kita melihat secara matematis untuk mengetahui performansi antara algoritma Rabin-Karp dan Jaro-Winkler menggunakan notasi Big-O. Dengan asumsi nilai variabel n adalah 10.

- Notasi Big-O algoritma Rabin-Karp

$$O(n^4 + 3n^3 + 3n^2 + n) = 13.310$$

- Notasi Big-O algoritma Jaro-Winkler dengan membandingkan kata di posisi yang sama

$$O(n^3 \times 2n^2) = 200.000$$

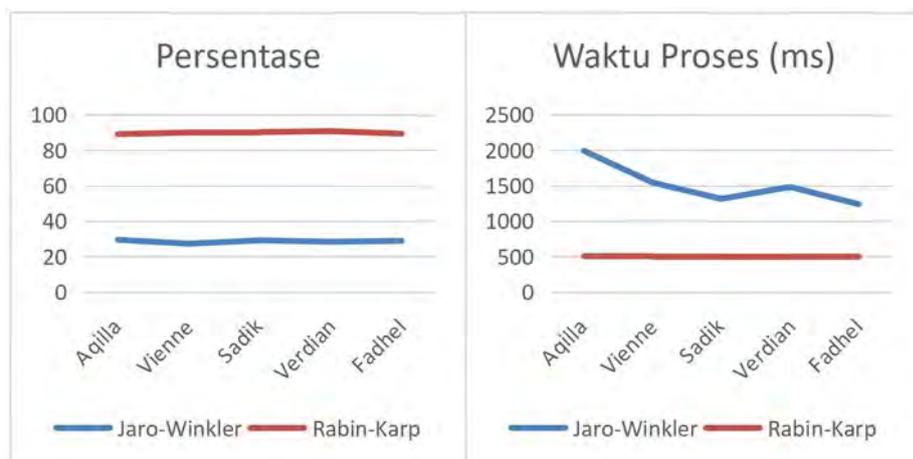
- Notasi Big-O algoritma Jaro-Winkler dengan membandingkan kata di posisi yang berbeda

$$O(n^4 \times 2n^2) = 2.000.000$$

6. Grafik pengujian kemiripan dokumen tugas akhir dengan kalimat dihapus

1. Metode pemrosesan algoritma Rabin-Karp dan Jaro-Winkler

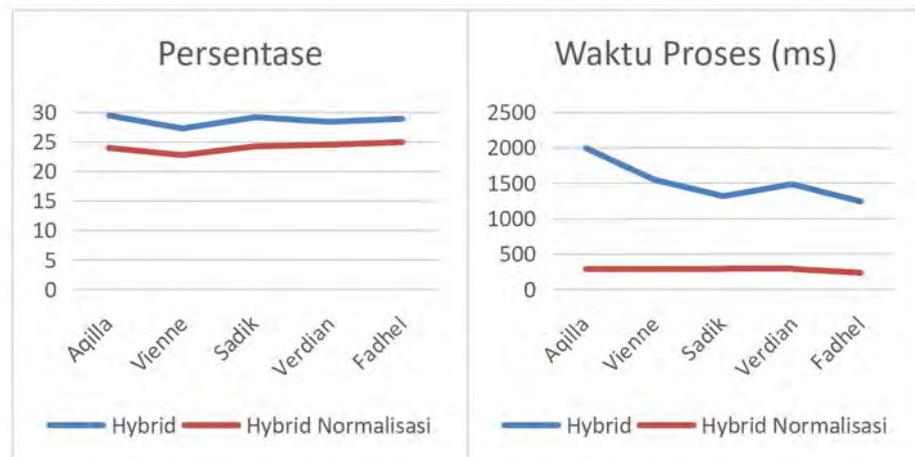
Berikut hasil pemrosesan dengan 2 dokumen tugas akhir yang menjadi acuan dari 6 dokumen yang ada menggunakan algoritma Rabin-Karp dan Jaro-Winkler. Dengan tugas akhir yang menjadi pembanding berjudul Implementasi Metode Naive Bayes Clasifier Untuk Optimalisasi Pencarian Dokumen Pada Website Repository Jurusan Teknologi Informasi Politenik Negeri Padang dengan penulis M. Aidil Putra.



Gambar 23. Grafik persentase dan waktu proses algoritma Rabin-Karp dan Jaro-Winkler pada 6 dokumen tugas akhir dengan kalimat dihapus

2. Metode pemrosesan kombinasi algoritma Rabin-Karp dan Jaro-Winkler dan Normalisasi text mining

Berikut hasil pemrosesan dengan 2 dokumen tugas akhir yang menjadi acuan dari 6 dokumen yang ada menggunakan kombinasi algoritma Rabin-Karp dan Jaro-Winkler dan Normalisasi text mining. Dengan judul Implementasi Metode Naive Bayes Clasifier Untuk Optimalisasi Pencarian Dokumen Pada Website Repository Jurusan Teknologi Informasi Politenik Negeri Padang oleh M. Aidil Putra



Gambar 24. Grafik persentase dan waktu proses kombinasi algoritma Rabin-Karp dan Jaro-Winkler dan Normalisasi *text mining* pada 6 dokumen tugas akhir dengan kalimat dihapus

7. Pengujian kemiripan dokumen tugas akhir dengan kalimat dihapus

Menggunakan 6 sampel yang sama pada sebelumnya dengan menambahkan beberapa kalimat yang akan dihapus pada dokumen tugas akhir. Kalimat yang dihapus adalah kalimat yang sama muncul pada keenam dokumen yang tidak ada kaitannya dengan inti sari yang dibahas. Berikut beberapa kalimat yang dihapus.

- PROGRAM STUDI TEKNIK KOMPUTER JURUSAN TEKNOLOGI INFORMASI POLITEKNIK NEGERI PADANG 2022
- Tugas akhir ini telah diuji dan dipertahankan di depan tim penguji sidang Tugas Akhir
- PROGRAM STUDI TEKNOLOGI REKAYASA PERANGKAT LUNAK JURUSAN TEKNOLOGI INFORMASI POLITEKNIK NEGERI PADANG 2021
- Diajukan untuk memenuhi persyaratan mencapai derajat Sarjana Terapan pada Program Studi Teknologi Rekayasa Perangkat Lunak Jurusan Teknologi Informasi
- Tugas Akhir ini telah diperiksa dan disetujui oleh:
- Tugas Akhir ini telah diuji dan dipertahankan didepan tim penguji sidang Tugas Akhir Diploma IV Politeknik Negeri Padang
- LAPORAN PROYEK AKHIR
- Tugas Akhir

1. Metode pemrosesan algoritma Rabin-Karp dan Jaro-Winkler

Berikut hasil pemrosesan dengan 2 dokumen tugas akhir yang menjadi acuan dari 6 dokumen yang ada menggunakan algoritma Rabin-Karp dan Jaro-Winkler. Dengan tugas akhir yang menjadi pembandingan berjudul Implementasi Metode Naive Bayes Classifier Untuk Optimalisasi Pencarian Dokumen Pada Website Repository Jurusan Teknologi Informasi Politeknik Negeri Padang dengan penulis M. Aidil Putra.

Tabel 6. Hasil Perbandingan Menggunakan Algoritma Rabin-Karp dan Jaro-Winkler

Judul	Penulis	Jaro-Winkler		Rabin-Karp	
		Persentase	Waktu (ms)	Persentase	Waktu (ms)
Fraud Detection Pada Aplikasi Akuntansi Pt Tre Jaya Shipping Menggunakan Algoritma K-Means	Aqilla Novia Sukma	29.42	1989	81.39	309
Implementasi Algoritma Rabin-Karp Untuk Penilaian Jawaban Essay Otomatis Pada Ujian Online Berbasis Website	Vienne Anggelica Kurnia	27.21	1542	83.30	262
Implementasi Cloud Computing Pada Sistem Monitoring Suhu Kelembaban Rumah Wale	Sadik Tulastu	29.12	1313	83.47	275
Rancang Bangun Alat Keamanan Pintu Menggunakan E-Ktp, Modul Rfid Dan Aws Ec2 Berbasis Nodemcu Esp8266	Verdian Ramdhani	28.34	1482	84.74	269
Media Pembelajaran Interaktif Fabrikasi Manual PCE Berbasis Augmented Reality	Fadhel Wedi Pratama	28.85	1238	82.08	263

2. Metode pemrosesan kombinasi algoritma Rabin-Karp dan Jaro-Winkler dan Normalisasi text mining

Berikut hasil pemrosesan dengan 2 dokumen tugas akhir yang menjadi acuan dari 6 dokumen yang ada menggunakan kombinasi algoritma Rabin-Karp dan Jaro-Winkler dan Normalisasi text mining. Dengan judul Implementasi Metode Naive Bayes Clasifier Untuk Optimalisasi Pencarian Dokumen Pada Website Repository Jurusan Teknologi Informasi Politenik Negeri Padang oleh M. Aidil Putra

Tabel 7. Hasil Pengujian Menggunakan Hybrid dan Hybrid dengan Normalisasi Text Mining

Judul	Penulis	Hybrid		Hybrid (Normalisasi Text Mining)	
		Persentase	Waktu (ms)	Persentase	Waktu (ms)
Fraud Detection Pada Aplikasi Akuntansi Pt	Aqilla Novia Sukma	29.42	1989	23.92	286

Tre Jaya Shipping
Menggunakan Algoritma
K-Means

Implementasi Algoritma Rabin-Karp Untuk Penilaian Jawaban Essay Otomatis Pada Ujian Online Berbasis Website	Vienne Anggelica Kurnia	27.21	1542	22.71	286
Implementasi Cloud Computing Pada Sistem Monitoring Suhu Kelembaban Rumah Wale	Sadik Tulastu	29.12	1313	24.20	293
Rancang Bangun Alat Keamanan Pintu Menggunakan E-Ktp, Modul Rfid Dan Aws Ec2 Berbasis Nodemcu Esp8266	Verdian Ramdhani	28.34	1482	24.46	287
Media Pembelajaran Interaktif Fabrikasi Manual PCE Berbasis Augmented Reality	Fadhel Wedi Pratama	28.85	1238	24.89	286

Dari keenam sampel yang diuji menggunakan metode pemrosesan Hybrid (kombinasi algoritma Rabin-Karp dan Jaro-Winkler) dan Hybrid dengan normalisasi text mining tidak mendapati perbedaan persentase yang signifikan, walaupun sudah menghapus beberapa kalimat yang sama antar dokumen. Dengan rata-rata kemiripan pada metode pemrosesan Hybrid 28.58 % dan Hybrid dengan normalisasi sebesar 20.03 % yang artinya berbeda 8.55 % saja pada dokumen yang Aidil sebagai acuan. Untuk hasil persentase ini juga tidak terlalu signifikan dengan algoritma Jaro-Winkler yang rata-rata persentasenya adalah 23.82 %. Jika kita melihat secara matematis untuk mengetahui performansi antara Hybrid (kombinasi algoritma Rabin-Karp dan Jaro-Winkler) dan Hybrid dengan normalisasi menggunakan notasi Big-O. Dengan asumsi nilai variabel n adalah 10.

- Notasi Big-O algoritma Hybrid

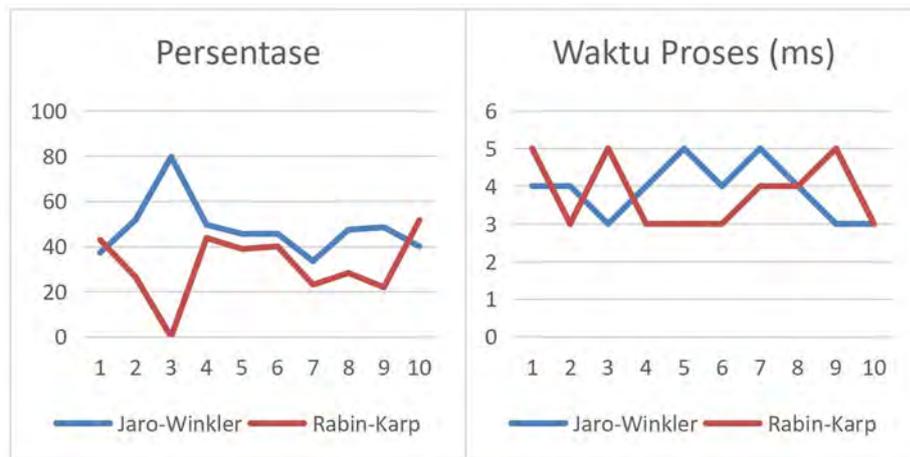
$$O(5n^2 + 4n^3) = 2.000.000$$

- Notasi Big-O algoritma Hybrid dengan normalisasi

$$O(6n^2 \times 4n^3) = 2.400.000$$

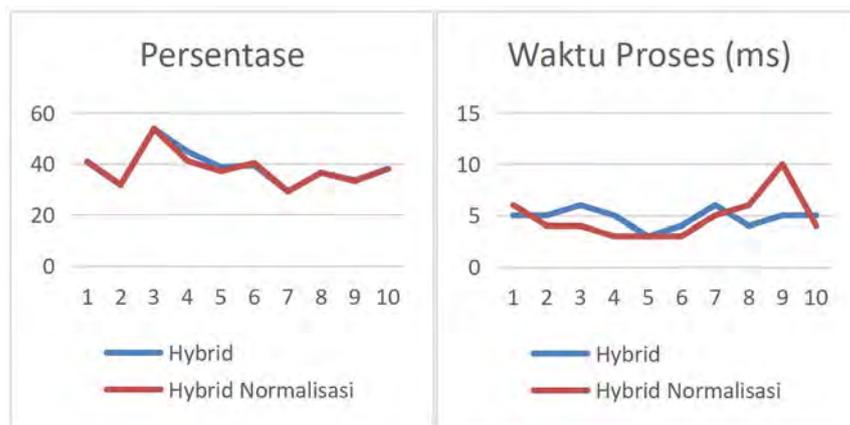
8. Grafik pengujian kemiripan kalimat

1. Algoritma Rabin-Karp dan Jaro-Winkler



Gambar 25. Grafik persentase kemiripan antar kalimat menggunakan algoritma Jaro-Winkler dan Rabin-Karp

2. Metode pemrosesan kombinasi algoritma Rabin-Karp dan Jaro-Winkler dan normalisasi text mining



Gambar 26. Grafik persentase kemiripan antar kalimat menggunakan kombinasi algoritma Rabin-Karp dan Jaro-Winkler dan normalisasi text mining

9. Pengujian kemiripan kalimat

Pada tahap ini menganalisis bagian persentase yang dihasilkan dari 10 kalimat yang dibanding menggunakan 4 metode pemrosesan yaitu dengan algoritma Rabin-Karp, Jaro-Winkler, kombinasi Rabin-Karp dan Jaro-Winkler, serta normalisasi text mining.

Tabel 8. Perbandingan Kemiripan dengan Algoritma Rabin-Karp dan Jaro-Winkler

No.	Kalimat 1	Kalimat 2	Rabin-Karp		Jaro-Winkler	
			Persen	Waktu (ms)	Persen	Waktu (ms)
1	Rembulan di malam hari	Rembulan sangat indah di malam ini	42.86	5	37.25	4
2	Waktu berjalan dengan cepat	Secepatnya waktu berdetik, waktu tetaplah konstan	26.42	3	51.87	4

3	Rembulan	rmblan	0.00	5	79.63	3
4	Jalan jalan dipagi hari	Jalan jalan di siang terik	43.75	3	49.56	4
5	Hari ini adalah hari minggu	Senin adalah hari kerja	38.89	3	45.59	5
6	Hari ini adalah hari minggu	hri ini adlah hri mnggu	40.00	3	45.69	4
7	Deras hujan membuat pohon di belakang rumah tumbang	Pohon tumbang di malam itu	23.08	4	33.52	5
8	Di era dunia maya sekarang rentan terjadi kasus penipuan	Penipuan terhadap orang yang gaptex marak terjadi	28.24	4	47.37	4
9	Indonesia adalah negara agraris	Dengan letaknya Indonesia di khatulistiwa, membuat Indonesia menjadi negara tropis	21.95	5	48.49	3
10	Melihat matahari terbit di atas gunung	Melihat matahari terbenam di tepi pantai	51.61	3	40.04	3

Tabel 9. Perbandingan Kemiripan dengan Algoritma Hybrid dan Hybrid Normalisasi

No.	Kalimat 1	Kalimat 2	Hybrid		Hybrid (Normalisasi Text Mining)	
			Persen	Waktu (ms)	Persen	Waktu (ms)
1	Rembulan di malam hari	Rembulan sangat indah di malam ini	40.83	5	40.83	6
2	Waktu berjalan dengan cepat	Secepatnya waktu berdetik, waktu tetaplah konstan	31.65	5	31.86	9
3	Rembulan	rmblan	53.78	6	53.78	4
4	Jalan jalan dipagi hari	Jalan jalan di siang terik	44.87	5	41.18	3

5	Hari ini adalah hari minggu	Senin adalah hari kerja	38.77	3	37.11	3
6	Hari ini adalah hari minggu	hri ini adlah hri mnggu	39.40	4	40.36	3
7	Deras hujan membuat pohon di belakang rumah tumbang	Pohon tumbang di malam itu	29.18	6	29.18	5
8	Di era dunia maya sekarang rentan terjadi kasus penipuan	Penipuan terhadap orang yang gaptek marak terjadi	36.54	4	36.54	6
9	Indonesia adalah negara agraris	Dengan letaknya Indonesia di khatulistiwa, membuat Indonesia menjadi negara tropis	33.53	5	33.20	10
10	Melihat matahari terbit di atas gunung	Melihat matahari terbenam di tepi pantai	38.06	5	38.06	4

Tabel 10. Rata-rata Persentase dan Waktu Proses

Rabin-Karp		Jaro-Winkler		Hybrid		Hybrid (Normalisasi Text Mining)	
Persen	Waktu (ms)	Persen	Waktu (ms)	Persen	Waktu (ms)	Persen	Waktu (ms)
31.68	3.8	47.90	3.9	38.66	4.8	41.77	5.3

Perbandingan 10 kalimat menggunakan algoritma Rabin-Karp dan Jaro-Winkler pada tabel 10 menunjukkan bahwa algoritma Rabin-Karp dapat memproses 10 kalimat dengan rata-rata waktu proses 3.8 ms, sedangkan algoritma Jaro-Winkler dapat dengan waktu 3.9 ms. Dapat disimpulkan bahwa algoritma Rabin-Karp lebih cepat memproses kemiripan pada kalimat, akan tetapi memiliki persentase kemiripan lebih rendah dari algoritma Jaro-Winkler dengan perbedaan 16.22 %. Dengan membandingkan persentase akurasi beserta waktu pemrosesan yang hanya berbeda 0.3 ms, dapat disimpulkan algoritma Jaro-Winkler lebih baik dalam memproses kemiripan pada kedua kalimat.

Pada pemrosesan menggunakan kombinasi algoritma Rabin-Karp dan Jaro-Winkler mendapati nilai persentase yang lebih rendah ketimbang menggunakan algoritma Jaro-Winkler saja. Dan juga dengan menambahkan normalisasi text mining tidak mendapati hasil yang memuaskan, hal tersebut wajar dikarenakan pada kalimat yang diuji masih terbilang simpel dan belum dikategorikan kalimat kompleks yang mengandung banyak kalimat konjungsi serta berulang-ulang.

Berdasarkan pengujian yang telah dilakukan, persentase kemiripan pada dokumen yang melalui proses algoritma Jaro-Winkler sangat rendah karena sistem hanya membandingkan kata di posisi yang sama. Sedangkan untuk pengecekan pada kalimat mendapati hasil persentase yang jauh lebih tinggi dari pada algoritma Rabin-Karp karena algoritma Jaro-Winkler kemiripan kalimat membandingkan kata yang di posisi berbeda sama halnya seperti algoritma Rabin-Karp.

4. Kesimpulan

Adapun kesimpulan dari hasil penelitian ini yaitu:

1. Dengan adanya sistem ini para dosen pembimbing bisa mengecek kemiripan antar dokumen tugas akhir mahasiswa.
2. Sistem ini mengimplementasikan 4 metode pemrosesan kemiripan, yaitu algoritma Rabin-Karp, Jaro-Winkler, Hybrid (kombinasi Rabin-Karp dan Jaro-Winkler) dan Hybrid dengan normalisasi text mining.
3. Sistem ini bisa mengecek kemiripan dokumen dan kalimat.
4. Berdasarkan pengujian terhadap 6 sampel dokumen tugas akhir, algoritma Rabin-Karp cocok untuk memeriksa kumpulan kata yang lebih dari 10.000 dengan waktu lebih singkat.
5. Algoritma Jaro-Winkler hanya cocok untuk memeriksa kata yang lebih sedikit, dikarenakan algoritma tersebut mengecek tiap karakter pada kata.
6. Dengan memeriksa tiap karakter, algoritma Jaro-Winkler sangat cocok untuk pengimplementasian sistem pencarian data, memeriksa kalimat typo, ujian essay online, auto correct kalimat yang typo atau suggestion pada kolom pencarian.
7. Pada tingkat persentase yang menggunakan kombinasi algoritma Rabin-Karp dan Jaro-Winkler dapat mentolerir kalimat yang sama tapi dengan makna berbeda. Ketimbang algoritma Jaro-Winkler yang nilai persentase tinggi, walaupun makna dari kalimat berbeda.

Ucapan Terima Kasih

Artikel ini dibiayai oleh Dana DIPA POLITEKNIK NEGERI PADANG.

Rujukan

- [1] J. Frando, I. Ruslianto, R. Hidayati, J. Rekeyasa Sistem Komputer, and J. H. Hadari Nawawi, "PENERAPAN JARO WINKLER DISTANCE DALAM APLIKASI PENGOREKSI KESALAHAN PENULISAN BAHASA INDONESIA BERBASIS WEB [1]," 2019.
- [2] J. Priambodo, "Pendeteksian Plagiarisme Menggunakan Algoritma Rabin-Karp Dengan Metode Rolling Hash," *UNIVERSITAS PAMULANG*, vol. 39, no. 1, 2018.
- [3] A. Putera Utama Siahaan, "ANALISIS K-GRAM, BASIS DAN MODULO RABIN-KARP SEBAGAI PENENTU AKURASI PERSENTASE KEMIRIPAN DOKUMEN," 2017.
- [4] A. Prasetyo, W. M. Baihaqi, and I. S. Had, "Algoritma Jaro-Winkler Distance: Fitur Autocorrect dan Spelling Suggestion pada Penulisan Naskah Bahasa Indonesia di BMS TV," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 5, no. 4, p. 435, Oct. 2018, doi: 10.25126/jtiik.201854780.
- [5] P. Novantara and O. Pasruli, "Implementasi Algoritma Jaro-Winkler Distance Untuk Sistem Pendeteksi Plagiarisme Pada Dokumen Skripsi."
- [6] M. A. Yulianto and N. Nurhasanah, "The Hybrid of Jaro-Winkler and Rabin-Karp Algorithm in Detecting Indonesian Text Similarity," *Jurnal Online Informatika*, vol. 6, no. 1, p. 88, Jun. 2021, doi: 10.15575/join.v6i1.640.