

IMPLEMENTASI ALGORITMA WINNOWING PADA SISTEM PENENTUAN REKOMENDASI DOSEN PEMBIMBING PROYEK AKHIR

Rudy Sofian¹, Fahmi Reza Ferdiansyah², Imam Firmansyah³

^{1,2} Manajemen Informatika, Insitut Digital Ekonomi LPKIA Bandung

³ Teknik Informatika, Institut Digital Ekonomi LPKIA Bandung

e-mail korespondensi: ¹ rudysofian@lpkia.ac.id, ² fahmirferdiansyah@gmail.com, ³ imamsyaa001@gmail.com

ABSTRACT

Faculty of Information and Digital Technology is one of the faculties at the LPKIA Digital Economics Institute. Consists of 3 study programs including Information Management, Information Systems and Informatics Engineering. Currently, the Faculty handles one of the academic processes, likely Final Project Management. The management system that is currently running is still carried out conventionally, namely by manually matching between student submissions and matching lecturers. This takes a long time and sometimes there is a discrepancy between the supervisor and the proposed student. The solution for these problems, it is proposed to apply the winnowing algorithm and jcard similarity into a web-based recommendation system that can produce the final percentage by matching student research to be carried out against previous research from prospective supervisors. Based on the results of the implementation of the winnowing and jcard similarity algorithms, it is possible to quickly and easily provide recommendations for the suitability between research submitted by students and research that has been carried out by previous supervisor candidates.

Keywords: Algorithm, Winnowing, Jcard, LPKIA, Recommendation

ABSTRAK

Fakultas Teknologi Informasi dan Digital LPKIA merupakan salah satu fakultas yang berada di Institut Digital Ekonomi LPKIA. Terdiri dari 3 program studi diantaranya Manajemen Informatika, Sistem Informasi dan Teknik Informatika. Saat ini Fakultas menangani salah satu proses akademik yaitu Pengelolaan Proyek Akhir. Sistem Pengelolaan yang saat ini berjalan masih dilakukan secara konvensional yaitu dengan pencocokan secara manual antara pengajuan mahasiswa dengan kecocokan dosen pembimbing. Hal tersebut membutuhkan waktu yang cukup lama dan terkadang terjadi ketidaksesuaian antara dosen pembimbing dengan mahasiswa yang sudah diusulkan. Untuk menangani permasalahan tersebut diusulkan penerapan algoritma winnowing dan jcard similarity kedalam sebuah sistem rekomendasi berbasis web yang dapat menghasilkan persentase akhir dengan mencocokkan penelitian mahasiswa yang akan dilakukan terhadap penelitian sebelumnya dari calon dosen pembimbing. Berdasarkan hasil implementasi algoritma winnowing dan jcard similarity dapat dengan cepat dan mudah memberikan rekomendasi kecocokan antara penelitian yang diajukan mahasiswa dengan penelitian yang telah dilakukan oleh calon pembimbing sebelumnya.

Kata Kunci: Algoritma, Winnowing, Jcard, LPKIA, Rekomendasi

PENDAHULUAN

Pada era digitalisasi khususnya pandemi ini membuat sistem terus berkembang secara cepat dan dipaksa mengganti segala aktivitas konvensional ke aktivitas komputerisasi, dimana data dan informasi bisa didapatkan secara cepat. Hal ini menyebabkan informasi

yang tersedia sangat banyak dan membuat orang kesulitan dalam menentukan pilihan. Begitu pula pada penentuan proyek akhir di Fakultas Teknologi Informasi dan Digital ini data mahasiswa dan data dosen pembimbingnya tidak sedikit, maka dari itu memerlukan sistem rekomendasi bagi

mahasiswa dan calon dosen pembimbing untuk proses penentuan proyek akhir.

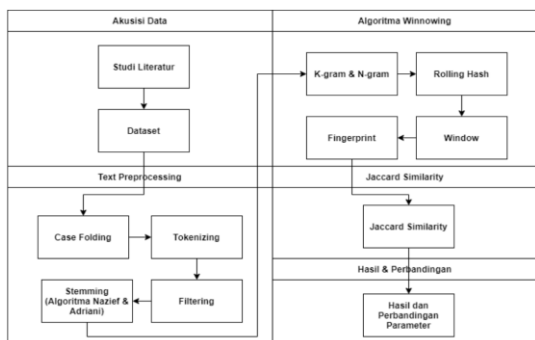
Fakultas Teknologi Informasi dan Digital merupakan fakultas yang ada di perguruan tinggi Institut Digital Ekonomi LPKIA yang berada di Soekarno-Hatta No.456, Batununggal, Bandung Kidul, Bandung City, West Java 40266. Pada sistem yang terdapat pada Fakultas Teknologi Informasi dan Digital ini baru terdapat beberapa kekurangan pada modul proyek akhir salah satunya dari sisi pengajuan topik dan penentuan pembimbing yang kurang mendukung kegiatan proses input proyek akhir. Penentuan calon dosen pembimbing masih dilakukan secara konvensional yaitu dengan jumlah data mahasiswa dan calon dosen pembimbing yang tidak sedikit, penanggung jawab proyek akhir harus melakukan pencocokan satu-persatu antara kesesuaian topik mahasiswa yang diajukan dengan kompetensi calon dosen pembimbing. Hal tersebut memerlukan waktu yang tidak sebentar, terkadang dari hasil penentuan proyek akhir antara mahasiswa dan calon dosen pembimbing terdapat adanya ketidakcocokan antara penelitian yang diajukan oleh mahasiswa dengan kompetensi maupun riwayat penelitian dosen yang pernah dilakukan sebelumnya. Maka dari itu, perlu adanya sistem rekomendasi agar dapat mendukung proses yang dapat mempercepat dan juga menghasilkan sebuah rekomendasi calon dosen pembimbing yang sesuai dengan penelitian yang diajukan.

Berdasarkan dari permasalahan diatas maka penelitian ini akan dilakukan dengan menerapkan Algoritma *Winnowing* dan *Jaccard Similarity* ke dalam sebuah sistem rekomendasi berbasis web yang dapat menghasilkan persentase akhir dengan mencocokkan penelitian mahasiswa yang akan dilakukan terhadap penelitian sebelumnya dari calon dosen pembimbing. Dalam hal ini hasil dari penelitian yang dilakukan agar bisa memberikan rekomendasi dalam proses penentuan dosen pembimbing dengan cepat dan tepat dengan meminimalisir ketidakcocokan antara penelitian mahasiswa dengan calon dosen pembimbing.

METODE

Metode yang digunakan dalam menyelesaikan permasalahan yaitu dengan menggunakan algoritma *Winnowing* yang merupakan algoritma yang digunakan untuk mendeteksi hasil kesamaan dengan melakukan perhitungan nilai-nilai hash dari setiap *k-gram*. Lalu, untuk mencari nilai hash dilakukan perhitungan menggunakan fungsi *rolling hash*. Kemudian akan dibentuk *window* dari nilai-nilai hash tersebut. Dalam setiap *window* dipilih nilai hash minimum yang akan disimpan dan dijadikan *fingerprint* dari suatu dokumen. *Fingerprint* ini yang akan dijadikan dasar pembandingan kesamaan teks yang telah disimpan [1]. Algoritma *Winnowing* dikembangkan oleh Nick Littlestone dengan memanfaatkan tingkat kesamaan dokumen yang mentransformasikan sebuah kata menjadi sebuah ukuran *k-gram*

ataupun *n-gram* [2]. Setelah pemotongan berdasarkan jumlah k-gram ataupun n-gram selanjutnya pembuatan *hashing* pada setiap pemotongan tersebut menggunakan rolling hash [3]. Dari hasil rolling hash tersebut dikelompokkan menjadi beberapa window dengan anggota window yang ditentukan, setelah dikelompokkan maka akan diidentifikasi nilai terkecil dari setiap window yang akan menjadi nilai fingerprint. Berdasarkan nilai terkecil atau fingerprint pada setiap window kemudian dicocokkan antar fingerprint dokumen lainnya, dari jumlah *fingerprint* yang sama dan akumulasi jumlah *fingerprint* di kedua dokumen [4].



Gambar 1 Alur Penelitian Algoritma Winnowing [5]

HASIL DAN PEMBAHASAN

Pembahasan yang akan dideskripsikan pada bagian ini yaitu alur algoritma *winnowing* ini digunakan dengan melalui beberapa tahapan.

Text Processing

Tahapan awal yang dilakukan pada *tahap case folding* untuk mengkonversi dokumen teks menjadi lower case.

Contoh Judul:

“Sistem Rekomendasi Pembimbing Berbasis Web dengan Menggunakan Algoritma Winnowing”

Hasil Case Folding

“sistem rekomendasi pembimbing berbasis web dengan menggunakan algoritma winnowing”

Kemudian dilanjutkan dengan tahapan *Tokenizing* yaitu tahapan yang akan menghilangkan spasi pada dokumen teks.

Hasil Tokenizing

“sistem|rekomendasi|pembimbing|berbasis|web|dengan|menggunakan|algoritma|winnowing”

Tahap selanjutnya yaitu *filtering* dimana akan dilakukan pengambilan kata yang penting dari hasil tahap tokenizing yang sudah dilakukan.

Hasil Filtering

“sistem|rekomendasi|pembimbing|berbasis|web|menggunakan|algoritma|winnowing”

Tahap terakhir yaitu *stemming* dimana dokumen teks yang sudah melalui tahapan *processing* ini akan dilakukan perubahan dokumen teks ke kata dasar dengan menghilangkan kata imbuhan [6].

Hasil Stemming

“sistem|rekomendasi|bimbing|basis|web|guna|algoritma|winnowing”

Proses Algoritma Winnowing

Pada tahapan *winnowing* ini untuk mendeteksi hasil kesamaan dengan melakukan perhitungan nilai-nilai hash dari

setiap k-gram. Lalu, untuk mencari nilai hash dilakukan perhitungan menggunakan fungsi rolling hash. Kemudian akan dibentuk window dari nilai-nilai hash tersebut. Dalam setiap window dipilih nilai hash minimum yang akan disimpan dan dijadikan fingerprint dari suatu dokumen. Fingerprint ini yang akan dijadikan dasar pembandingan kesamaan teks yang telah disimpan [1].

K-Gram

Tahapan pertama pada algoritma winnowing yaitu dengan mengambil potongan karakter huruf dengan jumlah nilai k dari sebuah teks yang secara berkelanjutan dibaca dari awal teks asal hingga akhir teks asal menggunakan angka bilangan prima 2, 3, 5, 7 [7].

Tabel 1 KGRAM

| | KGRAM (K=5) |
|-----------------------|---|
| Sampel Uji | siste, istem, stemr, temre, emrek, mreko, rekom, ekome, komen, omend, menda, endas, ndasi, dasib, asibi, sibim, ibimb, bimbi, imbin, mbing, bingb, ingba, ngbas, gbasi, basis, asisw, siswe, isweb, swebg, webgu, ebgun, bguna, gunaa, unaal, naalg, aalgo, algor, lgori, gorit, oritm, ritma, itmaw, tmawi, mawin, awinn, winno, innow, nnowi, nowin, owinn, winng |
| Data History 1 | terap, erapa, rapal, apalg, palgo, algor, lgori, gorit, oritm, ritma, itmaw, tmawi, mawin, awinn, winno, innow, nnowi, nowin, owing, wingp, ingpa, ngpad, gpada, padas, adasi, dasis, asist, siste, istem, stemr, temre, emrek, mreko, rekom, ekome, komen, omend, menda, endas, ndasi, dasit, asite, siten, itent, tentu, entud, ntudo, tudos, udose, dosen, |

| | KGRAM (K=5) |
|-----------------------|--|
| Data History 2 | osenb, senbi, enbim, nbimb, bimbi, imbin, mbing, bings, ingsk, ngskr, gskri, skrip, krips, ripsi |
| | siste, istem, stemr, temre, emrek, mreko, rekom, ekome, komen, omend, menda, endas, ndasi, dasib, asibu, sibuk, ibuku, bukub, ukuba, kubas, ubasi, basis, asisw, siswe, isweb, swebr, webre, ebret, bretri, retrie, etrie, triev, riev, ieval, evald, valde, alden, ldeng, denga, engan, ngang, gangu, angun, nguna, gunab, unabi, nabiw, abiwo, biwor, iword, wordw, ordwi, rdwin, dwinn, winno, innow, nnowi, nowin, owing, wingf, ingfi, ngfin, gfin, finge, inger, ngerp, gerpr, erpri, rprin, print |

Rolling Hash

Pada tahapan Rolling Hash yaitu salah satu metode pembuatan hashing. Hashing itu sendiri merupakan cara untuk mengubah sebuah string menjadi suatu nilai yang unik dengan panjang tertentu. Dan assets dalam proyek [8]. Fungsi Rolling Hash didefinisikan sebagai berikut:

$$H_{(ck)} = C_1 * b^{(k-1)} + C_2 * b^{(k-2)} + \dots + C_k * b^{(k-3)}$$

Rumus Rolling Hash [8]

Keterangan:

- H(ck) = Nilai Hasil
- Ck = Nilai ASCII pada karakter
- b = Basis Bilangan Prima
- k = Banyak karakter (Gram)

Contoh bila ingin membuat nilai hash dengan kata “saya” dengan basis bernilai 11, maka perhitungannya adalah:

$$H(ck) = 115 * 11(3) + 97 * 11(2) + 121 * 11(1) + 97 * 11(0)$$

$$H(ck) = 116230$$

Setelah melakukan perhitungan pada teks dokumen menggunakan fungsi rolling hash untuk merubah teks dokumen menjadi bentuk ASCII, akan dilanjutkan pada tahap window untuk mengelompokkan hasil dari perhitungan rolling hash.

Tabel 2 Rolling Hash

| | <i>Rolling Hash (K=5)</i> |
|-----------------------|---|
| Sampel UJI | 88556, 83514, 89559, 88521, 80212, 85546, 87214, 79921, 84090, 86175, 84097, 79975, 84355, 78123, 78220, 88084, 81143, 77695, 82335, 83653, 77738, 82537, 84675, 79730, 76890, 78319, 88571, 83578, 89868, 90082, 78645, 77697, 82332, 89893, 83943, 76076, 77369, 83825, 81741, 86939, 87917, 83454, 89250, 83860, 78785, 90911, 82799, 85975, 86235, 87528 |
| Data History 1 | 88572, 80457, 86768, 77693, 85451, 77369, 83825, 81741, 86939, 87917, 83454, 89250, 83860, 78785, 90911, 82799, 85975, 86235, 87528, 90877, 82607, 85010, 81397, 85225, 76230, 78140, 78316, 88556, 83514, 89559, 88521, 80212, 85546, 87214, 79921, 84090, 86175, 84097, 79975, 84355, 78125, 78236, 88170, 81576, 79865, 86923, 87845, 79959, 84268, 77695, 82335, 83653, 77755, 82632, 85149, 82100, 88737, 84425, 87855 |
| Data History 2 | 88556, 83514, 89559, 88521, 80212, 85546, 87214, 79921, 84090, 86175, 84097, 79975, 84355, 78123, 78232, 88142, 81452, 79233, 90012, 84550, 88480, 76890, 78319, 88571, 83578, 89879, 90121, 78846, 78719, 87450, 81101, 89998, 87587, 81793, 80943, 89207, 77395, 83947, 82333, 89895, 83969, 76206, 78019, 83945, 91719, 86825, 87360, 80660, 90911, 82799, 85975, 86235, 87528, 90867, 82565, |

Rolling Hash (K=5)

84810, 80403, 80241, 82569, 84832, 80524, 80850, 88735, 87541

Window

Pada tahapan window yaitu akan mengelompokkan hasil dari Rolling hash pada k-gram dan n-gram yang membentuk nilai hash menjadi beberapa window dengan ukuran window sebanyak 2 [9]. Setelah pengelompokan string dengan hasil nilai hash yang sudah dilakukan perhitungan pada rolling hash sebelumnya. Pada tahap terakhir dilakukan fingerprint untuk menghapus kata imbuhan berdasarkan kata dasar.

Tabel 3 Window

| | <i>Window</i> |
|-------------------|--|
| Sampel UJI | {88556, 83514} , {83514, 89559} , {89559, 88521} , {88521, 80212} , {80212, 85546} , {85546, 87214} , {87214, 79921} , {79921, 84090} , {84090, 86175} , {86175, 84097} , {84097, 79975} , {79975, 84355} , {84355, 78123} , {78123, 78220} , {78220, 88084} , {88084, 81143} , {81143, 77695} , {77695, 82335} , {82335, 83653} , {83653, 77738} , {77738, 82537} , {82537, 84675} , {84675, 79730} , {79730, 76890} , {76890, 78319} , {78319, 88571} , {88571, 83578} , {83578, 89868} , {89868, 90082} , {90082, 78645} , {78645, 77697} , {77697, 82332} , {82332, 89893} , {89893, 83943} , {83943, 76076} , {76076, 77369} , {77369, 83825} , {83825, 81741} , {81741, 86939} , {86939, 87917} , {87917, 83454} , {83454, 89250} , {89250, 83860} , {83860, 78785} , {78785, 90911} , {90911, 82799} , {82799, 85975} , {85975, 86235} , {86235, 87528} |

| | <i>Window</i> |
|-----------------------|---|
| Data History 1 | {88572, 80457} , {80457, 86768} , {86768, 77693} , {77693, 85451} , {85451, 77369} , {77369, 83825} , {83825, 81741} , {81741, 86939} , {86939, 87917} , {87917, 83454} , {83454, 89250} , {89250, 83860} , {83860, 78785} , {78785, 90911} , {90911, 82799} , {82799, 85975} , {85975, 86235} , {86235, 87528} , {87528, 90877} , {90877, 82607} , {82607, 85010} , {85010, 81397} , {81397, 85225} , {85225, 76230} , {76230, 78140} , {78140, 78316} , {78316, 88556} , {88556, 83514} , {83514, 89559} , {89559, 88521} , {88521, 80212} , {80212, 85546} , {85546, 87214} , {87214, 79921} , {79921, 84090} , {84090, 86175} , {86175, 84097} , {84097, 79975} , {79975, 84355} , {84355, 78125} , {78125, 78236} , {78236, 88170} , {88170, 81576} , {81576, 79865} , {79865, 86923} , {86923, 87845} , {87845, 79959} , {79959, 84268} , {84268, 77695} , {77695, 82335} , {82335, 83653} , {83653, 77755} , {77755, 82632} , {82632, 85149} , {85149, 82100} , {82100, 88737} , {88737, 84425} , {84425, 87855} |
| | {88556, 83514} , {83514, 89559} , {89559, 88521} , {88521, 80212} , {80212, 85546} , {85546, 87214} , {87214, 79921} , {79921, 84090} , {84090, 86175} , {86175, 84097} , {84097, 79975} , {79975, 84355} , {84355, 78123} , {78123, 78232} , {78232, 88142} , {88142, 81452} , {81452, 79233} , {79233, 90012} , {90012, 84550} , {84550, 88480} , {88480, 76890} , {76890, 78319} , {78319, 88571} , {88571, 83578} , {83578, 89879} , {89879, 90121} , {90121, 78846} , {78846, 78719} , {78719, 87450} , {87450, 81101} , {81101, 89998} , {89998, 87587} , {87587, 81793} , {81793, 80943} , {80943, 89207} , {89207, 77395} , {77395, 83947} , {83947, 82333} , {82333, 89895} , {89895, 83969} , {83969, 76206} , {76206, 78019} , {78019, 83945} , {83945, 91719} , {91719, 86825} , {86825, 87360} , {87360, 80660} , {80660, 90911} , {90911, 82799} , {82799, 85975} , {85975, 86235} , {86235, 87528} , {87528, 90867} , {90867, 82565} , {82565, 84810} , {84810, 80403} |

| <i>Window</i> |
|---|
| {80403, 80241} , {80241, 82569} , {82569, 84832} , {84832, 80524} , {80524, 80850} , {80850, 88735} , {88735, 87541} |

Fingerprint

Pada tahapan fingerprint merupakan metode pengenyaleksian dari nilai yang terkecil pada setiap window yang sudah dikelompokan, lalu apabila menemukan nilai yang terkecil yang sama maka akan diambil nilai salah satunya.

Tabel 4 Fingerprint

| | <i>Fingerprint</i> |
|-------------------|--|
| Sampel Uji | K5 83514, 83514, 88521, 80212, 80212 , 85546, 79921, 79921, 84090, 84097 , 79975, 79975, 78123, 78123, 78220 , 81143, 77695, 77695, 82335, 77738 , 77738, 82537, 79730, 76890, 76890 , 78319, 83578, 83578, 89868, 78645 , 77697, 77697, 82332, 83943, 76076 , 76076, 77369, 81741, 81741, 86939 , 83454, 83454, 83860, 78785, 78785 , 82799, 82799, 85975, 86235 |
| | 80457, 80457, 77693, 77693, 77369 , 77369, 81741, 81741, 86939, 83454 , 83454, 83860, 78785, 78785, 82799 , 82799, 85975, 86235, 87528, 82607 , 82607, 81397, 81397, 76230, 76230 , 78140, 78316, 83514, 83514, 88521 , 80212, 80212, 85546, 79921, 79921 , 84090, 84097, 79975, 79975, 78125 , 78125, 78236, 81576, 79865, 79865 , 86923, 79959, 79959, 77695, 77695 , 82335, 77755, 77755, 82632, 82100 , 82100, 84425, 84425 |

| <i>Fingerprint</i> | |
|-----------------------|---|
| K5 | |
| Data Histori 2 | 83514, 83514, 88521, 80212, 80212, 85546, 79921, 79921, 84090, 84097, 79975, 79975, 78123, 78123, 78232, 81452, 79233, 79233, 84550, 84550, 76890, 76890, 78319, 83578, 83578, 89879, 78846, 78719, 78719, 81101, 81101, 87587, 81793, 80943, 80943, 77395, 77395, 82333, 82333, 83969, 76206, 76206, 78019, 83945, 86825, 86825, 80660, 80660, 82799, 82799, 85975, 86235, 87528, 82565, 82565, 80403, 80241, 80241, 82569, 80524, 80524, 80850, 87541 |

JCard Similarity

Pada tahapan output similarity ini merupakan fitur yang diambil untuk pengukuran kemiripan dua buah dokumen teks yang sudah diproses dari tahapan sebelumnya agar bisa dibandingkan dan menghasillkan nilai akhir dalam bentuk persentase tertinggi [10] pada tahapan similarity ini perhitunganya menggunakan metode Jaccard Similarity untuk menghasillkan nilai kesamaan antara dokumen teks satu dengan teks dokumen lainnya [5].

Berikut rumus yang digunakan dalam perhitungan *Jaccard Similarity*:

$$\text{Similarity (X, Y)} = \frac{|X \cap Y|}{|X \cup Y|}$$

Rumus Jcard Similariy [11]

Keterangan:

X = Dokumen 1

Y = Dokumen 2

$|X \cap Y|$ = Jumlah *Finger* yang sama

$|X \cup Y|$ = Jumlah *Finger* yang berbeda

Pada rumus di atas merupakan rumus Jaccard Similarity atau Jaccard Coefficient yang digunakan untuk mencari kemiripan pada dua dokumen.

Pada data sampel diatas akan diuji kemiripan antara dokumen *Sampel Uji* dengan 2 dokumen *Data History* maka didapat perhitungan seperti dibawah ini

Tabel 5 Jcard Sampel Uji x Data History 1

| | Jumlah | Jcard |
|-----------------------|--------|----------|
| | Finger | Sama |
| Sampel Uji | 49 | |
| Data History 1 | 58 | 28, 0,35 |

Tabel 5 Jcard Sampel Uji x Data History 2

| | Jumlah | Jcard |
|-----------------------|--------|----------|
| | Finger | Sama |
| Sampel Uji | 49 | |
| Data History 1 | 63 | 23, 0,26 |

Implementasi Perangkat Lunak

Dari proses perhitungan diatas maka diimplementasikan kedalam sebuah sistem informasi berbasis web untuk mendukung proses pengelolaan perangkat lunak.

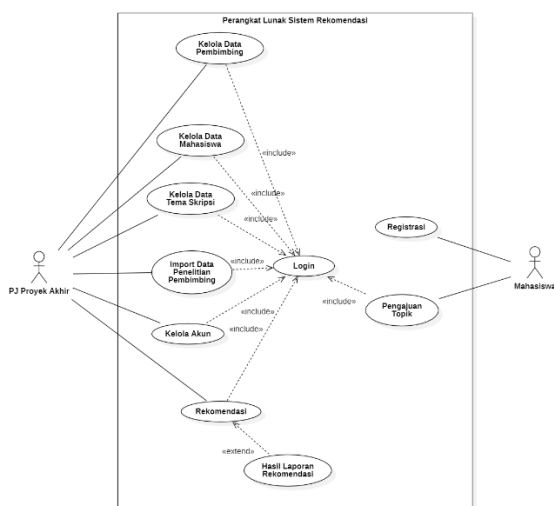
Analisis Fungsional

Perangkat lunak ini terdiri dari 2 buah aktor yaitu PJ Proyek Akhir dan Mahasiswa. Dimana PJ Proyek akhir bertugas untuk

mengelola sistem informasi dan melihat data rekomendasi penelitian.

Sedangkan mahasiswa bertugas untuk mengisi data penelitian atau judul skripsi yang akan diajukan. Dan nantinya sistem dapat menampilkan data rekomendasi pembimbing yang sesuai dengan judul yang diajukan oleh mahasiswa.

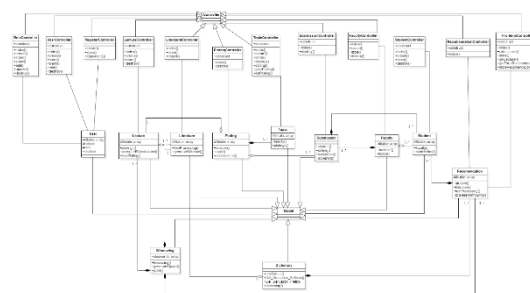
Terlihat pada Gambar 2 menjabarkan mengenai diagram *use case* mengenai sistem yang dibuat.



Gambar 2 Use Case Diagram Sistem Rekomendasi

Class Diagram

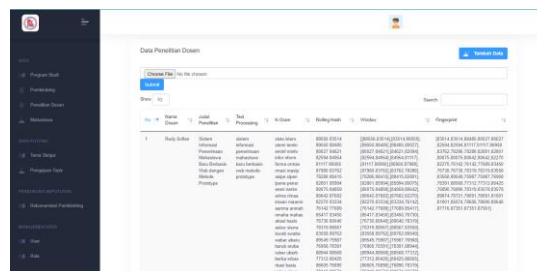
Pada Gambar 3 menjabarkan *class-class* yang dibutuhkan untuk membuat sebuah sistem rekomendasi ini. Class dijabarkan dengan mengusing konsep *framework* yang dibuat dengan Bahasa pemrograman PHP dengan *framework* Laravel



Gambar 3 Class Diagram Sistem Rekomendasi Implementasi Aplikasi

Dari perancangan dan konseptual yang telah dibuat, maka Langkah selanjutnya adalah menerapkan algoritma *winning* kedalam sebuah sistem. Berikut beberapa tampilan antarmuka fitur inti yang telah dibuat.

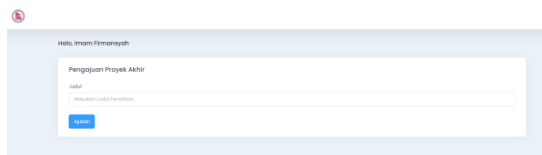
Import Data History Penelitian Pembimbing



Gambar 4 Tampilan Antarmuka Import Data Penelitian

Gambar 4 merupakan sebuah fitur untuk mengelola data penelitian yang pernah dilakukan oleh pembimbing sebelumnya. Data penelitian ini nantinya akan menjadi *data history* yang akan dicocokkan dengan data penelitian yang diajukan oleh mahasiswa.

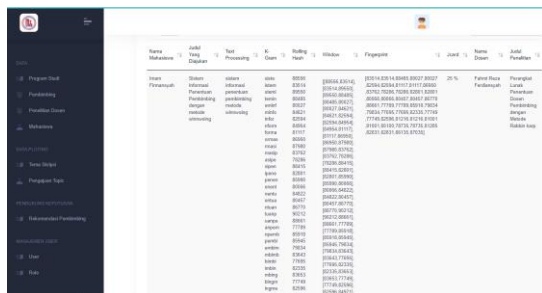
Pengajuan Judul Skripsi Mahasiswa



Gambar 5 Tampilan Antarmuka Pengajuan Judul Skripsi Mahasiswa

Gambar 5 merupakan tampilan antarmuka di sisi mahasiswa. Fitur ini digunakan oleh mahasiswa untuk mengajukan judul penelitian atau judul tugas akhir.

Laporan Rekomendasi Pembimbing Skripsi



Gambar 6 Laporan Rekomendasi Pembimbing Skripsi

Gambar 6 merupakan fitur untuk melihat kecocokan antara data penelitian yang diajukan oleh mahasiswa dengan data penelitian pembimbing terdahulu dengan menggunakan *JCard Similarity*.

SIMPULAN

Dari hasil dan pembahasan diatas maka dengan diterapkannya algoritma winnowing dan jcard similarity dapan membantu PJ Proyek Akhir dalam menentukan kecocokan antara penelitian yang akan dilakukan oleh mahasiswa dan Riwayat penelitian yang telah dilakukan oleh dosen pembimbing. Output

berupa persentase kecocokan 2 dokumen dimana data uji dan data history 1 menunjukkan kemiripan sebesar 35% sedangkan data uji dan data history 2 menunjukkan kemiripan sebesar 26%. Namun perlu disimpulkan bahwa rekomendasi ini bukan keputusan akhir hanya saja dapat membantu PJ dalam meminilisir waktu dan kesalahan dalam pencocokan.

DAFTAR PUSTAKA

- [1] S. E. Suroani Azis, “Klasifikasi Bank Soal Berdasarkan Kompetensi Dasar Menggunakan Hasil Similarity tertinggi Algoritma Winnowing,” Vol. 15, No. 2, Pp. 163–173, 2021.
- [2] M. Balcan, “8803 Machine Learning Theory,” *Mach. Learn.*, Pp. 1–4, 2011.
- [3] S. Yuliyanti And Rizky, “Implementasi Algoritma Rabin Karp Untuk Mendeteksi Kemiripan Dokumen Stmik Bandung,” *J. Bangkit Indones.*, Vol. 10, No. 02, P. 1, 2020, Doi: 10.52771/Bangkitindonesia.V10i02.124.
- [4] M. N. Khidfi And J. Y. Sari, “Rancang Bangun Aplikasi Pendeteksian Kesamaan Pada Dokumen Teks Menggunakan Algoritma,” Vol. 4, No. 2, Pp. 1–10, 2018, Doi: 10.5281/Zenodo.1407866.
- [5] W. Hidayat, E. Utami, And A. D. Hartanto, “Pemilihan Parameter Terbaik Pada Algoritma Winnowing Dalam Mendeteksi Tingkat Kesamaan Dokumen Bahasa Indonesia Selection Of The Best Parameters In The Winnowing Algorithm In Detecting The Level Of Similarity In Indonesian Documents,” *Citec J.*, Vol. 7, No. 2, 2020.
- [6] I. Bagus And K. Surya, “Implementasi Algoritma Winnowing Dalam Mendeteksi,” No. 1, Pp. 220–230, 2022.
- [7] R. Purnamasari, M. Fairuzabadi, And A. Riyadi, “Sistem Pengecekan

- Plagiasi Judul Tugas Akhir Menggunakan Algoritma Winnowing Di Fakultas Sains Dan Teknologi Universitas Pgrri Yogyakarta,” *Semin. Nas. Din. Inform.*, Pp. 16–20, 2021.
- [8] J. Evan Harya Chandra, V. Christiani M, And D. S.Naga, “Plagiarisme Abstrak Menggunakan Algoritma Winnowing Dan Synsets,” *J. Ilmu Komput. Dan Sist. Inf.*, Pp. 121–129, 2018.
- [9] W. Pramudita, B. P. Tomasouw, Z. A. Leleury, And M. E. Rijoly, “Design Of Undergraduated Thesis Plagiarism Detection System (Title And Anstract), Based On Matlab , Using Winnowing Algorithm,” Vol. 2, No. 2, Pp. 67–76, 2021.
- [10] F. Y. Pramudya, I. Moch, A. Bijaksana, M. Tech, And D. Ph, “Pemberian Peringkat Komentar Pada Community Question Answering Dengan Fitur Soft-Cosine Semantic Similarity Untuk Kasus Question-External Comment,” *E-Proceeding Eng.*, Vol. 6, No. 1, Pp. 2199–2207, 2019.
- [11] W. Desena And A. Solichin, “Pencarian Abstrak Tugas Akhir Mahasiswa Berdasarkan Tingkat Kemiripan Menggunakan Algoritma Winnowing Dan Jaccard Similarity Pada Universitas Budi Luhur,” *J. Inform.*, Vol. 4221, Pp. 112–122, 2021.