JITET (Jurnal Informatika dan Teknik Elektro Terapan)
Vol. 12 No. 3, pISSN: 2303-0577 eISSN: 2830-7062 http://dx.doi.org/10.23960/jitet.v12i3.4807

IMPLEMENTASI ALGORITMA *WEIGHTED TREE SIMILARITY* DAN *CONTENT BASED FILTERING* DALAM PENCARIAN SKRIPSI

Natalia Matondang[1*], Yisti Vita Via[2], Fawwaz Ali Akbar[3]

[1,2,3]Universitas Pembangunan Nasional "Veteran" Jawa Timur; Jl. Rungkut Madya No.1, Surabaya; telp/Fax +62 (031) 870 6372

Received: 10 Juli 2024
Accepted: 31 Juli 2024
Published: 7 Agustus 2024

Keywords:
3-5 keyword;
Weighted Tree Similarity;
Content based filtering;
Sistem Pencarian.

Corespondent Email:
matondangnatalia347@gmail.com

Abstrak. UPN "Veteran" Jawa Timur telah memiliki sebuah sistem *repository* untuk menyimpan semua data terkait skripsi yang telah diselesaikan oleh mahasiswa. Sistem tersebut dapat diakses secara *online* oleh seluruh mahasiswa UPN "Veteran" Jawa Timur. Namun terdapat kekurangan pada sistem tersebut, yaitu fitur pencarian yang belum dapat memberikan hasil yang relevan dengan *input* yang diberikan oleh pengguna. Oleh karena itu, penulis membuat sebuah sistem pencarian hasil penelitian skripsi agar mahasiswa dapat menemukan daftar judul yang relevan dengan topik yang ingin dicari oleh mahasiswa. Sistem menggunakan algoritma *Weighted tree similarity* dan *Content based filtering* agar hasil pencarian berorientasi pada atribut skripsi. Dilakukan pengujian pada sistem menggunakan *recall* dan *precision* dan mendapatkan hasil *precision* 74% dan *recall* 83%. Dengan demikian, sistem ini diharapkan mampu membantu mahasiswa untuk menemukan skripsi sesuai dengan topik yang diinginkan dan mengurangi peluang terjadinya *plagiarisme* atau kesamaan judul skripsi.

Abstract. *UPN "Veteran" East Java has a repository system to store all data related to theses completed by students. This system can be accessed online by all UPN "Veteran" East Java students. However, the system has a shortcoming: its search feature cannot provide results relevant to the user's input. Therefore, the author developed a thesis research search system to help students find lists of titles relevant to the topics they want to search for. The system uses the Weighted tree similarity algorithm and content-based filtering to ensure that search results are oriented towards thesis attributes. The system was tested using recall and precision, achieving a precision of 74% and a recall of 83%. Thus, this system is expected to help students find theses that match their desired topics and reduce the chances of plagiarism or title similarities.*

1. PENDAHULUAN

Sistem pencarian adalah suatu sistem yang memberikan daftar informasi yang relevan dengan masukan yang diberikan oleh pengguna. Sistem pencarian memberikan kemudahan bagi pengguna untuk menemukan suatu hal dalam waktu yang singkat tanpa perlu melakukan pencarian secara manual. Banyak perusahaan besar yang sudah menggunakan sistem pencarian pada layanan digital yang ditawarkan seperti Netflix, Shopee, dan Google. Hal ini karena sistem pencarian memberikan kenyamanan bagi pengguna untuk mempercepat proses pengguna dalam menemukan suatu produk.

Sebelum penyusunan skripsi, mahasiswa cenderung melihat skripsi yang telah diselesaikan oleh alumni yang sesuai dengan jurusan yang ditekuni untuk dijadikan sebagai referensi. Alasan lain yaitu untuk menghindari penggunaan judul yang sama agar tidak menimbulkan plagiarisme. Skripsi yang ada di ruang baca disusun rapi di rak berdasarkan tahun. Mahasiswa akan membutuhkan waktu yang lama untuk menemukan skripsi dengan topik tertentu apabila mencari secara manual. Apabila skirpsi tersebut sedang digunakan, maka mahasiswa harus menunggu atau mencari skripsi lain yang memiliki topik sama. Hal ini dapat diselesaikan dengan penggunaan *repository* sehingga mahasiswa dapat mengakses skripsi secara *online*.

UPN "Veteran" Jawa Timur telah memiliki sebuah sistem *repository* untuk menyimpan semua data terkait skripsi yang telah diselesaikan oleh mahasiswa. Sistem tersebut dapat diakses secara *online* oleh seluruh mahasiswa UPN "Veteran" Jawa Timur. Namun terdapat suatu kekurangan pada sistem tersebut, yaitu fitur pencarian yang belum dapat memberikan hasil yang relevan dengan input yang diberikan oleh pengguna. Pengguna hanya akan mendapat hasil yang relevan apabila memberikan inputan berupa judul lengkap terkait skripsi yang ingin dicari. Sehingga untuk dapat mengakses skripsi yang ada di *repository*, mahasiswa harus mengetahui judul skripsi yang akan dicari terlebih dahulu. Sedangkan mahasiswa sering kali mencari hanya menggunakan kata kunci berupa topik atau metode yang akan dipakai. Hal tersebut membuat proses mahasiswa menemukan skripsi menjadi tidak efektif. Oleh karena itu, penulis membuat sebuah sistem pencarian hasil penelitian skripsi agar mahasiswa dapat menemukan daftar judul yang relevan dengan topik yang ingin dicari oleh mahasiswa.

Content based filtering merupakan metode penyaringan konten berdasarkan informasi atau atribut tertentu untuk menghasilkan rekomendasi yang disesuaikan dengan pengguna[1]. Metode ini efektif digunakan apabila data memiliki sejumlah atribut yang dapat dijadikan sebagai parameter penentu keputusan. Skripsi memiliki sejumlah atribut seperti judul, pengarang, abstrak, dan tahun. Atribut tersebut dapat dijadikan sebagai parameter pencarian sehingga hasil pencarian menjadi lebih akurat dan relevan dengan kebutuhan pengguna. *Weighted tree similarity* merupakan metode pencarian dengan representasi dalam bentuk *tree* dengan cabang yang memiliki bobot dan label[2]. *Weighted tree similarity* digunakan untuk meningkatkan akurasi pencarian dengan mempertimbangkan struktur hierarkis dari penelitian skripsi. Dengan menggabungkan kedua metode tersebut, diharapkan sistem pencarian hasil penelitian skripsi ini mampu memberikan hasil yang lebih akurat dan relevan sesuai dengan kebutuhan pengguna.

Pada tahun 2020, Alkaff et al. melakukan sebuah penelitian dengan judul "Sistem Rekomendasi Buku Menggunakan *Weighted tree similarity* dan *Content based filtering*"[3]. Penelitian ini membahas terkait pembuatan sebuah sistem rekomendasi buku dengan menggunakan metode *Content based filtering* dengan menganalisa kemiripan tiap buku dari fitur yang dikandungnya dengan weighted *tree similarity*. Menerapkan metode *Content based filtering* dengan membandingkan 3 parameter data berupa judul, pengarang, dan sinopsis tiap buku lalu menghitung nilai *similarity* tiap parameter berdasarkan bobot TF tiap kata. Lalu nilai *similarity* total dihitung berdasarkan tiap bobot parameter pada *tree* sehingga didapat 5 buku terpilih dengan nilai *similarity* tertinggi. Dari hasil pengujian *precision*, sistem menghasilkan performa yang baik melalui 5 skenario pengujian dan memiliki akurasi sebesar 88%.

Berdasarkan latar belakang diatas, penulis tertarik untuk membuat sebuah sistem pencarian skripsi untuk mempermudah maasiswa menemukan skripsi yang relevan dengan topik yang ingin dicari. Penelitian ini mengubah konsep sistem rekomendasi menjadi sistem pencarian dengan menambahkan *query* masukan sebagai acuan mencari dokumen yang relevan. Data yang digunakan pada penelitian sebelumnya berupa buku dengan atribut judul, pengarang dan sinopsis, tetapi penelitian ini tidak menggunakan penulis sebagai atribut karena setiap mahasiswa hanya membuat satu skripsi sehingga penelitian ini hanya menggunakan atribut judul dan abstrak sebagai parameter penentu hasil pencarian. Penggunaan *precision score* pada penelitian sebelumnya dapat memberikan gambaran nilai akurasi sistem temu kembali informasi dengan baik

sehingga pada penelitian ini akurasi hasil pencarian akan dievaluasi menggunakan *precision score* beserta *recall*.

2. TINJAUAN PUSTAKA

2.1. Sistem Pencarian

Sistem pencarian adalah suatu sistem yang dirancang untuk menemukan informasi yang relevan dari suatu kumpulan data atau sumber informasi. Pengguna akan memberikan suatu kata kunci kepada sistem. Sistem akan melakukan proses pencarian data/dokumen yang berkaitan dengan kata kunci dan menampilkan dokumen berdasarkan tingkat kemiripan dengan kata kunci. Sistem pencarian merupakan alat yang mampu mempermudah pengguna dalam menemukan informasi terkait sebuah barang[4].

2.2. Sistem Temu Kembali Informasi

Sistem Temu Kembali Informasi (*Information Retrieval System*) adalah suatu sistem yang dirancang untuk mencari dan memperoleh informasi yang relevan dari sekumpulan data yang besar berdasarkan kebutuhan pengguna. Dasar teori dari sistem ini mencakup proses pengindeksan dokumen, pencocokan *query*, dan penilaian relevansi. Sistem temu kembali informasi merupakan alat pengukur rasio data perolehan (*recall*) dan ketepatan (*precision*)[5].

2.3. Text Preprocessing

Data awal yang digunakan dalam penelitian berupa teks yang masih memiliki noise dan belum terstruktur. Data tersebut akan menyulitkan komputer untuk mengolah data. Oleh karena itu, *text preprocessing* dilakukan untuk sebelum penentuan fitur[6]. Text *preprocessing* adalah proses mengubah data teks yang tidak terstruktur menjadi kumpulan kata yang dapat diproses oleh komputer[7]. Pada penelitian ini, terdapat sejumlah tahapan dalam text *preprocessing*, yaitu *case folding*, *tokenizing*, *filtering*, *stemming*

2.4. Pembobotan TF-IDF

TF-IDF merupakan suatu metode pembobotan antar suatu kata dengan suatu dokumen. Terdapat dua konsep perhitungan pada algoritma TF-IDF, yaitu penghitungan frekuensi suatu kata di dalam sebuah dokumen (TF) dan penghitungan *inverse* frekuensi dokumen yang mengandung kata tersebut (IDF)[8]. Nilai TF dan IDF digabungkan untuk mendapatkan nilai bobot *term* tiap kata. Sehingga menghasilkan rumus sebagai Berikut :

$$W(i,j) = \text{tf}(i,j)\ x\ \log\left(\frac{N}{df(i)}\right) \quad (1)$$

Keterangan :
$W(i,j)$ = bobot dokumen
$TF(i,j)$ = banyaknya kemunculan *term* t_i pada dokumen d_j
N = jumlah dokumen yang terambil oleh sistem

2.5. Content based filtering

Content based filtering memberi sejumlah rekomendasi berdasarkan hubungan antar deskripsi item. Metode *Content based filtering* menggunakan deskripsi item sebagai dasar penilaian untuk menghasilkan suatu rekomendasi[9]. Keuntungan dari penggunaan *Content based filtering* adalah kemampuan untuk memberikan rekomendasi dengan mempertimbangkan sejumlah parameter sekaligus.

2.6. Cosine Similarity

Cosine similarity merupakan metode untuk menghitung kemiripan antar vektor dokumen dengan vektor *query* berdasarkan sudut terkecil[10]. Nilai *similarity* antara *item* a dan b dapat dihitung menggunakan rumus di bawah ini :

$$S = \frac{\sum(W(i,\text{q}) * W(i,\text{p}))}{\sqrt{\sum(W(i,\text{q})^2)} * \sqrt{\sum(W(i,\text{p})^2)}} \quad (2)$$

Keterangan :
S = Tingkat kemiripan
$W(i,\text{q})$ = Bobot kata i pada *query*
$W(i,\text{p})$ = Bobot kata i pada dokumen p

2.7. Weigted Tree Similarity

Weighted tree similarity merupakan algoritma yang dapat digunakan untuk menghitung kesamaan antar dua objek dalam bentuk *tree*. Dokumen yang sudah melalui proses *preprocessing* dan TF-IDF diubah dalam

bentuk *tree*. Lalu dilakukan perhihtungan nilai kemiripan dari tiap parameter terhadap *query* dengan menggunakan *cosine similarity*. Kemudian dilakukan penghitungan bobot tiap parameter dengan membagi total *term* frekuensi tiap parameter dengan total frekuensi *term* keseluruhan dokumen[3].

$$W = TF_p \,/\, TF_{total} \qquad (3)$$

Keterangan :
W = bobot cabang
TF = frekuensi seluruh kata pada parameter p
TF_{total} = total frekuensi kata pada seluruh parameter

Nilai bobot cabang pada *tree* digunakan untuk menghitung total *similarity* antara *query* dengan dokumen. Perhitungan total *similarity* dapat dihitung dengan rumus sebagai berikut :

$$sim_{tot} = \sum(S_i \, x \, W_i) \qquad (4)$$

Keterangan :
S_i = kemiripan cabang
W_i = bobot cabang

2.8. Precision

Precision score merupakan metode penghitungan nilai uji suatu data dengan menggunakan nilai *true positives* dan *false positives*[11]. Nilai *precision* didapat melalui perbandingan antara hasil relevan terhadap semua pencarian yang ditemukan. Perhitungan nilai *precision* dapat menggunakan rumus sebagai berikut :

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \qquad (5)$$

Keterangan :
TP = *True Positive* / Dokumen yang relevan dengan *query*
FP = *False Positive*/ Dokumen yang tidak relevan

2.9. Recall

Recall adalah tingkat keberhasilan sistem dalam mengembalikan informasi yang dicari[12]. Rumus menghitung *recall* ialah dengan menghitung rasio jumlah dokumen relevan yang ditemukan dibandingkan dengan jumlah keseluruhan dokumen relevan. Hasil penghitungan *recall* menunjukkan kemampuan sistem untuk memanggil dokumen yang sesuai dengan pencarian.

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \qquad (6)$$

Keterangan :
TP = *True Positive* / Dokumen yang relevan dengan *query*
FP = *False Positive*/ Dokumen yang relevan tetapi terpanggil
FN = False Negative/Dokumen yang relevan tetapi tidak terpanggil

3. METODE PENELITIAN

3.1. Pengumpulan Data

Dataset yang digunakan ialah data skripsi yang ada di ruang baca FASILKOM UPN "Veteran" Jawa Timur. Penulis mengambil data skripsi dengan pengumpulan data melalui studi pustaka. Proses ini melibatkan poses pencarian, identifikasi, dan pengumpulan informasi skripsi yang ada di perpustakaan dan website *repository* UPNVJT. Penulis mengambil data terkait skripsi dengan atribut berupa judul, pengarang, abstrak, jurusan, dan *link*. Informasi tersebut disimpan dalam bentuk excel. Penulis menggunakan 100 skripsi sebagai *Dataset*.

Gambar 1 Kumpulan Skripsi di Ruang Baca FASILKOM

Gambar 1 merupakan tampilan dari kumpulan skripsi yang berada di ruang baca FASILKOM. Pihak ruang baca FASILKOM tidak memiliki arsip terkait skripsi dalam bentuk excel sehingga penulis mengambil data dari *repository* UPN "Veteran" Jawa Timur dan menyimpan data tersebut dalam bentuk excel.

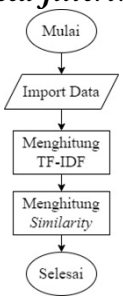
Penulis menggunakan 100 skripsi sebagai *Dataset*.

3.2. Preprocessing Data

Pada tahap ini, dilakukan beberapa proses untuk mempersiapkan data agar lebih terstruktur sehingga dapat diproses ketahap perancangan sistem. Terdapat beberapa proses pada tahap *preprocessing* yang dilakukan pada penelitian ini diantaranya adalah *case folding*, *tokenizing*, *filtering*, dan *stemming*.

3.3. Perancangan Sistem

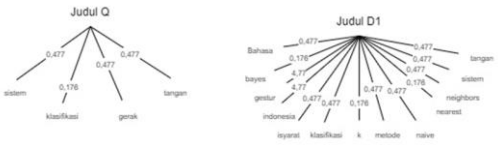
3.3.1. Content based filtering



Gambar 2 Tahap *Preprocessing* Data

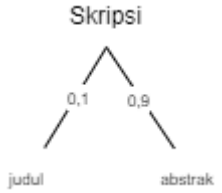
Pada gambar 2 terdapat alur proses *content based filtering*. Diawali dengan memasukkan data. Data tersebut merupakan data yang sudah selalui proses *preprocessing*. Setiap *term* pada data akan melalui proses pembobotan. Pada metode ini, penentuan bobot *term* dilakukan dengan skema TF-IDF.

3.3.2. Weighted Tree Similarity



Gambar 3 Contoh *Tree* Judul Q dan Judul D1

Gambar 3 merupakan representasi *term* yang ada pada judul Q dan judul D1. Tiap *term* akan menjadi cabang dan memiliki bobot. Nilai bobot didapat dari perhitungan TF-IDF. Setelah dilakukan penghitungan nilai TF-IDF , maka dilanjutkan dengan menghitung nilai *similarity* pada tiap-tiap parameter terhadap *query* dengan menggunakan *cosine similarity*. Selanjutnya menghitung bobot tiap parameter. Bobot parameter didapat dari membagi jumlah kata pada parameter tertentu dengan jumlah kata pada keseluruhan dokumen.



Gambar 4 Contoh *tree* skripsi

Gambar 4 merupakan representasi skripsi dalam bentuk *tree*. Langkah terakhir untuk menghitung tingkat kemiripan setiap skripsi dengan *query* ialah dengan mengalikan nilai kemiripan tiap parameter dengan tiap bobot parameter.

3.4. Skenario Pengujian

Skenario pengujian dilakukan untuk menghitung tingkat akurasi suatu sistem. Penulis menggunakan skenario pengujian presisi (*precision testing*) dan *recall* pada hasil pencarian. Pengujian dilakukan dengan menjalankan sistem dan menganalisis hasil pencarian yang diberikan oleh sistem. Setiap hasil pencarian akan dinilai Apakah relevan atau tidak. Apabila Ketika sistem dijalankan ditemukan hasil D1 dan D3 maka hasil pencarian dari diuji maka pengujian *precision* dan *recall* dapat dilakukan. Dari D1, D2, dan D3, hanya ada 1 skripsi yang relevan dengan *query* yaitu D1.

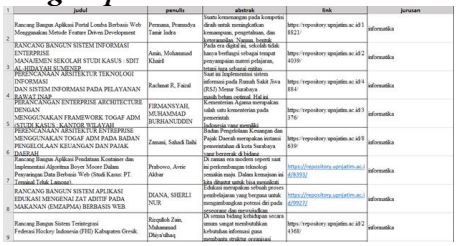
Precision =1/(1+1) x 100=0.5

Recall =1/(1+0)=1

Dari hasil evaluasi dapat disimpulkan bawa sistem memiliki nilai *precision* sebesar 50% dan *recall* 100%. Nilai *precision* menunjukkan bahwa 50% dari data yang ditemukan merupaka skripsi yang relevan dengan *query*. Sedangkan *recall* menunjukkan bahwa setiap data yang relevan dengan *query* dapat ditemukan.

4. HASIL DAN PEMBAHASAN

4.1. Pengumpulan Data



	judul	penulis	abstrak	link	jurusan
1	Rancang Bangun Aplikasi Portal Lomba Berbasis Web Menggunakan Metode Feature Driven Development	Permana, Pramudya Tamir Indra	Suatu kemenangan pada kompetisi diraih untuk meningkatkan kemampuan, pengetahuan, dan keterampilan. Namun, banyak...	https://repository.upnjatim.ac.id/18821/	informatika
2	RANCANG BANGUN SISTEM INFORMASI ENTERPRISE MANAJEMEN SEKOLAH STUDI KASUS : SDIT AL-HIDAYAH SUNENEP	Amin, Mohammad Khairil	Pada era digital ini, sekolah tidak hanya berfungsi sebagai tempat penyampaian materi pelajaran, tetapi juga sebagai entitas...	https://repository.upnjatim.ac.id/24039/	informatika
3	PERENCANAAN ARSITEKTUR TEKNOLOGI INFORMASI DAN SISTEM INFORMASI PADA PELAYANAN BANGKIT	Rachmat R, Faizal	Saat ini implementasi sistem informasi pada Rumah Sakit Jiwa (RSJ) Menur Surabaya masih belum optimal. Hal ini...	https://repository.upnjatim.ac.id/4884/	informatika
4	PERANCANGAN ENTERPRISE ARCHITECTURE DENGAN MENGGUNAKAN FRAMEWORK TOGAF ADM (STUDI KASUS : KANTOR WILAYAH	FIRMANSYAH, MUHAMMAD BURHANUDDIN	Kementerian Agama merupakan salah satu kementerian pada pemerintah Indonesia yang memiliki...	https://repository.upnjatim.ac.id/3376/	informatika
5	PERENCANAAN ARSITEKTUR ENTERPRISE MENGGUNAKAN TOGAF ADM PADA BADAN PENGELOLAAN KEUANGAN DAN PAJAK DAERAH	Zamani, Sahadi Ilahi	Badan Pengelolaan Keuangan dan Pajak Daerah merupakan instansi pemerintahan di kota Surabaya yang bertugas di bidang ...	https://repository.upnjatim.ac.id/8639/	informatika
6	Rancang Bangun Aplikasi Pendirian Koperasi dan Implementasi Algoritma Boyer Moore Dalam Penyesuaian Data Berbasis Web (Studi Kasus: PT. Temanal Tidak Lancar)	Prabowo, Arvie Akbar	Di zaman era modern seperti saat ini perkembangan teknologi semakin maju. Dalam kenajuan ini kita dituntut untuk bisa menyelesai...	https://repository.upnjatim.ac.id/8393/	informatika
7	RANCANG BANGUN SISTEM APLIKASI EDUKASI MENGENAI ZAT ADITIF PADA MAKANAN (ENZAPMA) BERBASIS WEB	DIANA, SHERLI NUR	Edukasi merupakan sebuah proses pembelajaran yang berguna untuk mengembangkan potensi diri pada seseorang dan menyediakan...	https://repository.upnjatim.ac.id/6992/	informatika
8	Rancang Bangun Sistem Terintegrasi Federasi Hockey Indonesia (FHI) Kabupaten Gresik	Rizqillah Zam, Muhammad Dhiyaulhaq	Di semua bidang kehidupan secara umum sangat membutuhkan keberadaan informasi guna mendukung struktur organisasi...	https://repository.upnjatim.ac.id/4368/	informatika

Gambar 5 Data Skripsi

Gambar 5 merupakan tampilan *Dataset* yang digunakan. Terdapat 5 kolom yang terdiri dari judul, penulis, abstrak, *link*, dan jurusan. Data disimpan ke dalam file excel. File tersebut berisikan 100 data terkait skripsi.

4.2. Preprocessing Data

	judul	penulis	abstrak	link	jurusan
0	Rancang Bangun Aplikasi Portal Lomba Berbasis ...	Permana, Pramudya Tamir Indra	Suatu kemenangan pada kompetisi diraih untuk m...	https://repository.upnjatim.ac.id/18821/	informatika
1	RANCANG BANGUN SISTEM INFORMASI ENTERPRISE \nM...	Amin, Mohammad Khairil	Pada era digital ini, sekolah tidak hanya berf...	https://repository.upnjatim.ac.id/24039/	informatika
2	PERENCANAAN ARSITEKTUR TEKNOLOGI INFORMASI\nDA...	Rachmat R, Faizal	Saat ini implementasi sistem informasi pada Ru...	https://repository.upnjatim.ac.id/4884/	informatika
3	PERANCANGAN ENTERPRISE ARCHITECTURE DENGAN\nME...	FIRMANSYAH, MUHAMMAD BURHANUDDIN	Kementerian Agama merupakan salah satu kemente...	https://repository.upnjatim.ac.id/3376/	informatika
4	PERENCANAAN ARSITEKTUR ENTREPRISE \nMENGGUNAKA...	Zamani, Sahadi Ilahi	Badan Pengelolaan Keuangan dan Pajak Daerah me...	https://repository.upnjatim.ac.id/8639/	informatika

Gambar 6 *Dataset* sebelum *preprocessing*

Data yang disimpan di dalam file excel masih dalam bentuk mentah. Proses *preprocessing* perlu dilakukan agar data dapat diproses oleh sistem.

	judul	abstrak
0	rancang bangun aplikasi portal lomba berbasis ...	suatu kemenangan pada kompetisi diraih untuk m...
1	rancang bangun sistem informasi enterprise ma...	pada era digital ini sekolah tidak hanya berfu...
2	perencanaan arsitektur teknologi informasi dan...	saat ini implementasi sistem informasi pada ru...
3	perancangan enterprise architecture dengan men...	kementerian agama merupakan salah satu kemente...
4	perencanaan arsitektur entreprise menggunakan...	badan pengelolaan keuangan dan pajak daerah me...

Gambar 7 *Dataset* setelah *Case folding*

Gambar 7 menampilkan *Dataset* sesudah dilakukan proses *case folding*. Tabel di atas terdiri dari kolom judul dan abstrak. Sebelumnya terdapat kolom penulis, jurusan, dan *link*. Ketiga kolom tersebut tidak mengalami *preprocessing* karena tidak mempengaruhi bobot pencarian nantinya. *Dataset* kini hanya terdiri dari huruf kecil.

	judul	abstrak
0	[rancang, bangun, aplikasi, portal, lomba, ber...	[suatu, kemenangan, pada, kompetisi, diraih, u...
1	[rancang, bangun, sistem, informasi, enterpris...	[pada, era, digital, ini, sekolah, tidak, hany...
2	[perencanaan, arsitektur, teknologi, informasi...	[saat, ini, implementasi, sistem, informasi, p...
3	[perancangan, enterprise, architecture, dengan...	[kementerian, agama, merupakan, salah, satu, k...
4	[perencanaan, arsitektur, entreprise, mengguna...	[badan, pengelolaan, keuangan, dan, pajak, dae...

Gambar 8 *Dataset* Setelah *Tokenizing*

Gambar 8 menunjukkan *Dataset* setelah melalui proses *tokenizing*. Sebelumnya tabel judul dan abstrak berisi teks. Setelah mengalami proses *tokenizing*, setiap kata pada teks tersebut dipisah dan disimpan dalam bentuk array.

	judul	abstrak
0	[rancang, bangun, aplikasi, portal, lomba, web...	[kemenangan, kompetisi, diraih, meningkatkan, ...
1	[rancang, bangun, sistem, informasi, enterpris...	[era, digital, sekolah, berfungsi, penyampaian...
2	[perencanaan, arsitektur, teknologi, informasi...	[implementasi, sistem, informasi, rumah, sakit...
3	[perancangan, enterprise, architecture, framew...	[kementerian, agama, salah, kementerian, pemer...
4	[perencanaan, arsitektur, entreprise, togaf, a...	[badan, pengelolaan, keuangan, pajak, daerah, ...

Gambar 9 *Dataset* setelah *Filtering*

Hasil proses *filtering* dapat dilihat pada gambar 9. Pada gambar 9, teks pada kolom judul dan abstrak mengalami perubahan. Sejumlah kata yang awalnya terdapat pada kolom judul dan abstrak kini telah hilang.

	judul	abstrak
0	[rancang, bangun, aplikasi, portal, lomba, web...	[menang, kompetisi, raih, tingkat, mampu, tahu...
1	[rancang, bangun, sistem, informasi, enterpris...	[era, digital, sekolah, fungsi, sampai, materi...
2	[rencana, arsitektur, teknologi, informasi, si...	[implementasi, sistem, informasi, rumah, sakit...
3	[ancang, enterprise, architecture, framework, ...	[menteri, agama, salah, menteri, perintah, ind...
4	[rencana, arsitektur, entreprise, togaf, adm, ...	[badan, kelola, uang, pajak, daerah, instansi,...

Gambar 10 *Dataset* setelah *Stemming*

Gambar 10 menampilkan *Dataset* setelah melewati proses *stemming*. Perubahan dapat dilihat pada kolom judul dan abstrak. Kata yang awalnya memiliki imbuhan kini telah berubah menjadi kata dasar

4.3. Implementasi Program

4.3.1. Content based filtering

Pada metode ini, setiap *term* pada judul dan abstrak akan dilakukan pembobotan dengan menggunakan TF-IDF. Lalu bobot tersebut akan digunakan sebagai dasar penghitungan kemiripan *query* dengan dokumen. Proses TF-IDF dibagi menjadi empat tahapan yaitu menghitung TF, DF, IDF lalu TF-IDF.

Tabel 1 Penghitungan TF, DF, dan IDF

Term	TF				DF	IDF
	Q	D1	D2	…		
sistem	1	0	1		33	0,4
rancang	0	1	1		8	1,09
bangun	0	1	1		9	1,04
aplikasi	0	1	0		16	0,79
portal	0	1	0		1	2

lomba	0	1	0		1	2
web	0	1	0		9	1,04
informasi	1	0	1		10	1
enterprise	1	0	1		2	1,69
…						

Tabel 1 menunjukkan TF dari judul. TF didapat dengan menghitung jumlah kemunculan *term* pada suatu dokumen. Lalu dilanjutkan dengan menghitung DF. DF didapat dari jumlah dokumen yang mengandung *term* tertentu. Sedangkan IDF didapat dari *invers* DF.

Tabel 2 Hasil Penghitungan TF-IDF

Term	TF-IDF			
	Q	D1	D2	…
sistem	0,4	0	0,4	
rancang	0	1,09	1,09	
bangun	0	1,04	1,04	
aplikasi	0	0,79	0	
portal	0	2	0	
lomba	0	2	0	
web	0	1,04	0	
informasi	1	0	1	
enterprise	1,69	0	1,69	
…				

Setelah TF dan IDF didapat, maka penghitungan TF-IDF dapat dilakukan dengan mengalikan TF dengan IDF. Hasil penghitungan TF-IDF dapat diliat pada tabel 2. Nilai TF-IDF inilah yang akan menjadi bobot tiap *term* untuk menghitung kemiripan suatu dokumen dengan *query*.

4.3.2. Weighted Tree Similarity

Setelah tiap *term* memiliki bobot, maka dilakukan penghitungan kemiripan dengan menggunakan *cosine similarity*. *Cosine similarity* digunakan untuk menghitung tingkat kemiripan *query* dengan judul maupun abstrak.

```
0 sim_judul =  0.0
  sim_abstrak =  0.09474708014484172

1 sim_judul =  0.3724656178696086
  sim_abstrak =  0.37926172976673794

2 sim_judul =  0.16916458657205527
  sim_abstrak =  0.08976307184009358

3 sim_judul =  0.2385428432953584
  sim_abstrak =  0.1559478156501483

4 sim_judul =  0.0
  sim_abstrak =  0.15886405202014778

5 sim_judul =  0.0
  sim_abstrak =  0.036204443977150746
```

Gambar 11 Hasil Penghitungan *Similarity* Judul dan Abstrak

Gambar 11 menunjukkan hasil perhitungan tingkat kemiripan *query* dengan judul maupun abstrak skripsi. *Query* yang digunakan ialah “Sistem Informasi Enterprise”. Dari gambar di atas, dapat disimpulkan bahwa judul pada skripsi index 0, 4, dan 5 tidak mengandung satu katapun yang terdapat pada *query*. Lalu menghitung *similarity* total dengan mengalikan nilai *similarity* dengan bobot parameter. Bobot parameter judul sebesar 0,1 dan abstrak 0,9.

```
{1: {'sim_judul': 0.3724656178696086,
  'sim_abstrak': 0.37926172976673794,
  'sim_total': 0.37722289619759913},
 3: {'sim_judul': 0.2385428432953584,
  'sim_abstrak': 0.1559478156501483,
  'sim_total': 0.18072632394371135},
 2: {'sim_judul': 0.16916458657205527,
  'sim_abstrak': 0.08976307184009358,
  'sim_total': 0.11358352625968209},
 4: {'sim_judul': 0.0,
  'sim_abstrak': 0.15886405202014778,
  'sim_total': 0.11120483641410343},
 9: {'sim_judul': 0.1004287013808925,
  'sim_abstrak': 0.10074747128460047,
  'sim_total': 0.10065184031348808},
```

Gambar 12 Hasil Penghitungan *Similarity* Total

Gambar 12 merupakan hasil perhitungan kemiripan dokumen skripsi dengan *query* dengan menggunakan *Content based filtering* dan *weighted tree similarity*. Gambar di atas menampilkan sejumlah dokumen skripsi dan diurutkan dari kemiripan tertinggi. Skripsi index 7 memiliki kemiripan yang paling dekat dengan *query*. Hal ini akan berubah sesuai dengan *query* yang dimasukkan oleh penguna.

4.4. Skenario Pengujian

Tabel 3 Hasil Pencarian dengan *Query* "Sistem Informasi Enterprise"

Hasil Pencarian 'Sistem Informasi Enterprise' dengan TF-IDF		
No	Judul Skripsi	Relevan
1	Rancang Bangun Sistem Informasi Enterprise Manajemen Sekolah Studi Kasus : Sdit Al-Hidayah Sumenep.	1
2	Perancangan Enterprise Architecture Dengan Menggunakan Framework Togaf Adm (Studi Kasus : Kantor Wilayah Kementerian Agama Provinsi Jawa Timur).	1
3	Perencanaan Arsitektur Entreprise Menggunakan Togaf Adm Pada Badan Pengelolaan Keuangan Dan Pajak Daerah (Bpkpd) Surabaya.	1
4	Rancang Bangun Sistem Informasi Akademik Sekolah Berbasis Web Menggunakan Algoritma Saw (Simple Additive Weighting) (Studi Kasus : Ma. Masyhudiyah Gresik).	1
5	Perencanaan Arsitektur Teknologi Informasi Dan Sistem Informasi Pada Pelayanan Rawat Inap Menggunakan Framework Togaf Adm Studi Kasus Rumah Sakit Jiwa Menur Surabaya.	1
6	Rancang Bangun Sistem Terintegrasi Federasi Hockey Indonesia (Fhi) Kabupaten Gresik.	1
7	Rancang Bangun Aplikasi Portal Lomba Berbasis Web Menggunakan Metode Feature Driven Development	1
8	Rancang Bangun Sistem Pelayanan Surat Laporan Kehilangan Berbasis Website (Studi Kasus Pada Polsek Taman).	1
9	Pengujian Sistem Informasi Stok Dan Penjualan Berbasis Web Menggunakan Metode Black Box Testing Dengan Teknik Equivalence Partitioning (Studi Kasus: Cv. Algani Karya Mandiri)	0
10	Rancang Bangun Sistem Aplikasi Edukasi Mengenai Zat Aditif Pada Makanan (Emzapma) Berbasis Web.	1
		9

Tabel 3 menampilkan daftar skripsi yang muncul pada sistem yang menggunakan TF-IDF dengan *query* "Sistem Informasi Enterprise". Pada tabel terdapat 9 judul skripsi yang relevan dan 1 skripsi yang tidak relevan. Selain itu, setelah dilakukan penelurusan pada *database*, terdapat 1 judul skripsi yang relevan tetapi tidak terpanggil oleh sistem. Analisa dilakukan dengan *query* lainnya dan diapat *precision* dan *recall* sebagai berikut :

Tabel 4 Hasil Uji *Precision* dan *Recall*

No	*Query*	TF	FP	FN	P	R
1	Sistem Informasi Enterprise	9	1	1	0,9	0,9
2	Analisis dan Pengujian Sistem	9	11	1	0,45	0,9
3	Sistem Informasi Geografis	3	1	0	0,75	1
4	Jaringan dan Keamanan Informasi	10	2	0	0,83	1
5	Komputasi Cerdas	4	3	6	0,57	0,4
6	Komputasi Visual	8	0	2	1	0,8
7	Data Mining	8	2	2	0,8	0,8
8	Robotika	2	2	0	0,5	1
9	Internet of Things	9	0	0	1	1
10	Aplikasi dan Gim	10	0	0	1	1
11	Solusi Sistem Informasi	6	7	2	0,46	0,46
12	Manajemen Sistem Informasi	7	3	1	0,7	0,7
RATA-RATA					0,74	0,83

Tabel 4 menunjukkan hasil pengujian sistem dengan menggunakan *query* kategori skripsi dengan pembobotan TF-IDF. Hasil *precision* menunjukan akurasi *precision* (P) sistem sebesar 74% dan *recall* (R) 83%.

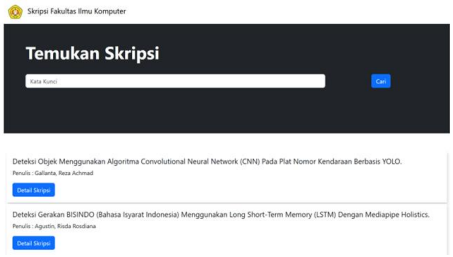
4.5. Implementasi Antar Muka

4.5.1. Tampilan Awal

Gambar 13 Tampilan Awal

Gambar 13 menunjukan sebuah tampilan yang memiliki sebuah *field input*. Pengguna dapat melakukan pencarian dengan memasukkan sebuah kata kunci atau kalimat yang relevan dengan topik skripsi pada kotak pencarian. Lalu klik tombol cari agar sistem memproses kata kunci yang diberikan oleh pengguna.

4.5.2. Tampilan Hasil Pencarian



Gambar 14 Tampilan Hasil Pencarian

Gambar 14 menunjukkan sebuah tampilan hasil pencarian. Daftar skrisi yang ditemukan oleh sistem akan ditampilkan di bawah jumbotron dalam bentuk card. Setiap card akan mewakili sebuah skripsi dengan menampilkan keterangan singkat terkait skripsi tersebut. Apabila pengguna ingin melihat informasi lengkap terakit skripsi tersebut, tombol detail skripsi akan mengarahkan pengguna ke halaman detail.

4.5.3. Tampilan Detail Skripsi

Gambar 15 Tampilan Detail Skripsi

Gambar 15 menunjukkan sebuah tampilan detail skripsi. Halaman detail skripsi menampilkan sejumlah informasi terkait skripsi. Pada bagian atas terdapat judul skripsi. Di bawah judul terdapat penulis, jurusan, dan abstrak skripsi. Pada bagian bawah terdapat sebuah slider yang berisi skripsi lainnya yang relevan dengan skripsi yang ditampilkan. *Slider* ini bertujuan untuk memberikan sebuah referensi skripsi yang berkaitan dengan topik skripsi yang dicari oleh pengguna. Selain itu di atas slider, terdapat tombol download skripsi. Tombol ini akan mengarahkan pengguna ke website *repository* UPN "Veteran" Jawa Timur sehingga pengguna dapat mendownload langsung file skripsi pada website resmi UPN.

5. KESIMPULAN

a. Penelitian ini telah berhasil merancang dan mengimplementasikan sistem pencarian hasil penelitian skripsi dengan menggunakan *Weighted tree similarity* dan *content based filtering*. Pembuatan sistem dimulai dari text *preprocessing* lalu dilakukan pembobotan kata dengan TF-IDF. Setelah itu dilakukan pembobotan tiap parameter dan tiap kata yang ada di parameter dihitung kemiripannya dengan menggunakan *cosine similarity*. Hasil *cosine similarity* tiap parameter dihitung kambali dengan bobot parameter masing-masing dan dijumlahkan. Maka didapat daftar data yang memiliki kemiripan dengan *query*. Langkah terakhir dilakukan pengujian dengan menggunakan *precision* dan *recall*. Berdasarkan hal tersebut terdapat beberapa hal yang dapat disimpulkan sebagai berikut :
b. Hasil pengujian menunjukkan bahwa sistem memiliki tingkat akurasi yang cukup baik dengan nilai *precision* 74% dan *recall* 83%

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Tuhan Yang Maha Esa atas penyertaan dan rahmatNya, penulis dapat menyelesaikan penelitian ini. Terima kasih kepada dosen pembimbing, keluarga, dan teman atas doa, bimbingan, dan dukungan serta motivasi. Penulis menyadari bahwa penelitian ini masih

jauh dari kata sempurna dan berharap mendapat kritik yang membangun. Semoga penelitian ini bermanfaat dan dapat menginspirasi penelitian selanjutnya.

DAFTAR PUSTAKA

[1] H. J. Permana and A. T. Wibowo, "Movie Recommendation System Based on Synopsis Using Content-Based Filtering with TF-IDF and Cosine Similarity," International Journal on Information and Communication Technology, vol. 9, no. 2, pp. 1-14, Dec. 2023.

[2] Nurdin, Rizal, and Rizwan, "Pendeteksian Dokumen Plagiarisme Dengan Menggunakan Metode Weight Tree," Jurnal Telematika, vol. 1, no. 1, pp. 31-45, Feb. 2019.

[3] M. Alkaff, H. Khatimi, and A. Eriady, "Sistem Rekomendasi Buku Menggunakan Weighted Tree Similarity dan Content Based Filtering," Matrik: Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer, vol. 20, no. 1, pp. 193-202, Jul. 2020.

[4] Y. Faqih, Y. Rahmanto, A. A. Aldino, and B. Waluyo, "Penerapan String Matching Menggunakan Algoritma Boyer-Moore Pada Pengembangan Sistem Pencarian Buku Online," Bulletin of Computer Science Research, vol. 2, no. 3, pp. 100-106, Aug. 2022.

[5] A. Saeroji, R. Andriyati, And Muhsin, "Analisis Efektivitas Aplikasi E-Arsip Sebagai Media Temu Kembali Informasi," Efisiensi: Kajian Ilmu Administrasi, Vol. 18, No. 1, Pp. 1-14, Feb. 2021.

[6] M. Alfyando, F. T. Anggraeny, And A. N. Sihananto, "Perbandingan Algoritma Random Forest Dan Logistic Regression Untuk Analisis Sentimen Ulasan Aplikasi Tumbuh Kembang Anak Di Play Store," Jurnal Sistem Informasi Dan Ilmu Komputer, Vol. 2, No. 1, Pp. 77-86, Feb. 2024.

[7] A. Sabrani, I. G. P. W. Wedashwara, And F. Bimantoro, "Metode Multinomial Naïve Bayes Untuk Klasifikasi Artikel Online Tentang Gempa Di Indonesia," Jurnal Teknologi Informasi, Komputer, Dan Aplikasinya, Vol. 2, No. 1, Pp. 89-100, Mar. 2020.

[8] H. Sujadi, S. Fajar, And C. Ron, "Analisis Sentimen Pengguna Media Sosial Twitter Terhadap Wabah Covid-19dengan Metode Naive Bayes Classifier Dan Support Vector Machine," Infotech Journal, Vol. 8, No. 1, Pp. 22-27, Jun. 2022.

[9] P. Lestari, "Sistem Rekomendasi untuk Maksimalisasi Industri Film dengan Metode Demographic Filtering dan Content Based Filtering," Jurnal Ilmu Komputer dan Informatika (JIKI), vol. 4, no. 1, pp. 1-10, Jun. 2024.

[10] Supiyanto and Sriyono, "Metode Cosine Similarity Untuk Mendeteksi Kemiripan Pada Dokumen Teks," SAINS Jurnal MIPA dan Pengajarannya, vol. 1, no. 1, pp. 1-7, Jan. 2023.

[11] D. P. Wijaya, L. D. Murti, and M. R. Rachman, "Recall dan Precision pada Online Public Access Catalog (OPAC) Dinas Arsip dan Perpustakaan Kota Bandung," VISI PUSTAKA, vol. 24, no. 1, pp. 81-91, Apr. 2022.

[12] Putri, D. D., Nama, G. F., & Sulistiono, W. E. (2022). Analisis Sentimen Kinerja Dewan Perwakilan Rakyat (DPR) Pada Twitter Menggunakan Metode Naive Bayes Classifier. Jurnal Informatika dan Teknik Elektro Terapan, 10(1).