

SISTEM DETEKSI *PLAGIARISM* PADA JUDUL TUGAS AKHIR MENGUNAKAN METODE *RABIN-KARP* BERBASIS WEB

Ahmad Subadri, Irfan Pratama

Fakultas Teknologi Informasi, Universitas Mercu Buana Yogyakarta

email: 17121035@student.mercubuana-yogya.ac.id

Fakultas Teknologi Informasi, Universitas Mercu Buana Yogyakarta

email: irfanp@mercubuana-yogya.ac.id

Abstrak

Dengan adanya perkembangan teknologi informasi saat ini membuat pertukaran informasi semakin mudah. Tidak hanya membawa dampak positif saja. Dampak negatif dari penggunaan pun tidak sedikit, salah satunya sebagai sarana untuk melakukan tindakan plagiarisme atau kegiatan mencuri hasil karya orang lain dan mengakuinya sebagai karya sendiri dalam proses penyelesaian tugas akhir. Sehingga diperlukannya sebuah sistem untuk melakukan pengecekan plagiarisme pada judul tugas akhir mahasiswa dengan cepat dan tepat. Metode yang dapat digunakan salah satunya adalah algoritma Rabin-Karp. Algoritma ini memiliki keunggulan pencarian string dengan pola yang panjang. Algoritma dalam sistem ini memiliki langkah-langkah text preprocessing yang terdiri dari beberapa langkah diantaranya case folding, filtering, dan tokenizing. Hasil dari preprocessing akan di proses menggunakan algoritma Rabin-Karp, hasil dari metode ini adalah nilai kemiripan dari judul-judul mahasiswa yang dihitung menggunakan rumus similarity sebagai tahap akhir. Hasil yang dikeluarkan adalah dalam bentuk persentase kemiripan dari judul yang diajukan dengan judul-judul pembanding. Hasil pengujian akurasi pada pendeteksian similaritas judul tugas akhir diperoleh hasil akurasi similaritas sebesar 70 %. Algoritma Rabin-Karp ini masih cukup layak untuk digunakan sebagai suatu metode sederhana untuk mendeteksi kemungkinan terjadinya plagiarisme. Tujuan akhir yang diharapkan adalah dengan adanya sistem ini kemungkinan untuk dilakukannya praktek plagiarisme dapat dideteksi dan dicegah sedini mungkin.

Kata kunci: *Plagiarism, Rabin-Karp, Preprocessing, String Matching.*

1. PENDAHULUAN

Perkembangan teknologi saat ini sangat pesat dan perannya sangat penting untuk mendukung aktivitas manusia agar dapat mengoptimalkan waktu dengan lebih baik. Salah satu contoh teknologi yang membantu aktivitas manusia adalah sistem informasi. Instansi yang bergerak di bidang pendidikan seperti universitas juga membutuhkan dukungan sistem informasi dalam peningkatan mutu pelayanan terhadap proses pendidikan (Alamsyah, 2017). Tugas akhir atau biasa disebut dengan skripsi merupakan istilah yang digunakan untuk menggambarkan suatu karya tulis ilmiah berupa tulisan hasil penelitian seorang sarjana yang disusun untuk memenuhi persyaratan memperoleh gelar sesuai dengan jenjangnya (Asvarizal Filcha, 2019).

Setiap tahunnya Universitas Mercu Buana Yogyakarta khususnya di semester akhir mahasiswa melaksanakan bimbingan tugas akhir, namun terdapat beberapa kendala dalam

hal pengelolaan data administrasi tugas akhir yang saat ini masih terkomputerisasi secara sederhana, dimana data tugas akhir masih dikelola dalam format excel (*.xlsx), DVD, atau word (*.docx), karena pengolahan data yang belum tersistem dan terintegrasi dengan database serta belum adanya fitur deteksi plagiarisme judul tugas akhir yang di ajukan oleh mahasiswa, sehingga membuat dosen sulit menentukan atau mengetahui apakah judul dan laporan tersebut pernah diajukan atau digunakan sebelumnya, karena masih banyak terjadi mahasiswa yang menggunakan judul skripsi yang sudah pernah digunakan angkatan-angkatan yang sebelumnya (Neva Satyahadewi, 2019).

Dengan adanya permasalahan tersebut, banyak terjadi kasus yang disebut plagiat judul tugas akhir atau penggunaan judul skripsi milik orang lain (*plagiarism*), kemudian beberapa mahasiswa yang tidak melaporkan progres pengerjaan tugas akhir atau tidak melakukan

bimbingan secara rutin, maka dari itu diperlukan sebuah sistem informasi yang dapat menyelesaikan masalah tersebut.

Sistem ini diharapkan dapat menyelesaikan permasalahan yang terjadi pada proses pelaksanaan dan manajemen tugas akhir, mengurangi adanya plagiat (*plagiarism*) judul tugas akhir. Untuk itu berdasarkan uraian di atas, maka peneliti dalam hal ini mengambil judul “Sistem Deteksi plagiarism pada judul tugas akhir menggunakan metode rabin-karp berbasis web” dengan harapan sistem yang dibuat dapat menyelesaikan masalah dengan tepat dan memberikan kemudahan pelaksanaan tugas akhir.

2. METODE PENELITIAN

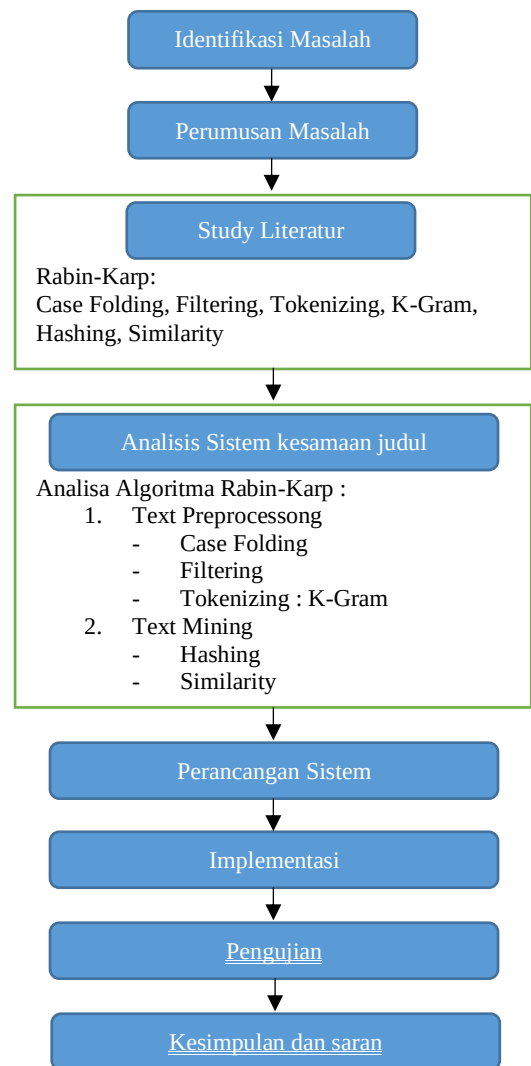
Penelitian ini menggunakan data sekunder di Universitas Mercu Buana Yogyakarta. Data sekunder diperoleh berupa data-data mengenai data judul-judul tugas akhir dari mahasiswa terdahulu, serta data pendukung lainnya yang mendukung penelitian ini.

Penelitian ini menggunakan metode *waterfall* dengan model SDLC (*systems development life cycle*) sebagai metode analisa dan perancangan system informasi (Pratama, 2013). Data yang telah dikumpulkan maka dilakukan analisa. Tahapan ini merupakan tahapan pertama dari fase *waterfall* dari penelitian ini. Pada tahap ini dianalisis proses bisnis mengenai sistem deteksi *plagiarism* pada judul dan manajemen tugas akhir, selanjutnya diusulkan perbaikan dari proses bisnis. Sebagai output dari analisa dan pengembangan akan di buat rancangan *interface* sistem deteksi *plagiarism* pada judul tugas akhir menggunakan metode *Rabin-Karp*. Kemudian akan dilakukan pengkodean dengan menggunakan bahasa pemrograman PHP dan MySQL sebagai tool DBMS nya sehingga menghasilkan output Sistem deteksi *plagiarism* pada judul tugas akhir menggunakan metode *Rabin-Karp* (Satia Suhada, 2017).

3. HASIL DAN PEMBAHASAN

3.1 TAHAPAN PENELITIAN

Metodologi penelitian menjelaskan bagaimana langkah-langkah atau tahapan-tahapan yang akan dilakukan dalam penelitian untuk dapat menjawab perumusan masalah penelitian sesuai konsep sistem temu balik informasi.



Gambar 1 Tahapan penelitian

3.2 PERANCANGAN PROSES

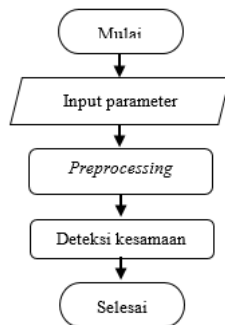
Sistem deteksi plagiat secara umum dirancang untuk dapat mendeteksi kemiripan isi Judul skripsi, yang dimungkinkan kemiripan ini adalah hasil plagiat. Input sistem diperoleh dari judul yang di input oleh mahasiswa atau user. Judul yang di input otomatis akan tersimpan dalam database sistem (Wulansari, 2013).



Gambar 2 Tahapan perancangan proses

3.3 TAHAP IMPLEMENTASI

Pada tahap ini akan dilakukan beberapa proses seperti yang terlihat di *flowchart* di bawah ini:



Gambar 4 Flowchart tahap implementasi

3.4 UJI RABIN KARP PADA SISTEM

Didalam tampilan hasil uji plagiarisme ini menampilkan proses perhitungan plagiarisme hingga muncul hasil berapa persen judul yang di ajukan terdeteksi plagiat. Didalam lembar hasil uji terdapat beberapa sub menu yang menampilkan di setiap masing-masing proses *plagiarism*.

3.4.1 ANALISA METODE RABIN KARP

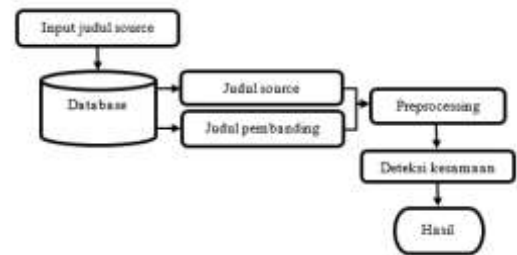
Pada tahap ini dilakukan analisa terhadap alur atau *flow* dari algoritma *rabin-karp*. *Flowchart* dapat dilihat di bawah ini:



Gambar 6 Flowchart analisa Rabin-Karp

3.4.2 PERANCANGAN SISTEM

Secara garis besar sistem pendeteksi kesamaan pada judul skripsi dengan menggunakan metode Rabin-Karp akan dirancang dan kemudian akan dibuat sistemnya seperti yang tergambar pada skema alur kerja sistem berikut ini:



Gambar 3 Flowchart perancangan sistem

3.4.3 TAHAP INPUT PARAMETER

Berikut adalah rancangan arsitektur sistem deteksi tingkat kesamaan dua dokumen dengan menggunakan metode Rabin Karp, seperti yang terlihat pada gambar di bawah ini.



Gambar 5 Flowchart tahap input parameter

Pada *flowchart* program tersebut, digambarkan bahwa proses sistem yang akan dibuat dimulai dari memasukkan judul source atau judul diajukan kemudian dibanding dengan judul pembanding.

Berikut ini peneliti memberikan contoh perhitungan kesamaan judul yang akan di proses dengan menggunakan metode *Rabin-Karp*.

Judul asli atau judul yang di ajukan:

Tabel 1 Judul asli atau judul diajukan

id	judul
A	Sistem Pakar Diagnosa Penyakit Kulit Pada Manusia Menggunakan Metode Naive Bayes Berbasis Web di RS Depok 2 Yogyakarta

Judul pembanding, disini peneliti hanya memberikan 2 judul penguji sebagai contoh:

Tabel 2 Judul pembanding dengan contoh 2

id	judul
PB	Sistem Manajemen Inventori Komputer Menggunakan Near Field Communication Berbasis Android Studi Kasus di STIE Pelita Indonesia Pekanbaru.
PC	Sistem Informasi Registrasi Jenazah Pada Instalasi Forensik Rumah Sakit Abdul Wahab Sjahranie Samarinda Berbasis Web

Nilai K-Gram = 5

Nilai Basis = 3

3.4.4 TAHAP PREPROCESSING

Tahap *Preprocessing* harus dilalui untuk menentukan *keyword* pada judul source dan judul. Pada tahap *preprocessing* terdapat beberapa proses yang dilakukan oleh sistem terhadap dokumen yang diinputkan. Proses-proses tersebut dapat dilihat dari *flowchart* dibawah ini:



Gambar 8 Flowchart tahap preprocessing

1. PROSES CASE FOLDING

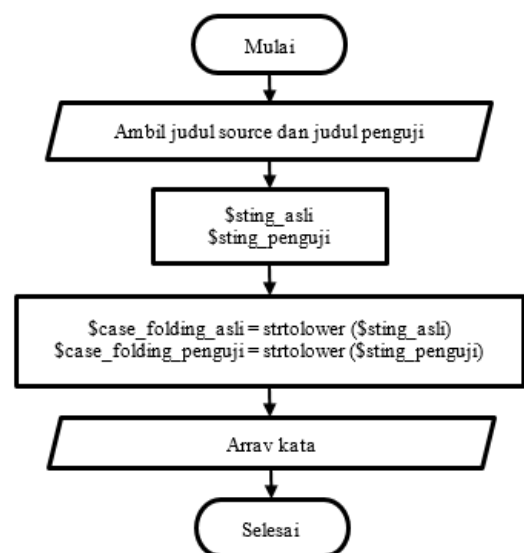
Proses *case folding* mengubah huruf kapital menjadi huruf kecil merupakan tahap pertama yang akan dilakukan dari rangkaian tahapan yang terdapat pada *preprocessing*. Berikut adalah contoh hasil proses *case folding* dari judul yang ada di tabel di atas:

Tabel 3 Hasil proses case folding

Judul A	PB	PC
sistem pakar diagnosa penyakit kulit	sistem manajemen inventori	sistem informasi registrasi

pada manusia menggunakan metode naive bayes berbasis web di rs depok 2 yogyakarta	komputer menggunakan near field communication berbasis android studi kasus di stie pelita indonesia pekanbaru.	jenazah pada instalasi forensik rumah sakit abdul wahab sjahranie samarinda berbasis web
---	---	--

Pada contoh proses *case folding* diatas, kalimat yang terdapat huruf kapitalnya adalah pada kata “Sistem” dan pada kata “Manajemen”. Dimana pada kedua kata tersebut setelah dilakukan proses *case folding* akan diubah menjadi huruf kecil semua menjadi “sistem” dan “manajemen”. *Flowchart* dari *case folding* dapat dilihat dibawah ini:



Gambar 7 Flowchart proses case folding

2. FILTERING

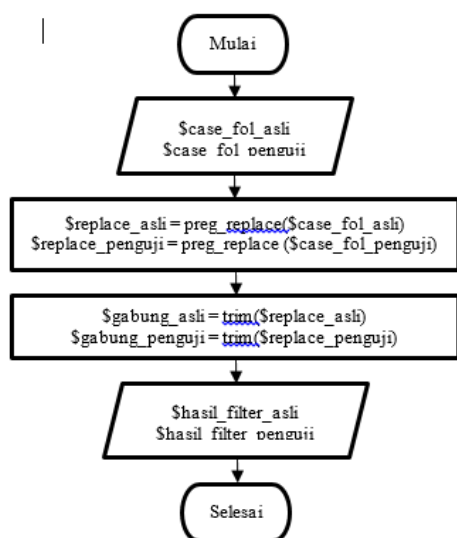
Proses *filtering* dilakukan untuk mengambil kata-kata penting dari hasil *case folding*. Setelah proses *case folding*, judul selanjutnya masuk ke tahap *filtering*, yaitu proses membuang kata yang tidak penting, spasi, dan membuang tanda baca. Berikut ini contoh hasil proses *filtering*:

Tabel 4 Hasil proses filtering

Judul A	PB	PC
sistempakar diagnosapen yakitkulitpa damanusiam	sistemmanaje meninventorik omputermeng gunakannearfi	sisteminforma siregistra sijenazahpa dainstalasif

enggunakan	eldcommnicat	orensikrum
metodenaive	ionberbasian	ahsakitabdu
bayesberbasi	droidstudikas	lwahabsjahr
swebdirsdep	sdistiepelitain	aniesamarin
ok2yogyaka	donesiapekan	daberbasis
rta	baru	web

Pada contoh *filtering* diatas yaitu proses menghilangkan tanda baca dan menghilangkan stop word. Pada contoh judul diatas tanda baca yang di hilangkan adalah spasi, titik (.), koma (,) dan tanda petik (“ ”). Proses *filtering* tersebut seperti yang digambarkan dalam sebuah *flowchart* pada gambar dibawah ini.



Gambar 10 Flowchart proses filtering

3. TOKENIZING

Setelah dilakukan proses *filtering*, selanjutnya adalah proses *tokenizing*, proses ini merupakan proses pembentukan pola kata, dimana pola katanya dalam bentuk gram dengan panjang k. Pada proses *tokenizing* di bagi menja dua sub proses yaitu proses *parsing k-gram* dan proses *hashing*.

Berikut ini adalah proses *tokenizing* yang digambarkan dalam bentuk *flowchart*, seperti yang terlihat pada gambar dibawah ini.



Gambar 9 Flowchart proses tokenizing

Didalam proses *Tokenizing* terdapat proses yang di jalan yaitu *Parsing K-gram* dan *hashing*.

A. PARSHING K-GRAM

Langkah selanjutnya dalam proses *tokenizing* adalah *parsing k-gram*, dimana pada proses ini kata dipecah menjadi potongan-potongan dimana setiap potongan mengandung karakter sebanyak k. Berikut ini adalah hasil contoh proses *parshing k-gram* = 5 dari judul diatas:

Tabel 5 Hasil proses *parshing k-gram* judul A

Judul A
{siste}{istem}{stemp}{tempa}{empak}{mp aka}{pakar}{akard}{kardi}{ardia}{rdiag}{ diagn}{iagno}{agnos}{gnosa}{nosap}{osa pe}{sapen}{apeny}{penya}{enyak}{nyaki} {yakit}{akitk}{kitku}{itkul}{tkuli}{kulit}{ ulitp}{litpa}{itpad}{tpada}{padam}{adama {daman}{amanu}{manus}{anusi}{nusia}{ usiam}{siame}{iamen}{ameng}{mengg}{e nggu}{nggun}{gguna}{gunak}{unaka}{na kan}{akanm}{kanme}{anmet}{nmeto}{me tod}{etode}{toden}{odena}{denai}{enaiv} {naive}{aiveb}{iveba}{vebay}{ebaye}{bay es}{ayesb}{yesbe}{esber}{sberb}{berba}{ erbas}{rbasi}{basis}{asisw}{siswe}{isweb {swebd}{webdi}{ebdir}{bdirs}{dirsd}{irs de}{rsdep}{sdepo}{depok}{epok2}{pok2y {ok2yo}{k2yog}{2yogy}{yogy}{ogyak} {gyaka}{yakar}{akart}{karta}

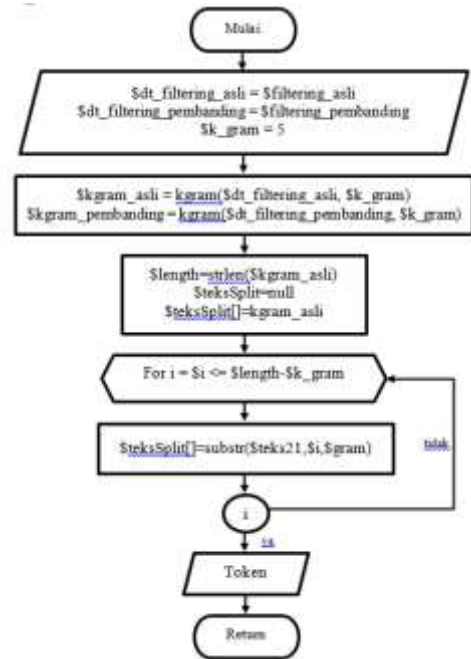
Tabel 6 Hasil proses *parshing* k-gram PB

PB
{siste}{istem}{stemm}{temma}{emman}{mmana}{manaj}{anaje}{najem}{ajeme}{jemen}{emeni}{menin}{eninv}{ninve}{inven}{nvent}{vento}{entor}{ntori}{torik}{oriko}{rikom}{ikomp}{kompu}{omput}{mpute}{puter}{uterm}{terme}{ermen}{rmeng}{mengg}{enggu}{nggun}{gguna}{gunak}{unaka}{nakan}{akann}{kanne}{annea}{nnear}{nearf}{earfi}{arfie}{rfiel}{field}{ieldc}{eldco}{ldcom}{dcomm}{commn}{ommni}{mmnic}{mnica}{nicat}{icati}{catio}{ation}{tionb}{ionbe}{onber}{nberb}{berba}{erbas}{rbasi}{basis}{asisa}{sisan}{isand}{sandr}{andro}{ndroi}{droid}{roids}{oidst}{idstu}{dstud}{studi}{tudik}{udika}{dikas}{ikasu}{kasus}{asusd}{susdi}{usdis}{sdist}{disti}{istie}{stiep}{tiepe}{iepel}{epeli}{pelit}{elita}{litai}{itain}{taind}{aindo}{indon}{ndone}{dones}{onesi}{nesia}{esia}{siape}{iapek}{apeka}{pekan}{ekanb}{kanba}{anbar}{nbaru}

Tabel 7 Hasil proses *parshing* k-gram PC

PC
{siste}{istem}{stemi}{temin}{eminf}{minfo}{infor}{nform}{forma}{ormas}{rmasi}{masir}{asire}{sireg}{iregi}{regis}{egist}{gistr}{istra}{stras}{trasi}{rasij}{asije}{sijen}{ijena}{jenaz}{enaza}{nazah}{azahp}{zahpa}{ahpad}{hpada}{padai}{adain}{dains}{ainst}{insta}{nstal}{stala}{talas}{alasi}{lasif}{asifo}{sifor}{ifore}{foren}{orens}{rensi}{ensik}{nsikr}{sikru}{ikrum}{kruma}{rumah}{umahs}{mahsa}{ahsak}{hsaki}{sakit}{akita}{kitab}{itabd}{tabdu}{abdul}{dulw}{dulwa}{ulwah}{lwaha}{wahab}{ahabs}{habsj}{absja}{bsjah}{sjahr}{jahra}{ahran}{hrani}{ranie}{anies}{niesa}{iesam}{esama}{samar}{amari}{marin}{arind}{rinda}{indab}{ndabe}{daber}{aberb}{berba}{erbas}{rbasi}{basis}{asisw}{siswe}{isweb}

Pada contoh *parshing* k-gram diatas adalah proses pembentukan pola kata dalam bentuk gram dengan panjang karakter k=5, sehingga isi kalimat pada judul asli dan judul uji dirubah dalam bentuk kgram. Berikut ini adalah *flowchart* dari proses *parsing* k-gram. Proses *parsing* k-gram digambarkan pada sebuah *flowchart* berikut ini :



Gambar 11 Flowchart proses *parshing* k-gram

B. PROSES HASHING

Pada proses ini hash berfungsi untuk mengkonvert setiap string menjadi bilangan. Dengan cara mengalikan nilai ASCII hasil huruf hasil k-gram dengan basis bilangan tertentu, dimana basis bilang merupakan bilangan prima. Metode *Rabin-Karp* didasarkan pada fakta jika dua buah *string* sama maka harga hash valuenya pasti sama.

Sebagai contoh kita ambil kata dari hasil k-gram yang pertama pada judul A, yaitu kata {siste}. Contoh proses *hashing* untuk menghitung nilai hash dari kata {siste}, dengan nilai kgram= 5 dan basis=3.

Nilai ASCII dari kata {siste}

ASCII (s) = 115

ASCII (i) = 105

ASCII (s) = 115

ASCII (t) = 116

ASCII (e) = 101

Rumus:

$H(C1...CK) = C1.b^{(K-1)} + C2.b^{(K-2)} + ... + C(k-1).b(k) + ck$

Hitung:

$$H = (115.3^4) + (105.3^3) + (115.3^2) + (116.3^1) + (101.3^0)$$

$$H = (115.81) + (105.27) + (115.9) + (116.3) + (101.1)$$

$$H = (9315) + (2835) + (1035) + (348) + (101)$$

$$H = \mathbf{13634}$$

Jadi nilai hash pada judul A *k-gram* yang pertama adalah 13634, proses perhitungan nilai hash di ulang kembali hingga *k-gram* keseluruhan dihitung. Berikut ini adalah nilai hasil hash dari judul asli dan hash judul uji yang sudah di lakukan *Group by* yang mana fungsinya adalah untuk mengelompokkan data dalam sebuah kolom yang ditunjuk. Fungsi ini akan menghasilkan kelompok data dengan menghilangkan data yang sama dalam satu tabel. Maka apabila dalam satu kolom terdapat beberapa data yang sama maka data yang akan ditampilkan hanya salah satu:

Tabel 8 Hasil proses *hashing* judul A

Judul A
{13634}{13066}{13795}{13537}{12530}{13144}{13059}{12061}{12717}{12247}{13273}{12227}{12492}{12076}{12754}{13345}{13406}{13355}{12241}{13249}{12638}{13476}{13814}{12146}{12984}{13059}{13767}{13229}{13798}{13060}{13036}{13690}{12991}{11854}{12101}{12120}{12904}{12330}{13516}{13927}{13451}{12518}{12142}{12958}{12504}{13079}{12604}{12890}{13738}{12893}{12058}{12704}{12227}{13221}{13033}{12713}{13706}{13027}{12213}{12457}{12929}{12155}{12991}{13579}{12164}{12064}{12476}{13958}{12585}{13310}{12082}{12547}{13203}{12022}{12371}{13643}{13082}{13831}{13653}{12156}{12040}{12406}{13019}{13654}{13371}{12275}{12575}{13303}{12804}{11542}{8746}{14185}{13259}{12901}{13788}{12077}{12757} = 97

Tabel 9 Hasil proses *hashing* judul PB

PB
{13634}{13066}{13792}{13528}{12506}{13072}{12835}{12119}{12895}{12056}{12707}{12468}{12971}{12544}{13190}{12950}{13451}{13734}{12642}{13488}{13841}{13446}{13474}{12832}{13098}{13409}{13355}{13692}{13969}{13577}{12653}{13519}{12958}{12504}{13079}{12604}{12890}{13738}{12893}{12059}{12707}{12217}{13194}{12954}{12237}{12269}{13344}{12430}{12603}{12405}{12781}{12208}{12434}{13350}{13176}{13138}{13043}{12504}{12108}{12377}{13658}{1288

7}{13260}{12905}{12082}{12547}{13203}{12022}{12349}{13586}{12913}{13338}{12180}{13074}{12592}{13591}{13187}{12705}{12700}{13905}{13877}{13540}{12304}{12729}{12787}{12460}{13914}{13912}{13421}{12423}{13070}{13807}{13577}{12651}{12543}{13202}{12487}{13023}{12935}{13390}{12093}{12818}{13040}{12505}{13320}{13084}{12634}{13460}{12542}{12208}{13163}{12371}{12667}{12114}{12888} = \mathbf{115}
--

Tabel 10 Hasil proses *hashing* judul PC

PB
{13634}{13066}{13788}{13529}{12501}{13071}{12840}{13114}{12709}{13456}{13500}{12912}{12350}{13582}{12906}{13318}{12368}{12675}{13093}{13879}{13797}{13309}{12326}{13517}{12703}{12716}{12487}{13022}{12448}{13870}{12064}{12718}{12987}{11855}{12109}{12143}{12955}{13458}{13741}{13393}{12096}{12819}{12324}{13515}{12701}{12698}{13423}{13401}{12608}{13395}{13572}{12880}{13222}{13769}{13720}{12826}{12098}{12828}{13328}{12136}{12935}{12904}{13314}{11862}{12134}{12685}{13859}{13243}{13583}{11947}{12376}{11953}{12392}{13476}{12580}{12092}{12810}{13259}{12190}{13096}{12667}{12583}{13320}{12120}{12899}{12310}{13456}{12764}{12878}{12018}{11852}{12082}{12547}{13203}{12022}{12371}{13643}{13082} = \mathbf{98}

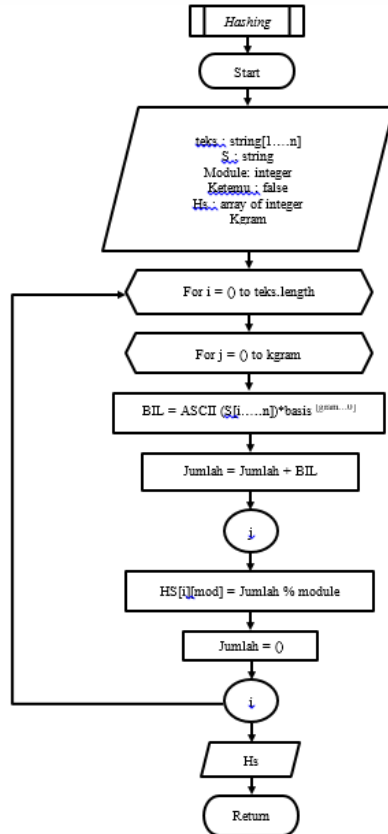
Setelah didapatkan hasil *hashing* dari *group by* sistem akan menjumlah total hasil *hashing* yang akan di gunakan untuk menghitung *similarity*, hasil ditunjukan di angka terakhir setiap tabel.

Hash yang sama antara $A \cap B$ dan $A \cap C$

Tabel 11 Hasil persamaan AB dan AC

$A \cap B$	$A \cap C$
{12022}{12082}{12504}{12547}{12604}{12890}{13066}{12958}{12893}{13079}{13203}{13634}{13738} = \mathbf{13}	{12371}{12022}{12082}{12547}{13066}{13082}{13203}{13634}{13643} = \mathbf{9}

Hasil kesamaan diatas di dapat dari perbandingan antara **source:B** dan **source:C** dengan fungsi *query inner join* sehingga di dapat hasil 13 dan 9. Gambaran proses *hashing* seperti yang terlihat pada *flowchart* berikut ini:



Gambar 12 Flowchart proses *hashing* data

C. SIMILARITY

Setelah diketahui nilai *hashing* nya, jumlah hash pada judul A 97, jumlah hash pada PB 115 dan jumlah hash pada PC 98, hash yang sama pada kedua judul $\sum A \cap B = 13$ dan $\sum A \cap C = 9$. Proses selanjutnya adalah menghitung *similarity* yaitu tingkat kesamaan judul A dengan judul penguji PB PC.

$$AB = A \cap B: 13$$

$$AC = A \cap C: 9$$

$$A = \sum A: 97$$

$$B = \sum PB: 115$$

$$C = \sum PC: 98$$

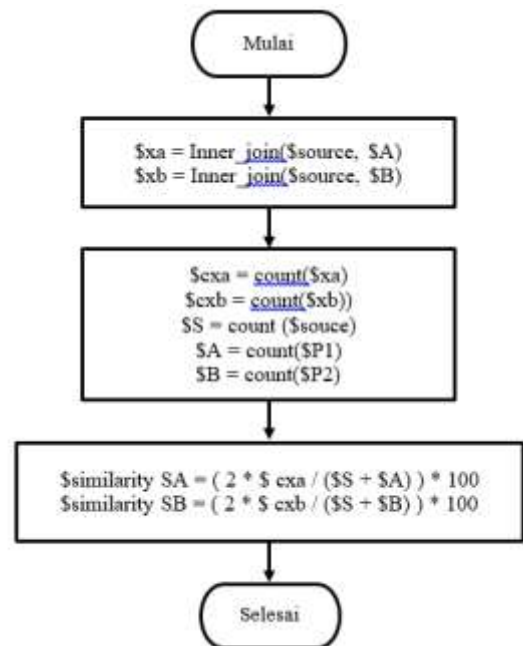
$$\begin{aligned} \text{Similarity AB} &= (2 * AB / (A + B)) * 100 \\ &= (2 * 13 / (97 + 115)) * 100 \\ &= (26 / 212) * 100 \\ &= 0.122642 * 100 \\ &= 12.26415 \% \end{aligned}$$

$$\text{Similarity AC} = (2 * AC / (A + C)) * 100$$

$$\begin{aligned} &= (2 * 9 / (97 + 98)) * 100 \\ &= (18 / 195) * 100 \\ &= 0.092308 * 100 \\ &= 9.230769 \% \end{aligned}$$

Hasil perhitungan *similarity* dari Judul A dengan Pembanding B diatas (AB) adalah **12.26415 %** dan Judul A dengan Pembanding C diatas (AC) adalah **9.230769 %**, dan judul yang diuji bisa dikatakan tidak plagiat, karena tingkat kesamaannya sangat rendah.

Gambaran proses deteksi kesamaan (plagiarisme) judul dapat dilihat pada *flowchart* berikut ini:



Gambar 13 Flowchart *Similarity*

3.5 DESIGN SYSTEM INTERFACE USER

Design tampilan *Frontend user* seperti yang terlihat pada gambar dibawah ini. Halaman utama dirancang dengan beberapa menu yang dapat diakses semua user, terdiri dari halaman *home* atau halaman utama itu sendiri, halaman *about* yaitu halaman yang dirancang untuk informasi tentang sistem tersebut, halaman *timeline* merupakan halaman yang dirancang untuk menampilkan timeline tugas akhir, halaman *announcement* halaman yang dirancang untuk pengumuman, halaman *download* yaitu halaman yang dirancang untuk *download* file yang di butuhkan dalam tugas akhir, halaman *contact* halaman yang di rancang untuk informasi *contact* sistem, dan halaman *login* berisi halaman yang di rancang sebagai *form* akses *dashboard user*.



Gambar 14 Design interfase user

Design form login user ini di rancang sebagai form akses untuk masuk kedalam dashboard masing masing user.



Gambar 16 Design form login user

Tampilan hasil uji *rabin-karp* ini menampilkan proses perhitungan plagiarisme hingga muncul hasil berapa persen judul yang di ajukan terdeteksi plagiat.



Gambar 17 Design hasil plagiarisme

Menghitung hasil akhir *similarity* untuk mendapatkan berapa persen tingkat plagiat dari judul yang di ajukan tersebut, persentase tersebut yang dijadikan sebagai patokan dan penanggulangan sejak dini tingkat pengguna judul yang sama. Hasil bisa dilihat gambar dibawah ini:



Gambar 15 Design tampilan hasil similarity

Tampilan diagram ini menampilkan hasil persentase yang sudah di dapatkan untuk menggambarkan secara visual persentase plagiarisme.



Gambar 18 Design diagram bar hasil

Apakah judul yang di ajukan termasuk ke plagiarisme ringan, plagiarisme sedang, atau plagiarisme berat, yang mana masing-masing memiliki kriteria sendiri :

- Plagiarisme ringan $\leq 30\%$ kesamaan.
- Plagiarisme sedang $>30\% \leq 70\%$ kesamaan.
- Plagiarisme berat $> 70\%$ kesamaan.



Gambar 19 5 judul persentase tertinggi

4. KESIMPULAN

Berdasarkan hasil dari pembahasan dan uji coba sistem maka peneliti dapat menyimpulkan bahwa Algoritma Rabin-Karp berhasil diimplementasikan pada sistem deteksi *plagiarism* pada judul tugas akhir berbasis web. Sistem ini berhasil mendeteksi dan menampilkan persentase kemiripan judul antar mahasiswa. Hasil perhitungan akurasi dengan metode Rabin-Karp mencapai 70% akurat. Dan algoritma yang di gunakan pada sistem ini tidak memiliki perbedaan persentase saat urutan perbandingan diubah.

5. REFERENSI

- Alamsyah, N. (2017). PERBANDINGAN ALGORITMA WINNOWING DENGAN ALGORITMA RABIN KARP UNTUK MENDETEKSI PLAGIARISME PADA KEMIRIPAN TEKS JUDUL SKRIPSI. *Technologia*, 124-134.
- Andi Riansyah, E. N. (2017). Penerapan Sistem Informasi Monitoring Tugas Akhir Untuk Memantau Perkembangan Tugas Akhir Mahasiswa Universitas Islam Sultan Agung. *Jurnal Transistor Elektro dan Informatika (TRANSISTOR EI)*, 100-109.
- Asvarizal Filcha, M. H. (2019). Implementasi Algoritma Rabin-Karp untuk Pendeteksi Plagiarisme pada Dokumen Tugas Mahasiswa. *JUITA p-ISSN: 2086-9398 (print); e-ISSN: 2579-9801 (online); Volume VII, Nomor 1, Mei 2019*, 25-32.
- Atmopawiro, A. (2006). *Pengkajian dan Analisis Tiga Algoritma Efisien Rabin-Karp, Knuth-morris-Pratt, dan Boyer-Moore dalam pencarian Pola dalam Suatu Teks*. Bandung: STEI ITB.
- Elchison, J. M. (2012). Cedarville University - CS3410-Algorithm. [Online]. Available: http://www.cedarville.edu/personal/personalpages/shomper/cs3410_web/resources/rabin_karp_matching.ppt.
- Firdaus, H. B. (2008). Deteksi Plagiat Dokumen Menggunakan Algoritma Rabin-Karp.
- Ibrahim, A. (2011). Pengembangan Sistem Informasi Monitoring Tugas Akhir Berbasis Short Message Service (SMS) Gateway di Fasilkom Unsri. *08-JUSI-Vol-1-No-2- _Pengembangan-Sistem-Informasi-Monitoring-TA-Berbasis-SMS-di-Fasilkom-Unsri*.
- Imam Nawawi, P. P. (2019). DETEKSI PLAGIARISME PADA DOKUMEN SKRIPSI BERDASARKAN TINGKAT KESAMAAN DENGAN MENGGUNAKAN METODE LONGEST COMMON SUBSEQUENCE . *JANAPATI*, 217-226.
- Neva Satyahadewi, N. M. (2019). SISTEM INFORMASI MONITORING TUGAS AKHIR (SIMTA) BERBASIS WEB FAKULTAS MIPA UNIVERSITAS TANJUNGPURA. *CESS (Journal of Computer Engineering System and Science)*, 83-87.
- Ngroho, E. (2011). *Perancangan Sistem Deteksi Plagiarism Dokumen Teks Dengan Menggunakan Algoritma Rabin-Karp Tugas Akhir*. Malang: Universitas Brawijaya Malang.
- Pratama, M. e. (2013). *Aplikasi Pendeteksi Dupliasi Dokumen Teks Bahasa Indonesia Menggunakan Algoritma Winnowing Dengan Metode K-Gram Dan Synonym Recognition*. Malang: Jurusan Teknik Informatika Universitas Muhammadiyah Malang.
- Satia Suhada, S. B. (2017). IMPLEMENTASI ALGORITMA RABIN KARP DAN STEMMING NAJIEF ANDRIANI UNTUK DETEKSI PLAGIARISME DOKUMEN. *SWABUMI*, 84-89.
- Wulansari, E. (2013). Perancangan Sistem Informasi Manajemen Tugas Akhir Berbasis WebJurusan Teknik Sipil Fakultas Teknik Universitas Tanjungpura. *Perancangan_Sistem_Informasi_Manajemen_T*.