IMPLEMENTASI ALGORITMA RABIN-KARP UNTUK MENENTUKAN KETERKAITAN ANTARPUBLIKASI PENELITIAN DOSEN TAHUN 2013

Handrie Noprisson, Boko Susilo, dan Ernawati

Program Studi Teknik Informatika, Fakultas Teknik, Universitas Bengkulu

ABSTRACT

Rabin-Karp algorithm is one kind of string matching algorithm that can be use to do multiple pattern search. To reduce the processing time and increase accurate, before calculating percentage of similarity using dice's similarity coefisient method from value of string matching of Rabin-Karp algorithm, text documents will pass 3 phases of preprocessing which are: case folding, filtering dan stemming using Nazief-Adriani algorithm. In application that have builded, text document is an abstract of publication of lecturer research in Lembaga Penelitian, University of Bengkulu. Result of recommendation of publication or journal according to value of similarity percentage from abstract of publication or journal that accesed by visitor. System development method that used to develop this application using waterfall model. In analysis and design phase using the structured approach. To test application we use feasibiltiy test and compare result of calculation of algorithm manually and using an application.

Keywords : *Text Mining, Website, Waterfall, Algoritma Rabin-Karp*

1. PENDAHULUAN

Kemudahan mendapatkan informasi menjadi salah satu harapan bagi pengunjung *website*. Fasilitas pencarian dalam sebuah *website* terkadang tidak cukup membantu untuk menemukan beberapa informasi yang diinginkan. Ketika seorang pengunjung telah mendapatkan informasi yang dicari dan ingin mencari informasi terkait lainnya, terkadang fasilitas pencarian tidak cukup membantu untuk mengatasi hal tersebut.

Sebagai contoh, pengunjung *website* ingin mencari publikasi jurnal yang akan digunakan sebagai bahan referensi penelitiannya. Ketika pengunjung telah menemukan sebuah publikasi jurnal sebagai referensi yang sesuai, seringkali pengunjung melakukan pencarian ulang untuk mendapatkan publikasi jurnal lain yang sejenis melalui fasilitas pencarian atau fasilitas kategorisasi yang dilakukan oleh *admin website*.

Namun, pada fasilitas kategorisasi rentan terjadi ketidaksesuaian antar label kategori dengan isi dari publikasi jurnal. Hal ini terjadi mungkin dikarenakan kesalahan input atau kesalahan dalam penafsiran isi dokumen teks oleh *admin*. Selain itu, kategorisasi akan menambah beban kerja *admin* yaitu bertambahnya kolom pengkategorian yang perlu diisi oleh *admin* dalam menginputkan data.

Oleh karena itu, diperlukan fasilitas tambahan untuk mempermudah pengunjung dalam menemukan dokumen teks publikasi jurnal yang saling memiliki keterkaitan, yaitu fasilitas rekomendasi dokumen teks yang berkaitan. Tingkat keterkaitan fasilitas ini ditentukan berdasarkan persentase kesamaan dari abstrak publikasi jurnal dengan menggunakan metode pencocokan *string*.

Masalah pencocokan *string* cukup menarik untuk diteliti, ada banyak algoritma yang dapat digunakan antara lain Boyer-Moore, Brute Force, Knuth-Morris-Pratt, Rabin-Karp, Smith-Waterman dan lain-lain. Algoritma-algoritma tersebut dapat diklasifikasikan menjadi algoritma untuk pencocokan string *single pattern* maupun pencocokan *string multiple pattern*. Pada pencocokan *string single pattern*, paket atau informasi yang dicari berdasarkan pola satu *string* saja. Sedangkan pada pencocokan string *multiple pattern,* paket atau informasi yang dicari berdasarkan beberapa susunan pola *string* [1].

Algoritma Rabin-Karp apabila digunakan pada pencocokan *single pattern* masih kurang mangkus dibandingkan dengan algoritma Knuth-Morris-Pratt (KMP) atau Boyer-Moore, karena kasus terburuknya.

Akan tetapi Rabin-Karp adalah sebuah algoritma yang tepat untuk pencocokan *multiple pattern*. Algoritma lainnya bisa memiliki kompleksitas O(n) untuk pencocokan *single pattern* dan kompleksitas O(nk) untuk pencocokan k *pattern*. Sebaliknya algoritma Rabin-Karp diatas bisa mencari k *pattern* dengan kompleksitas sebesar O(n+k) 2].

Lembaga Penelitian Universitas Bengkulu sebagai pelaksana, pengelola dan pelayanan sumber informasi tentang kegiatan penelitian dan pengembangan sektor Unggulan Provinsi Bengkulu memiliki strategi pencapaian misi dalam menyebarluaskan hasil penelitian. Salah satu bentuk kegiatan dalam pencapaian strategi tersebut adalah dibangunnya sebuah *website* lembaga penelitian yang mampu menyebarluaskan hasil penelitian secara mudah dan cepat. Lembaga penelitian telah membangun sebuah *website,* namun *website* hanya menampilkan informasi berupa berita dan agenda saja.

Berdasarkan latar belakang di atas, maka penulis tertarik untuk melakukan penelitian dan memilih judul "Implementasi Algoritma Rabin-Karp Untuk Menentukan Keterkaitan Antar Publikasi Penelitian Dosen Tahun 2013 (Studi Kasus: Website Lembaga Penelitian Universitas Bengkulu)". Penelitian ini diharapkan dapat membantu pengelolaan dan penyebaran hasil penelitian Lembaga Penelitian Universitas Bengkulu serta mempermudah pengunjung *website* untuk menemukan informasi penelitian yang diperlukan.

2. TINJAUAN PUSTAKA

2.1 Penelitian yang Relevan

Beberapa penelitian yang berhubungan dengan pendekatan pengukuran persentase kemiripan (*percentage similiarity*) dengan menggunakan metode pencocokan string *multiple-pattern*:

a. Pembuatan Sistem Deteksi Plagiarisme Dokumen Teks dengan Menggunakan Algoritma Rabin-Karp oleh Eko Nugroho, Program Studi Ilmu Komputer, Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Brawijaya Malang pada tahun 2011. Pada penelitian ini dilakukan modifikasi pada algoritma Rabin-Karp yaitu dengan menyisipkan metode *stemming* dengan menggunakan algoritma Arifin-Setiono pada tahap *preprocessing*-nya dan melakukan modifikasi pada saat proses *hashing* serta perubahan pada proses *string-matching*. [3]
b. Pembuatan Sistem Penilaian Otomatis Pada Jawaban Ujian Berbentuk Esai Menggunakan Metode Rabin Karp oleh David Indra Lesmana, Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang pada tahun 2012. Penelitian ini bertujuan untuk merancang dan membangun sistem penilaian otomatis pada jawaban ujian berbentuk esai menggunakan metode rabin karp sehingga dapat digunakan untuk membantu kinerja dosen dalam melakukan penilaian. [4]
c. Arsitektur Untuk Aplikasi Deteksi Kesamaan Dokumen Bahasa Indonesia oleh Anna Kurniawati, Kemal Ade Sekarwati, dan I Wayan Simri Wicaksana, Jurusan Sistem Informasi, Fakultas Ilmu komputer dan Teknologi Informasi Universitas Gunadarma pada tahun 2012. Penelitian pengukuran kesamaan dokumen Bahasa Indonesia yang ada hanya mengukur kesamaan kata ataupun kalimat, belum mempertimbangkan struktur kalimat, jumlah kalimat, posisi kalimat dan makna kata untuk membandingkan kalimat. Penelitian ini bertujuan untuk mengembangkan arsitektur aplikasi deteksi kesamaan dokumen teks Bahasa Indonesia dengan mempertimbangkan struktur kalimat, jumlah kalimat, posisi kalimat dan memperhitungkan faktor sinonim kata untuk melihat dari sisi makna kata. [5]

2.2 Landasan Teori

1. Text Mining

Text mining atau yang dikenal dengan *text data mining* atau penemuan informasi (*knowledge*) dari basis data tekstual, umumnya mengacu pada proses penggalian informasi atau pengetahuan berdasarkan sebuah pola tertentu dari dokumen teks yang tidak terstruktur.

Text mining memiliki tugas yang lebih kompleks karena melibatkan data teks yang sifatnya tidak terstruktur dan kabur (*fuzzy*). *Text mining* merupakan bidang multidisiplin yang melibatkan *information retrieval*, analisis teks, ekstraksi informasi, *clustering*, kategorisasi, visualisasi, teknologi basis data, *machine learning*, dan *data mining* [6].

Perbedaan mendasar antara *text mining* dan *data mining* terletak pada sumber data yang digunakan. Pada *data mining*, pola-pola diekstrak dari basis data yang terstruktur, sedangkan *text mining*, pola-pola diekstrak dari data tekstual (*natural language*). Secara umum, basis data didesain untuk program dengan tujuan melakukan pemrosesan secara otomatis, sedangkan teks ditulis untuk dibaca langsung oleh manusia [7].

2. Ekstraksi Dokumen

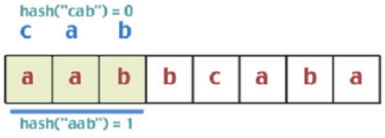
Teks yang akan dilakukan proses *text mining*, pada umumnya memiliki beberapa karakteristik diantaranya adalah memiliki dimensi yang tinggi, terdapat *noise* pada data, dan terdapat struktur teks yang tidak baik. Cara yang digunakan dalam mempelajari suatu data teks adalah dengan terlebih dahulu menentukan fitur-fitur yang mewakili setiap kata untuk setiap fitur yang ada pada dokumen. Sebelum menentukan fitur-fitur yang mewakili, diperlukan tahap *preprocessing* yang dilakukan secara umum dalam *text mining* pada dokumen, yaitu *case folding*, *tokenizing*, *filtering*, *stemming*, *tagging* dan *analyzing* [8].

3. Algoritma Rabin-Karp

Algoritma Rabin-Karp adalah algoritma pencocokan string yang menggunakan fungsi hash sebagai pembanding antara string yang dicari (m) dengan substring pada teks (n). Apabila *hash value* keduanya sama maka akan dilakukan perbandingan sekali lagi terhadap karakter-karakternya. Apabila hasil keduanya tidak sama, maka substring akan bergeser ke kanan. Pergeseran dilakukan sebanyak (n-m) kali. Perhitungan nilai *hash* yang efisien pada saat pergeseran akan mempengaruhi performa dari algoritma ini [9]

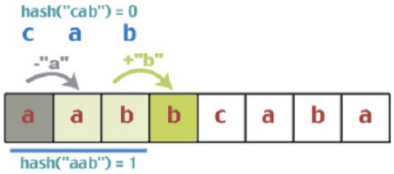
Berikut ini adalah ilustrasi cara kerja algoritma Rabin-Karp:

Diberikan masukan “cab” dan teks “aabbcaba”. Fungsi *hash* yang dipakai misalnya akan menambahkan nilai keterurutan setiap huruf dalam alfabet (a = 1, b = 2, dst.) dan melakukan modulo dengan 3. Didapatkan nilai hash “cab” adalah 0 dan tiga karakter pertama pada teks yaitu “aab” adalah 1.



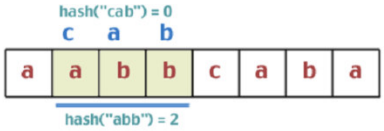
Gambar 1 *Fingerprint Awal* **[9]**

Hasil perbandingan ternyata tidak sama, maka *substring* pada teks akan begeser satu karakter ke kanan. Algoritma tidak menghitung kembali nilai *hash substring*. Disinilah dilakukan apa yang disebut *rolling hash* yaitu mengurangi nilai karakter yang keluar dan menambahkan nilai karakter yang masuk sehingga didapatkan kompleksitas waktu yang relatif konstan pada setiap kali pergeseran.



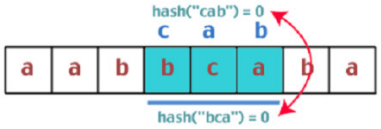
Gambar 2 Menggeser *fingerprint* [9]

Setelah pergeseran, didapatkan nilai hash dari fingerprint "aab" (abb=aab - a + b) menjadi dua (2 = 1 ± 1 + 2).



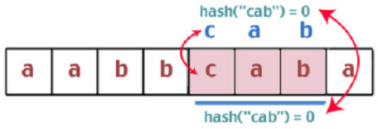
Gambar 3 Pembandingan kedua [9]

Hasil perbandingan juga tidak sama, maka dilakukan pergeseran. Begitu pula dengan perbandingan ketiga. Pada perbandingan keempat, didapatkan nilai *hash* yang sama.



Gambar 4 Perbandingan keempat (nilai *hash* sama) [9]

Karena nilai hash sama, maka dilakukan perbandingan string karakter per karakter antara "bca" dan "cab". Didapatkan hasil bahwa kedua *string* tidak sama. Kembali *substring* bergeser ke kanan.



Gambar 5 Perbandingan kelima (string ditemukan) [9]

Pada perbandingan yang kelima, kedua nilai hash dan karakter pembentuk string sesuai, sehingga solusi ditemukan. Dari hasil perhitungan, kompleksitas waktu yang dibutuhkan adalah O(m+n) dengan m adalah panjang *string* masukan dan n adalah jumlah looping yang dilakukan untuk menemukan solusi. Hasil ini jauh lebih mangkus daripada kompleksitas waktu yang didapat menggunakan algoritma *brute-force* yaitu O(mn).

Algoritma Rabin-Karp diciptakan oleh Michael O. Rabin dan Richard. Karp pada tahun 1987 yang menggunakan fungsi *hashing* untuk menemukan pattern di dalam *string* teks. Karakteristik Algoritma Rabin-Karp [10]:

1. Menggunakan sebuah fungsi *hashing*
2. Fase *prepocessing* menggunakan kompleksitas waktu O(m)
3. Untuk fase pencarian kompleksitasnya : O(mn)
4. Waktu yang diperlukan O(n+m)

Fungsi *hashing* menyediakan metode sederhana untuk menghindari perbandingan jumlah karakter yang quadratik di dalam banyak kasus atau situasi. Daripada melakukan pemeriksaan terhadap setiap posisi dari teks ketika terjadi pencocokan pola, akan lebih baik efisien untuk melakukan pemeriksaan

hanya jika teks yang sedang proses memiliki kemiripan seperti pada *pattern*. Untuk melakukan pengecekan kemiripan antara dua kata ini digunakan fungsi *hash* [10].

Fungsi *hash* yang digunakan biasanya modulo berbasis bilangan prima besar. Alasan dipilih bilangan prima yang cukup besar adalah untuk mengurangi kemungkinan dua buah *corresponding number value* yang sama [11].

4. Hashing

Hashing adalah suatu cara untuk mentransformasi sebuah string menjadi suatu nilai yang unik dengan panjang tertentu (*fixed-length*) yang berfungsi sebagai penanda *string* tersebut. Fungsi untuk menghasilkan nilai ini disebut fungsi hash, sedangkan nilai yang dihasilkan disebut nilai *hash*. Contoh sederhana *hashing* adalah:

Firdaus, Hari; Munir, Rinaldi
Rabin, Michael; Karp, Richard
menjadi :
7864 = Firdaus, Hari; 9802 = Munir, Rinaldi
1990 = Rabin, Michael; 8822 = Karp, Richard

Nilai *hash* pada umumnya digambarkan sebagai *fingerprint* yaitu suatu string pendek yang terdiri atas huruf dan angka yang terlihat acak (data biner yang ditulis dalam heksadesimal) [9].

Algoritma Rabin-Karp didasarkan pada fakta jika dua buah string sama maka harga *hash value*-nya pasti sama. Akan tetapi ada dua masalah yang timbul dari hal ini, masalah pertama yaitu ada begitu banyak string yang berbeda, permasalahan ini dapat dipecahkan dengan meng-*assign* beberapa *string* dengan *hash value* yang sama. Masalah yang kedua belum tentu *string* yang mempunyai *hash value* yang sama cocok untuk mengatasinya, maka untuk setiap string yang di-*assign* dilakukan pencocokan string secara Brute-Force. Kunci agar algoritma Rabin-Karp efisien, terdapat pada pemilihan *hash value*-nya. Salah satu cara yang terkenal dan efektif adalah memperlakukan setiap substring sebagai suatu bilangan dengan basis tertentu [9].

5. K-grams

K-grams 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-grams ini digunakan untuk mengambil potongan-potongan karakter huruf sejumlah k dari sebuah kata yang secara kontinuitas dibaca dari teks sumber hingga akhir dari dokumen.

Berikut ini adalah contoh k-grams dengan k=5:

Text: A do run run run, a do run run

Kemudian dilakukan penghilangan spasi:

adorunrunrunadorunrun

Sehingga dihasilkan rangkaian 5-grams yang diturunkan dari text: adoru dorun orunr runru unrun nrunr runru unrun nruna runad unado nador adoru dorun orunr runru unrun [12]

3. METODE PENELITIAN

3.1 Jenis Penelitian

Jenis penelitian terapan adalah satu jenis penelitian yang hasilnya dapat secara langsung diterapkan untuk memecahkan permasalahan yang dihadapi. Penelitian ini menguji manfaat dari teori-teori ilmiah serta mengetahui hubungan empiris dan analisis dalam bidang-bidang tertentu. Penelitian terapan lebih difokuskan pada pengetahuan teoretis dan praktis dalam bidang-bidang tertentu bukan pengetahuan yang bersifat universal misalnya bidang kedokteran, pendidikan, atau teknologi [13].

Tujuan Penelitian terapan ini adalah untuk merancang dan membangun *website* sebagai media untuk menyimpan dan menginformasikan hasil penelitian di Lembaga Penelitian Universitas Bengkulu yang

dilengkapi dengan fasilitas rekomendasi dokumen teks terkait lain dengan memanfaatkan algoritma Rabin- Karp.

3.2 Gambaran Umum Lokasi Penelitian

Lokasi penelitian yang menjadi objek studi kasus pada skripsi adalah Lembaga Penelitian Universitas Bengkulu. Lembaga Penelitian Universitas Bengkulu merupakan unit kerja yang mengkoordinasikan semua kegiatan penelitian yang dilaksanaan oleh staf pengajar di lingkungan Universitas Bengkulu. Kegiatan penelitian tersebut meliputi penelitian kerjasama dengan instansi pemerintah maupun swasta, serta penelitian yang memanfaatkan pendanaan yang bersifat kompetitif dari Universitas Bengkulu (DIPA), dana Dikti, BPPT, LIPI dan sebagainya.

3.3 Teknik Pengumpulan Data

Dalam mengumpulkan data, teknik pengumpulan data yang digunakan dalam penelitian ini yaitu:

1) Wawancara
   Wawancara dilakukan terhadap individu yang mengetahui tentang data dan informasi masalah yang dibahas dalam penelitian ini. Topik wawancara mengenai jenis data lembaga penelitian yang akan ditampilkan pada *website*, desain antarmuka *website*, spesifikasi fungsi, kemampuan serta fasilitas dari *website*.
2) Studi Dokumentasi
   Analisis dokumen dilakukan untuk mengumpulkan data yang bersumber dari arsip dan dokumen yang berhubungan dengan penelitian. Dokumen yang diperlukan berupa data dosen, data penelitian dan data publikasi di Lembaga Penelitian Universitas Bengkulu.

3.4 Jenis dan Sumber Data

Jenis data yang dibutuhkan dan digunakan dalam penelitian ini berasal dari 2 (dua) jenis data yaitu:

1) Data Primer
   Data primer adalah data yang diperoleh langsung dari dari responden melalui wawancara. Data primer pada penelitian ini terdiri dari:
   a) Jenis data masukkan dan keluaran yang akan ditampilkan di *website*.
   b) Spesifikasi fungsi, kemampuan serta fasilitas dari *website* yang akan dibangun.
2) Data Sekunder
   Data sekunder adalah data yang sudah jadi atau dipublikasikan untuk umum oleh instansi atau lembaga yang mengumpulkan, mengolah dan menyajikan. Data sekunder berupa data-data deskripsi maupun abstrak hasil penelitian yang merupakan hasil studi dokumentasi di Lembaga Penelitian Universitas Bengkulu.

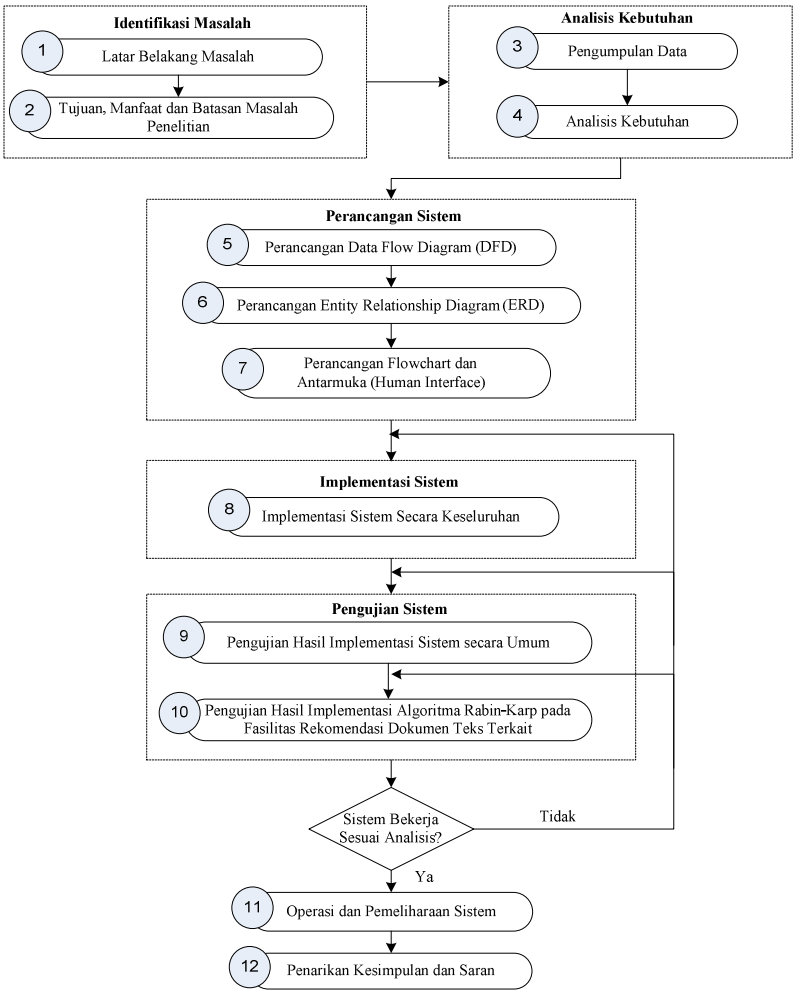
3.5 Metode Pengembangan Sistem

Metode pengembangan system *website* menggunakan model air terjun (*waterfall*). Tahap-tahap utama dari model air terjun adalah memetakan kegiatan-kegiatan pengembangan dasar, antara lain:

1) Analisis dan definisi persyaratan.
2) Perancangan sistem dan perangkat lunak. Tahapan meliputi pembuatan Data Flow Diagram (DFD), Entity Relationship Diagram (ERD) dan diagram alir (*flowchart*).
3) Implementasi dan pengujian unit. Bahasa pemrograman yang digunakan yaitu adalah PHP dan memanfaatkan *database* MySQL pada sisi *server*.
4) Integrasi dan pengujian sistem.
5) Operasi dan pemeliharaan. Sistem diterapkan dan digunakan oleh pengguna. Pemeliharaan mencakup koreksi dari berbagai *error* yang tidak ditemukan pada tahap-tahap terdahulu.

3.6 Diagram Alir Penelitian

Penelitian dilakukan berdasarkan diagram alir dibawah ini, hal ini juga disesuaikan dengan metode pengembangan sistem yang dipilih. Adapun diagram alir pada penelitian ini dapat digambarkan sebagai berikut:



Gambar 2 Diagram Alir Penelitian

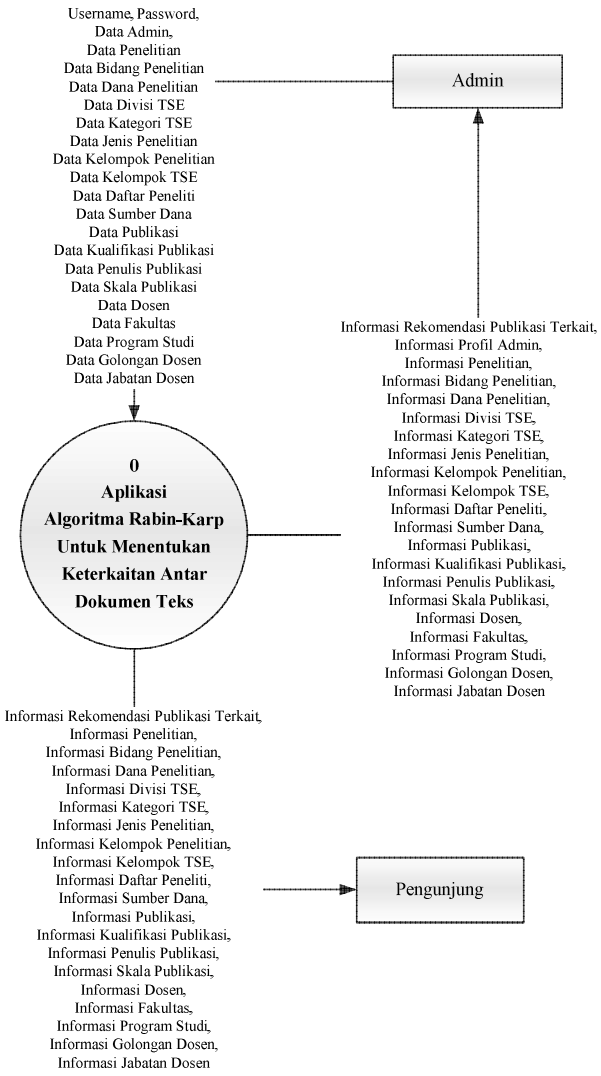
4. HASIL PENELITIAN DAN PEMBAHASAN

4.1 Analisis dan Perancangan Sistem

Analisis dan perancangan sistem adalah tahap penting yang menjadi proses awal untuk memahami sistem yang akan dibuat. Hasil akhir dari tahap ini nantinya adalah berupa sistem komputerisasi sesuai dengan kebutuhan pengguna.

1. Diagram Konteks Sistem/Website

Untuk menggambarkan alur data yang ada pada sistem secara umum digunakan notasi Data Flow Diagram (DFD).

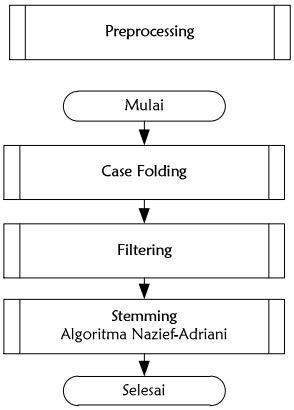


Gambar 3 Perancangan Diagram Konteks

2. Perancangan Flowchart Algoritma

a. Perancangan Flowchart Preprocessing

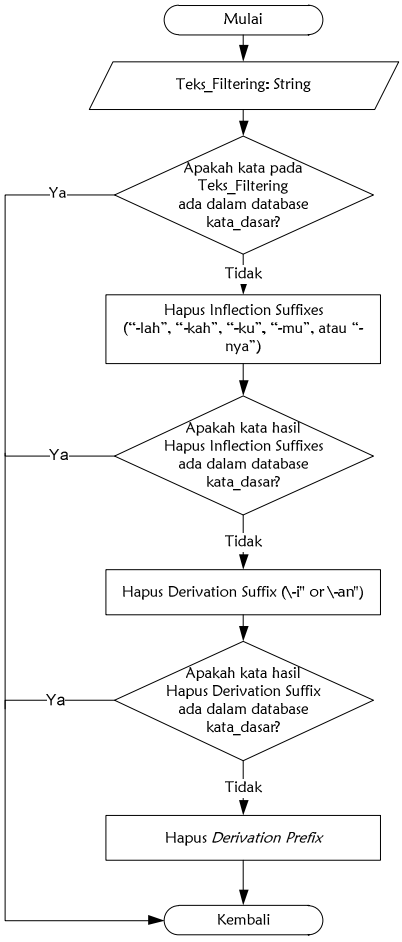
Tahap *preprocessing* cukup penting untuk memaksimalkan kinerja Algoritma Rabin-Karp.



Gambar 4 *Flowchart* Preprocessing

b. *Flowchart Stemming Nazief-Adriani*

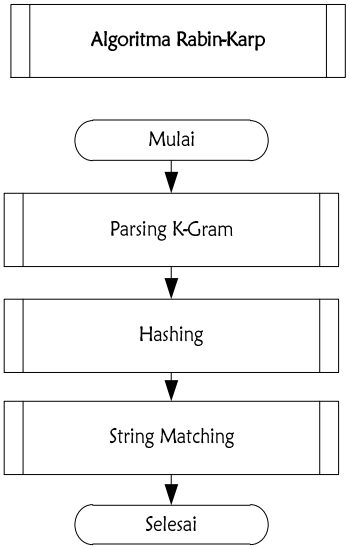
Algoritma Nazief dan Mirna Adriani ini mengelompokkan bentuk imbuhan menjadi *Inflection Suffixes* (“-lah”, “-kah”, “-ku”, “-mu”, atau “-nya”), *Derivation Suffixes* (“-i”, “-an” atau “-kan”), dan *Derivation Prefix*. Efisiensi dari algoritma ini tergantung pada banyaknya kata dasar yang ada dalam *database*.



Gambar 5 *Flowchart* Stemming Nazief-Adriani

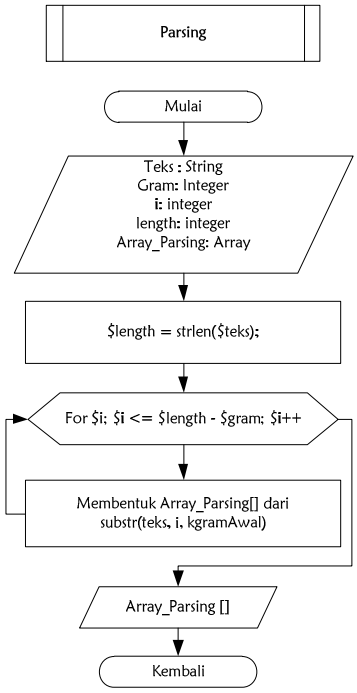
c. *Flowchart Algoritma Rabin Karp*

Flowchart Algoritma Rabin-Karp digunakan utuk menggambarkan tahap dari proses Algoritma Rabin-Karp dalam membandingkan kedua dokumen teks yang berbeda.



Gambar 6 *Flowchart* Algoritma Rabin-Karp

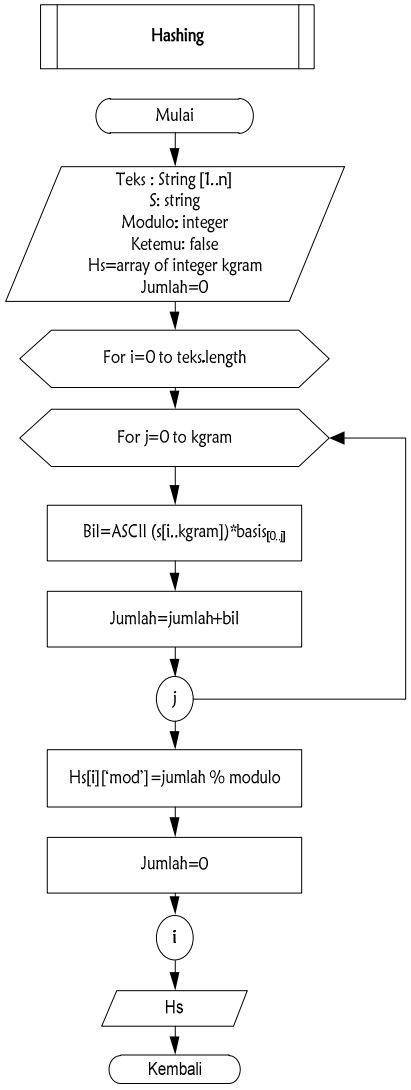
d. *Flowchart Parsing dengan Metode K-Gram*



Gambar 7 *Flowchart Parsing* dengan Metode K-Gram

e. Flowchart Hashing

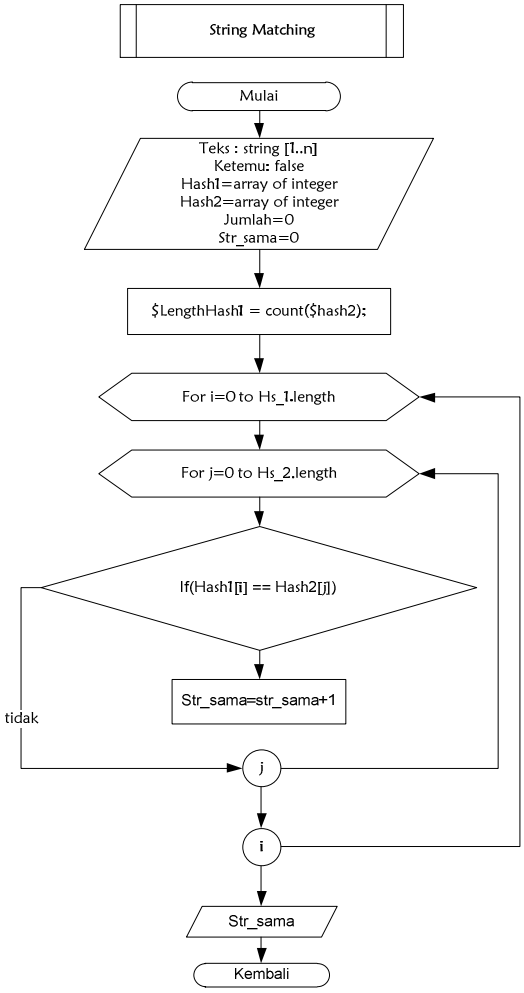
Dalam aplikasi ini nilai *hashing* didapatkan dari modulo dari hasil perkalian bilangan basis dengan bilangan ASCII.



Gambar 8 *Flowchart Hashing*

f. Flowchart String Matching

String matching adalah proses membandingkan *string* dari dokumen teks yang berbeda yang mempunyai *hash value* yang sama cocok. Jumlah *string* yang sama akan dihitung dan untuk perhitungan persentasenya menggunakan metode Dice's Similarity Coefisient.



Gambar 9 *Flowchart String Matching*

4.2 Pembahasan

Pada saat administrator menambahkan atau mengubah data publikasi, *preprocessing* algoritma Rabin-Karp meliputi *case folding*, *filtering* dan *stemming* telah diimplementasikan pada proses ini pada atribut atau kolom abstrak. Sebagai contoh, diinputkan abstrak dari penelitian yang berjudul "Model Pendidikan Kecakapan Hidup Bagi Remaja Miskin Putus Sekolah Dalam Usaha Hidup Mandiri Melalui Pelatihan Kewirausahaan Kerajinan Pembuatan Ikan Asin" yang diketuai oleh Dr. Osa Juarsa seperti pada Gambar 17 dan kemudian abstrak tersebut dilakukan proses *case folding*, *filtering* dan *stemming* tersebut.

Model Pendidikan Kecakapan Hidup Bagi Remaja Miskin Putus Sekolah Dalam Usaha Hidup Mandiri Melalui Pelatihan Kewirausahaan Kerajinan Pembuatan Ikan Asin   Pendidikan   Desa Pekik Nyaring, Kecamatan Pondok Kelapa, Kabupaten Benteng, banyak ditemukan remaja miskin putus sekolah. Berdasarkan survei awal, menunjukkan 42,4% lulusan dari SLTA, 34,4% dari lulusan SLTP dan 3,6% dari lulusan SD tidak melanjutkan pendidikan ke jenjang yang lebih tinggi. Mereka perlu mendapat perhatian agar tidak menjadi beban masyarakat. Oleh karena itu perlu direncanakan model pendidikan yang dapat mengubah manusia beban menjadi manusia produktif. Tujuan penelitian ini adalah untuk meningkatkan pengetahuan, keterampilan dan model pendidikan kecakapan hidup kepada remaja putus sekolah agar mampu hidup mandiri. Penelitian terdiri dari tiga variabel: (1) Intensitas Pelayanan model; (2) Keterampilan kerja remaja miskin putus sekolah; (3) Efektivitas pelayanan model.
.....

Gambar 10 Contoh Abstrak Publikasi

Proses *case folding* yang dilakukan pada dokumen teks abstrak akan mentransformasikan seluruh abjad menjadi seluruh huruf kecil dengan menggunakan kode strtolower dan menghapus beberapa karakter yang tidak penting (*delimiter*), seperti *tag* HTML dan PHP dengam menggunakan kode strip_tags, menghilangkan angka dan simbol lainnya dengan kode preg_replace dengan menuliskan *string* yang akan dihapus dalam bentuk ekspresi regular dan terakhir menggunakan kode str_replace dengan menuliskan *delimiter* lain yang mungkin tidak terhapus oleh kode preg_replace maupun str_replace.

model pendidikan kecakapan hidup bagi remaja miskin putus sekolah dalam usaha hidup mandiri melalui pelatihan kewirausahaan kerajinan pembuatan ikan asin pendidikan desa pekik nyaring kecamatan pondok kelapa kabupaten benteng banyak ditemukan remaja miskin putus sekolah berdasarkan survei awal menunjukkan lulusan dari slta dari .....

Gambar 11 Contoh Abstrak Publikasi Hasil *Case Folding*

Proses *filtering* dilakukan dengan menggunakan bantuan kode array_intersect, array_diff dan implode. *Stoplist* atau *stopword* dari *database* akan digunakan di-*filter* dengan menggunakan array_intersect yang kemudian disimpan dalam *array* dengan nama $stopList sedangkan untuk menghapus *stopword* dari abstrak jurnal digunakan kode array_diff yang kemudian disimpan dalam *array* dengan nama $hapusstopList dan terakhir seluruh isi *array* hasil dari proses filtering ini digabungkan dengan menggunakan kode implode.

Setelah dokumen abstrak publikasi hasil proses *case* folding melalui proses *filtering*, maka hasil dari proses *filtering* dapat dilihat pada Gambar 19.

model pendidikan kecakapan hidup remaja miskin putus sekolah usaha hidup mandiri pelatihan kewirausahaan kerajinan pembuatan ikan asin pendidikan desa pekik nyaring kecamatan pondok kelapa kabupaten benteng ditemukan remaja miskin putus sekolah berdasarkan survei lulusan slta lulusan sltp lulusan sd pendidikan jenjang tinggi perhatian beban .....

Gambar 12 Contoh Abstrak Publikasi Hasil *Case Folding* dan *Filtering*

Proses *stemming* merupakan proses lanjutan dari proses *case folding* dan proses *filtering* yang digunakan untuk mentransformasikan kata berimbuhan menjadi kata dasar. Bahasa Indonesia cukup rumit dalam hal tata penulisan dengan adanya banyak aturan dalam penulisan imbuhan meliputi afiks-prefiks (imbuhan awal/awalan), afiks infiks (imbuhan tengah/sisipan), afiks-sufiks (imbuhan akhir/akhiran), konfiks imbuhan awal dan akhir yang sama-sama membentuk satu arti dan afiks gabung (imbuhan awal dan akhir, dan tiap-tiap imbuhan masih tetap).

Kekurangan dari algoritma Nazief-Adriani adalah belum mampunya untuk mendeteksi dan menghapus afiks-infiks atau imbuhan tengah/sisipan. Misalnya pada kata "pemetaan" akan ditransformasi menjadi "meta" dikarenakan imbuhan pada kata tersebut dikenali sebagai imbuhan "pe-an", sehingga untuk mengatasinya tabel baru yang berisi imbuhan afiks-infiks dan kata dasarnya yang kemudian diproses dengan menggunakan kode array_intersect, array_diff dan implode.

Setelah proses *stemming* seperti yang telah dijelaskan diatas dilakukan terhadap dokumen abstrak publikasi hasil proses sebelumnya (lihat Gambar 19), maka hasil dari proses *case folding* dapat dilihat pada Gambar 20.

model didik cakap hidup remaja miskin putus sekolah usaha hidup mandiri latih kewirausahaan rajin buat ikan asin didik desa pekik nyaring camat pondok kelapa kabupaten benteng temu remaja miskin putus sekolah dasar survei lulus slta lulus sltp lulus sd didik jenjang tinggi hati beban masyarakat rencana model didik ubah manusia beban manusia ....

Gambar 13 Contoh Abstrak Publikasi Hasil *Case Folding, Filtering* dan *Stemming*

Hasil dari ketiga tahap dari *preprocessing* disimpan kedalam *field* atau kolom hasil_prepro pada tabel data_publikasi yang ada dalam *database.* Output dari manajemen data penelitian ini juga ditampilkan pada halaman pengunjung berupa tabel yang dilengkapi dengan fitur pencarian berdasarkan dengan memilih jenis kategori pencarian yang tersedia di *combobox* dengan memasukkan kata kunci yang akan dicari.

Gambar 14 Antarmuka Tabel Data Publikasi pada Halaman Pengunjung

Pada antarmuka detail data publikasi diimplementasikan algoritma Rabin-Karp berupa fasilitas rekomendasi publikasi jurnal terkait. Implementasi algoritma Rabin-Karp dimulai dengan melakukan proses *preprocessing* meliputi, proses *tokenizing*, *filtering* dan *stemming* yang dilakukan pada halaman administrator. Setelah semua *preprocessing* selesai kemudian dilakukan proses algoritma Rabin-Karp yang terdiri dari tahap *parsing* dengan metode K-gram, tahap *hashing* dan terakhir tahap *string matching*. Hasil luaran dari proses algoritma Rabin-Karp akan dihitung kembali dengan menggunakan metode Dice's similiarity untuk mendapatkan persentase kesamaan yang digunakan sebagai parameter untuk mengetahui keterkaitan antar dokumen teks tersebut.

Proses *parsing* seperti yang telah dijelaskan diatas dilakukan terhadap dokumen abstrak publikasi hasil proses sebelumnya (lihat Gambar 20), maka hasil dari proses *parsing* ini dapat dilihat pada Gambar 22.

mod ode del eld ldi did idi dik ikc kca cak aka kap aph phi hid idu dup upr pre rem ema maj aja jam ami mis isk ski kin inp npu put utu tus uss sse sek eko kol ola lah ahu hus usa sah aha hah ahi hid idu dup upm pma man and ndi dir iri ril ila lat ati tih ihk hke kew ewi wir ira rau aus usa sah aha haa aan anr nra raj aji jin inb nbu bua uat ati tik ika kan ana nas .....

Gambar 15 Contoh Abstrak Publikasi Hasil *Parsing* dengan K-Gram = 3

Proses *hashing* merupakan proses lanjutan dari proses *parsing* sebelumnya. Proses *hashing* dilakukan dengan menggunakan bantuan kode strlen, ord dan pow. Nilai basis dan modulo diiniasi terlebih dahulu menjadi $basis dan $modulo seperti yang ditunjukkan pada baris kode 1 dan 2, kemudian hitung panjang *string* yang akan dihitung *hash value*-nya bantuan kode strlen. Panjang *string* perlu dihitung untuk menentukan deret pangkat dari basis yang dipilih, yang nantinya akan digunakan pada kode pow (*number $base, number $exp*) seperti yang terlihat pada baris kode 7 dan 8.

Kemudian dilakukan konversi setiap abjad pada *string* menjadi kode ASCII seperti yang terlihat pada baris kode 6 yang diikuti dengan mengalikan nilai ASCII dari abjad-abjad tersebut dengan nilai basis berpangkat dan modulo yang telah ditentukan sebelumnya.

Jika dihitung secara manual perhitungan deret pangkat dilakukan mengalikan basis dengan nilai ASCII *string*. Misalnya akan dilakukan perhitungan deret pangkat dari *string* "un", dimana nilai ASCII huruf u adalah 117 dan "n" adalah 110, maka bentuk perhitunganya, $(117*10^1)+(110*10^0)=1280$. Setelah melalui proses *hashing* seperti yang telah dijelaskan diatas terhadap dokumen abstrak publikasi hasil proses *parsing* (lihat Gambar 22), maka hasil dari proses *hashing* dapat dilihat pada Gambar 13.

6346 7413 17591 18441 5389 17687 3361 17694 3537 4679 16810 15708 4642 15845 8197 20391 3373 18011 11794 8453 9475 18464 5988 15682 3963 15768 6205 3753 10303 4848 3628 7065 8546 11901 11249 11873 10507 10149 18426 5002 7617 5310 15650 20718 11855 10042 15630 20187 15638 20391 3373 18011 11789 8319 5992 15789 6741 17701 3725 9578 3561 .....

Gambar 16 Contoh Hasil *Hashing* dari Hasil *Parsing* Abstrak Publikasi dengan K-Gram = 3

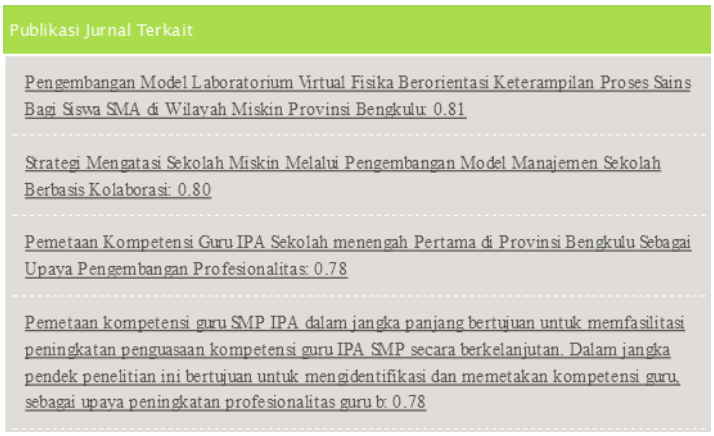
Hasil *hashing* yang sama antara kedua teks akan dihitung pada saat *string matching*, kemudian untuk mengukur persentase kesamaanya menggunakan metode Dice's Similarity Coefisient. *Hash value* yang sama dari dua dokumen teks abstrak yang berbeda dihitung dan jumlahnya disimpan dalam bentuk array. Proses kalkulasi *hash value* dilakukan dengan bantuan kondisi *for* yang akan berhenti jika seluruh komponen *array hash value* telah selesai dibandingkan.

Jumlah *hash value* dari kedua teks abstrak yang sama dilakukan perhitungan persamaan dengan "Dice's Similarity Coefisient" dengan rumus sebagai berikut [14]:

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

Keterangan:
S = nilai *similarity* antar kedua teks
A = jumlah dari kumpulan K-grams dalam teks 1
B = jumlah dari kumpulan K-grams dalam teks 2
C = jumlah dari K-grams yang sama dari teks yang dibandingkan.

Hasil dari ketiga tahap dari algoritma Rabin-Karp ditampilkan pada halaman pengunjung berupa tabel yang berisi publikasi jurnal yang memiliki kesamaan *string* terbanyak dengan dokumen teks abstrak yang sedang diakses oleh pengunjung seperti yang terlihat pada Gambar 24.



Gambar 17 Antarmuka Tabel Rekomendasi Publikasi Jurnal Terkait

Pada penelitian ini, pengujian hasil rekomendasi dilakukan dengan menggunakan data publikasi penelitian dosen tahun 2013. Pengujian dilakukan untuk menguji tujuan penelitian yaitu membuat sebuah fasilitas yang dapat menampilkan lima rekomendasi publikasi terkait yang sedang diakses oleh pengunjung *website* yang memiliki hasil perhitungan persentase kesamaan *string* dengan menggunakan metode Dice's Simlarity Coefficient diatas 40%.

Tabel 1 Pengujian Hasil Rekomendasi dengan Menggunakan Data Publikasi Penelitian Dosen Tahun 2013

No	Judul Publikasi yang Diakses Pengunjung	Rekomendasi Jurnal	
		Judul jurnal yang direkomendasikan	Nilai hasil perhitungan Dice's Similarity Coefisient (dalam %)
1	Strategi Mengatasi Sekolah Miskin Melalui Pengembangan Model Manajemen Sekolah Berbasis Kolaborasi	Optimalisasi Potensi Lokal Desa Rawan Bahaya Tsunami Dalam Rangka Mitigasi Menuju Terwujudnya Desa Siaga Bencana Mandiri di Pesisir Provinsi Bengkulu	51%
		Pemetaan Kompetensi Guru IPA Sekolah menengah Pertama di Provinsi Bengkulu Sebagai Upaya	47%

		Pengembangan Profesionalitas	
		Mikro-zonasi Tingkat Potensi Resiko Bencana Gempa Bumi di Wilayah Pesisir Provinsi Bengkulu untuk Mendukung Mitigasi Bencana	47%
		Pengembangan Model Laboratorium Virtual Fisika Berorientasi Keterampilan Proses Sains Bagi Siswa SMA di Wilayah Miskin Provinsi Bengkulu	47%
2	Perancangan Sistem Informasi Kebencanaan Tsunami Melalui Penyusunan Peta Kerawanan Dan Jalur Evakuasi Bencana Di Pesisir Kota Bengkulu	Optimalisasi Potensi Lokal Desa Rawan Bahaya Tsunami Dalam Rangka Mitigasi Menuju Terwujudnya Desa Siaga Bencana Mandiri di Pesisir Provinsi Bengkulu	47%
		Pengembangan Model Laboratorium Virtual Fisika Berorientasi Keterampilan Proses Sains Bagi Siswa SMA di Wilayah Miskin Provinsi Bengkulu	43%
		Pemetaan Kompetensi Guru IPA Sekolah menengah Pertama di Provinsi Bengkulu Sebagai Upaya Pengembangan Profesionalitas	42%
3	Pengembangan Model Quantum Teaching Dalam Pembelajaran Matematika Menggunakan Realistic Mathematics Education Untuk Meningkatkan Prestasi Belajar, Kreativitas, dan Karakter Siswa SD	Pemetaan Kompetensi Guru IPA Sekolah menengah Pertama di Provinsi Bengkulu Sebagai Upaya Pengembangan Profesionalitas	50%
		Pengembangan Model Laboratorium Virtual Fisika Berorientasi Keterampilan Proses Sains Bagi Siswa SMA di Wilayah Miskin Provinsi Bengkulu	49%
		Strategi Mengatasi Sekolah Miskin Melalui Pengembangan Model Manajemen Sekolah Berbasis Kolaborasi	45%
		Mikro-zonasi Tingkat Potensi Resiko Bencana Gempa Bumi di Wilayah Pesisir Provinsi Bengkulu untuk Mendukung Mitigasi Bencana	42%
4	Pemurnian Dan Perbaikan Sifat Plasma Nutfah Padi Lokal Bengkulu Untuk Adaptivitas Terhadap Wilayah Pesisir	Mikro-zonasi Tingkat Potensi Resiko Bencana Gempa Bumi di Wilayah Pesisir Provinsi Bengkulu untuk Mendukung Mitigasi Bencana	45%
		Optimalisasi Potensi Lokal Desa Rawan Bahaya Tsunami Dalam Rangka Mitigasi Menuju Terwujudnya Desa Siaga Bencana Mandiri di Pesisir Provinsi Bengkulu	44%
		Pemetaan Kompetensi Guru IPA Sekolah menengah Pertama di Provinsi Bengkulu Sebagai Upaya Pengembangan Profesionalitas	41%
5	Pemetaan Kompetensi Guru Bimbingan Dan Konseling Di Propinsi Bengkulu	Pemetaan Kompetensi Guru IPA Sekolah menengah Pertama di Provinsi Bengkulu Sebagai Upaya Pengembangan Profesionalitas	61%

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan analisa perancangan sistem, implementasi dan pengujian sistem, maka dapat disimpulkan bahwa:

1. Penelitian ini menghasilkan *website* lembaga penelitian yang telah dilengkapi fasilitas rekomendasi dokumen teks terkait yang sedang diakses oleh pengunjung yang dibangun dengan menggunakan bahasa pemograman *Hypertext Markup Language* (HTML), *Hypertext Preprosessor* (PHP), *Cascading Style Sheet* (CSS), *Javascript* dan *Structured Query Language* (SQL) dengan berbantuan editor Notepad++ serta MySQL sebagai *database server.*
2. Algoritma Rabin-Karp dapat digunakan dalam memberikan rekomendasi dokumen teks (publikasi jurnal) terkait yang memiliki kesamaan kata-kata secara otomatis dengan hasil rekomendasi berupa maksimal 5 dokumen teks (publikasi jurnal) yang memiliki nilai rekomendasi tertinggi berdasarkan perhitungan dengan metode Dice's Similarity Coefisient dari hasil proses string matching Algoritma Rabin Karp.
3. Dalam pemilihan nilai K-gram dan basis, untuk k-gram=2, nilai basis dan modulo yang menghasilkan nilai/angka unik 676 kombinasi *string* yang ada adalah 26 dan 677, sedangkan untuk k-gram=3, nilai basis dan modulo yang menghasilkan nilai/angka unik 17576 kombinasi *string* yang ada adalah 26 dan 17581.

5.2 Saran

Pelaksanaan peneltian ini hanya terbatas pada data-data dosen, penelitian dan publikasi yang telah diperoleh sebelumnya. Oleh karena itu, peneliti menyarankan agar aplikasi ini dapat terus dikembangkan lebih lanjut baik dalam hal *atribut database*, hal ini dikarenakan pada aplikasi yang dibangun berdasarkan analisis 2012-2013 sedangkan *atribut database* untuk data-data penelitian terus mengalami perubahan setiap tahunnya. Dengan adanya saran ini, diharapkan agar aplikasi web ini yang akan dibangun selanjutnya bisa lebih baik lagi.

Sedangkan untuk peningkatan kinerja algoritma dapat menggunakan satu algoritma lain pada proses *preprocessing* untuk menentukan sinonim kata antara kedua teks yang dibandingkan, artinya ketika ada dua kata yang maknanya sama dapat terhitung sebagai kata yang sama. Selain itu, pada implementasi Algoritma Nazief Adriani (pada *preprocessing*) masih memiliki banyak kesalahan pada saat melakukan transformasi kata berimbuhan menjadi kata dasar.

PERNYATAAN ORIGINALITAS

"Saya menyatakan dan bertanggung jawab dengan sebenarnya bahwa Artikel ini adalah hasil karya saya sendiri kecuali cuplikan dan ringkasan yang masing-masing telah saya jelaskan sumbernya".
[Handrie Noprisson - NPM. G1A009018]

DAFTAR PUSTAKA

[1] Qutaiba Ibrahim and Sahar Lazim, "Applying an Efficient Searching Algorithm for Intrusion Detection on Ubicom Network Processor," , 2011.

[2] Najib Baedlowi and Deka Aditia Adam, "String Matching dengan Menggunakan Algoritma Rabin Karp," , Bandung, 2006.

[3] Eko Nugroho, "Pembuatan Sistem Deteksi Plagiarisme Dokumen Teks dengan Menggunakan Algoritma Rabin-Karp ," Program Studi Ilmu Komputer, Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Brawijaya Malang , Malang, 2011.

[4] David Indra Lesmana, "Pembuatan Sistem Penilaian Otomatis Pada Jawaban Ujian Berbentuk Esai Menggunakan Metode Rabin Karp," Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim, Malang, 2012.

[5] Anna Kurniawati, Kemal Ade Sekarwati, and I Wayan Simri Wicaksana, "Arsitektur Untuk Aplikasi Deteksi Kesamaan Dokumen Bahasa Indonesia," Jurusan Sistem Informasi, Fakultas Ilmu komputer dan Teknologi Informasi Universitas Gunadarma , 2012.

[6] Ah-Hwee Tan, "Text Mining: The State of The Art and The Challenges," 21 Heng Mui Keng Terrace, 1999.

[7] Marti Hearst. (2003, Oktober) What Is Text Mining? [Online]. HYPERLINK "http://people.ischool.berkeley.edu/~hearst/text-mining.html" http://people.ischool.berkeley.edu/~hearst/text-mining.html

[8] Chandra Triawati. (2009) Metode Pembobotan Statistical Concept Based untuk Klastering dan Kategorisasi Dokumen Berbahasa Indonesia.

[9] Hari Bagus Firdaus. (2008) Deteksi Plagiat Dokumen Menggunakan Algoritma Rabin-Karp.

[10] Hary Fernando. (2009) Perbandingan dan Penguj ian Beberapa Algoritma Pencocokan String.

[11] Nicolas. Christopher. Saloko, Hadi Andres. (2008) Penelaahan Algoritma Rabin-Karp dan Perbandingan Prosesnya dengan Algoritma Knut-Morris-Path.

[12] Saul Schleimer, Daniel S. Wilkerson, and Aiken Alex, "Winnowing: Local Algorithms for Document Fingerprinting," in *SIGMOD*, San Diego, CA, 2003, p. 2003.

[13] Tim Dosen LPTK dan Widya Iswara. (2008, Juni) Pendekatan, Jenis dan Metode Penelitian Pendidikan.

[14] Serhiy Kosinov, "Evaluation Of N-Grams Conflation Approach In Text-Based Information Retrieval," in *Proceedings of International Workshop on Information Retrieval*, Edmonton, Alberta, Canada, 2001, pp. 136-142.