

Sistem Rekomendasi Referensi Jurnal Ilmiah dengan Metode Frequent Itemset

Dini Nurmalasari^{1*}, Warnia Nengsih^{2**}

* Teknik Komputer, Politeknik Caltex Riau

** Sistem Informasi, Politeknik Caltex Riau

dini@pcr.ac.id¹, warnia@pcr.ac.id²

Article Info

Article history:

Received 2021-09-07

Revised 2021-11-08

Accepted 2021-12-04

Keyword:

*Frequent Itemset Mining,
Publikasi Ilmiah,
Reduksi Dimensi.*

ABSTRACT

Salah satu tahapan dalam penelitian yang harus dilakukan adalah mencari referensi jurnal yang relevan dengan penelitian yang akan dikerjakan. Dalam melakukan pencarian referensi, biasanya peneliti memasukan kata kunci atau keyword yang sesuai dengan tema yang akan diteliti. Kata kunci tersebut biasanya terdiri dari satu atau dua kata, sehingga hasil yang didapatkan kurang sesuai dengan pencarian yang diinginkan, atau diperlukan studi literatur yang cukup banyak untuk mencukupi referensi keseluruhan penelitian. Publikasi ilmiah tentunya memiliki dimensi kata yang besar, dimensi yang besar ini mempengaruhi kecepatan pencarian dan alokasi ruang penyimpanan yang harus disediakan. Salah satu cara untuk mengatasi dimensi kata yang besar adalah melakukan proses reduksi. Dalam perspektif text mining, reduksi dimensi merupakan teknik yang diterapkan untuk memperoleh informasi berdasarkan masukan dari datasets. Pencarian frequent itemset dapat mengatasi permasalahan reduksi dimensi. Maka pada penelitian ini akan dilakukan pencarian referensi yang relevan dengan menggunakan masukan berupa dokumen teks publikasi ilmiah atau jurnal ilmiah. Dokumen teks yang dijadikan masukan, akan dilakukan ekstraksi kemudian akan dicari tingkat kemiripan dengan dokumen lain dalam database. Keluaran dari system ini berupa daftar dokumen yang relevan dengan masukan dokumen teks beserta persentase kemiripannya. Melalui system rekomendasi referensi karya ilmiah yang dibuat dengan menerapkan metode frequent itemset mining (FIM), akan dilakukan analisis keberhasilan reduksi dimensi yang dapat mempengaruhi akurasi hasil. Dari pengujian yang telah dilakukan diperoleh bahwa FIM berhasil mereduksi dimensi fitur kata dengan rata-rata sebesar 80,17% menggunakan minimum support 0,1. Sehingga dapat disimpulkan bahwa metode FIM efektif dalam melakukan reduksi dimensi sehingga menjadi salah satu factor yang mempengaruhi akurasi hasil.



This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.

I. PENDAHULUAN

Proses pencarian referensi karya ilmiah yang relevan biasanya diaplikasikan dalam bentuk *search engine* yang meminta pengguna untuk menginputkan *keyword*. *Search engine* pada umumnya belum mendukung pencarian referensi dengan inputan berupa file dokumen. Pada penelitian ini, akan dibangun sistem yang menerima file dokumen sebagai input dan memberikan output berupa data referensi karya ilmiah yang memiliki keterkaitan konteks dengan file

dokumen yang diinputkan. File dokumen karya ilmiah tentunya memiliki dimensi kata yang besar, yang akan mempengaruhi kecepatan proses pencarian dan kapasitas penyimpanan.

Salah satu cara mengatasi dimensi kata yang besar adalah melakukan proses reduksi. Menurut [1] dan [2] dalam perspektif text *mining*, reduksi dimensi merupakan pengurangan dimensi suatu dataset dengan pertimbangan bahwa informasi-informasi penting tetap dipertahankan. Salah satu metode yang dapat digunakan untuk mereduksi

dimensi adalah dengan menggunakan *frequent itemset*. Akan tetapi, metode pengurangan dimensi seperti *preprocessing text* yaitu *stop words*, *stemming words* tetap dilakukan terlebih dahulu sebelum melakukan pencarian *frequent itemset*. Menurut [2] kemampuan FIM lebih efisien dibanding algoritma lainnya karena menggunakan struktur data yang padat.

II. STUDI LITERATUR

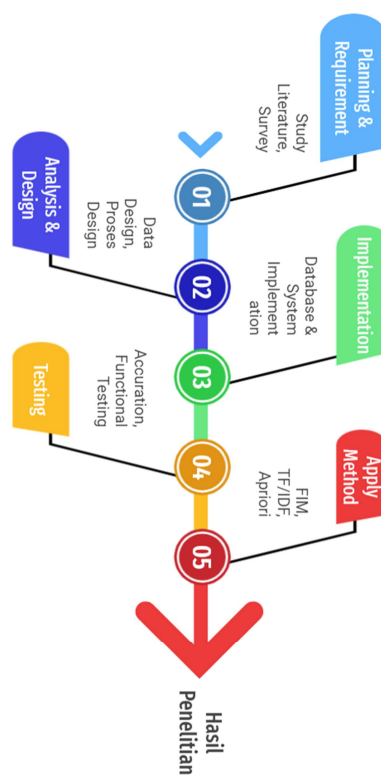
Dalam penelitian yang dilakukan oleh [3] dengan judul *information retrieval document classification with K=Nearest Neighbor*, yang bertujuan melakukan klasifikasi dokumen dengan menggunakan *cosine similarity* dan *KNN*. Dari 160 dokumen yang diklasifikasikan, dengan melakukan 20 dokumen sebagai data testing, berhasil mengklasifikasikan dokumen tersebut pada kelompok yang sesuai. Sementara pada tahun 2020, pada penelitian [4] mengenai *Automatic Document Classification* dengan menerapkan metode *Support Vector Machine*. Dalam penelitian tersebut dihasilkan kesimpulan bahwa *feature selection* tidak hanya berhasil digunakan untuk mereduksi dimensi, namun juga mengurangi waktu dalam proses klasifikasi.

Sementara [2] dalam jurnalnya *Document Classification with frequent itemset mining*, melakukan penelitian pada beberapa dokumen seperti email, dokumen gambar, dokumen teks dengan menggunakan algoritma clustering yaitu *frequent itemset* dengan pendekatan ontology untuk menyelesaikan permasalahan semantic atau makna kata dari dokumen, tidak hanya kata dalam bentuk sintaktik. Penelitian lainnya telah mengimplementasikan *frequent itemset mining* pada beberapa dokumen yaitu diantaranya [4] pada laporan keuangan OJK untuk mempermudah pembacaan data laporan, [5] diimplementasikan padadatan set dari *Kaggle* untuk mengelompokkan *food review* dari pengguna, serta [6] melakukan pengelompokan dokumen berdasarkan abstrak penelitian.

Semua penelitian yang sudah disebutkan tersebut, proses pengelompokan dokumen dilakukan berdasarkan fitur seleksi yang diambil dengan *frequent itemset mining*, untuk kemudian dikelompokkan. Sementara pada penelitian ini metode *Frequent Itemset Mining* (FIM) diimplementasikan untuk menemukan kumpulan kata yang sering muncul pada publikasi ilmiah, untuk kemudian digunakan sebagai kata kunci utama dalam melakukan pencarian data yang relevan. Kemudian pencarian referensi karya ilmiah yang relevan atau memiliki keterkaitan konteks dilanjutkan dengan melakukan perhitungan *Cosine Similarity* untuk menghitung persentase tingkat keterkaitan konteks antara file dokumen yang diinputkan dengan sejumlah karya ilmiah yang terdapat pada system yang relevan untuk dijadikan sebagai referensi. FIM diaplikasikan menggunakan algoritma Apriori. Sehingga kontribusi penelitian ini adalah penggunaan kata kunci dalam melakukan pencarian berasal dari dokumen, sehingga diharapkan dapat menghasilkan rekomendasi referensi jurnal yang lebih akurat.

III. METODE PENELITIAN

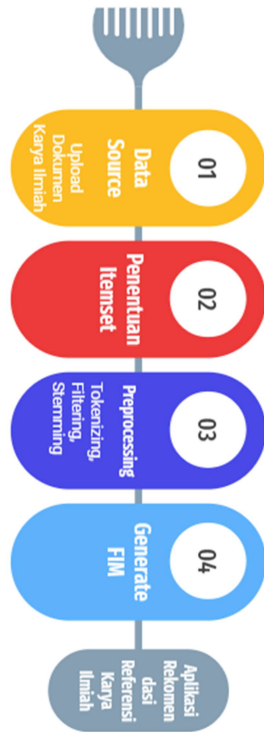
Secara umum tahapan dari penelitian ini adalah tahapan *requirement* untuk menentukan bisnis proses dari system yang akan dibuat, analisis dan perancangan untuk menentukan sumber data dan komponen yang dibutuhkan, serta melakukan perancangan proses dan perancangan data, implementasi yaitu tahapan membangun system, serta pengujian yaitu tahapan pengujian system. Pada tahapan 6 pengujian system akan dilakukan pengujian kinerja metode FIM, TF-IDF dan *Cosine Similarity* [7] [8]. Analisis pada metode FIM dilakukan untuk melihat kinerja dalam mereduksi dimensi, sedangkan TF-IDF akan dilakukan analisis dan pengujian untuk melihat kinerja dalam menghasilkan *item-set*, dan *Cosine Similarity* untuk melihat akurasi keterkaitan karya ilmiah yang relevan.



Gambar 1 Fishbone Metode Penelitian

Sedangkan secara khusus proses rekomendasi referensi karya ilmiah diawali dengan proses input berupa file dokumen teks ke dalam system kemudian dilakukan proses penentuan *itemset*. Lalu dilanjutkan dengan proses *text mining* yang terdiri dari beberapa proses *preprocessing* yaitu *tokenizing*, *filtering*, dan *stemming*. *Text mining* digunakan agar menghasilkan kumpulan kata yang dapat digunakan untuk proses pencarian kata yang sering muncul (*frequent*) pada dokumen karya ilmiah. Proses ini juga akan membentuk dimensi yang optimal setelah dilakukan proses reduksi

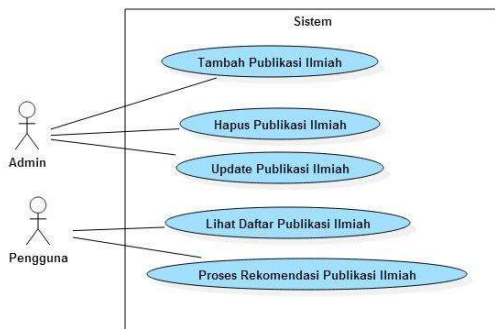
dimensi sesuai kebutuhan. Proses selanjutnya adalah mencari kata yang sering muncul (*frequent itemset*) yang akan digunakan sebagai fitur kata menggunakan FIM. Setelah fitur kata ditemukan, maka dilakukan perhitungan menggunakan algoritma *TF-IDF* dan *Cosine Similarity* untuk pencarian persentase rekomendasi publikasi ilmiah. Gambar 2 berikut ini merupakan alur dari proses system rekomendasi referensi karya ilmiah.



Gambar 2 Langkah-Langkah Aplikasi

IV. HASIL DAN PEMBAHASAN

Sistem rekomendasi publikasi ilmiah akan dibangun menggunakan pendekatan beorientasi objek.



Gambar 3 Use Case Aplikasi

Berikut ini adalah langkah-langkah dalam melakukan pencarian rekomendasi referensi yang relevan pada tahapan Generate FIM [10] :

1. Mencari bobot kemunculan kata (term) terhadap jurnal
2. Menghitung kemiripan vector query dalam setiap jurnal yang ada
3. Menghitung panjang vector dengan mengkuadratkan bobot setiap term dalam setiap jurnal
4. Menerapkan rumus cosine similarity
5. Menghitung relevansi antar query
6. Mengurutkan hasil

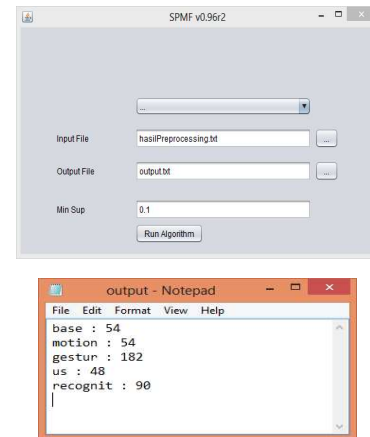
Table 1 merupakan rancangan struktur data yang digunakan untuk penyimpanan data hasil dari proses ekstraksi dokumen karya ilmiah.

TABLE I
STRUKTUR DATA PENYIMPANAN

Nama Field	Type	Keterangan
Kode_PubIlmiah	Int(10)	Primary key
Judul_PubIlmiah	Varchar(100)	
Penulis_PubIlmiah	Varchar(20)	
Tahun	Int(5)	
File_PubIlmiah	Varchar(100)	
Persentase	Double(22)	

A. Pembahasan Metode Frequent Itemset Mining

Metode *Frequent Itemset Mining* diimplementasikan dalam menemukan kata yang *frequent*. Kata yang *frequent* diperoleh dari hasil perhitungan *minimum support* [11]. *User interface* untuk menemukan kata yang *frequent* adalah sebagai berikut:



Gambar 4 User Interface Penemuan Kata Frequent

Gambar 4 merupakan *user interface* penemuan kata *frequent*. *User* diminta untuk menyediakan *file input* yang berisi publikasi ilmiah yang telah *dipreprocessing*. Kemudian

user juga diminta untuk menyediakan *file output* untuk menyimpan hasil kata *frequent* yang telah ditemukan. Lalu, *user* menginputkan nilai *minimum support* antara 0-1. Setelah *user* menekan tombol *Run Algorithm*, maka akan muncul hasil *output* berupa *file .txt*, *output* berupa informasi jumlah baris yang merupakan jumlah *itemset*, dan jumlah fitur kata yang terpilih.

B. Pembahasan Perhitungan TF-IDF dan Cosine Similarity

TF-IDF dan *Cosine Similarity* diimplementasikan dalam perhitungan persentase rekomendasi publikasi ilmiah. TF-IDF digunakan untuk menghitung bobot setiap kata pada dokumen. Pada gambar 5 berikut merupakan potongan perhitungan TF-IDF dalam implementasi program.

```

1. public double tfCalculator(String[]
   totalterms, String termToCheck) {
2.     double count = 0;
3.     for (String s : totalterms)
4.     {
5.         if
6.         (s.equalsIgnoreCase(termToCheck)) {
7.             count++ }
8.         System.out.print(count + "
9.         ");
10.        return count / totalterms.length;}
11. public double idfCalculator(List<String[]>
12. allTerms, String termToCheck) {
13.     double count = 0;
14.     for (String[] ss : allTerms) {
15.         for (String s : ss) {
16.             if
17.             (s.equalsIgnoreCase(termToCheck)) {
18.                 count++; }
19.         }
20.         System.out.println(count+"
21.         "+
22.         (Math.log10(allTerms.size() / count) ));
23.         return Math.log10(allTerms.size() /
24.         count); }

```

C. Pembahasan Perhitungan Cosine Similarity

Cosine Similarity digunakan dalam perhitungan nilai *cosinus* sudut antara vektor *query* dengan vektor publikasi ilmiah yang ada pada sistem. Perhitungan nilai *cosine similarity* dan *User Interface* hasil perhitungan TF-IDF dan *Cosine Similarity* adalah sebagai berikut.

```

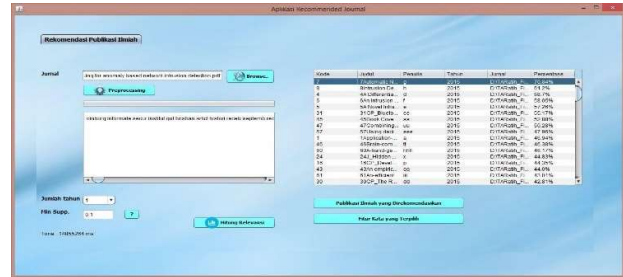
1. public double cosineSimilarity(double[]
   docVector1, double[] docVector2) {
2.     double dotProduct = 0.0;
3.     double magnitude1 = 0.0;
4.     double magnitude2 = 0.0;
5.     double cosineSimilarity = 0.0;
6.     for (int i = 0; i < docVector1.length;
7.         i++) {
8.         dotProduct += docVector1[i] *
9.         docVector2[i]; //a.b
10.        magnitude1 += Math.pow(docVector1[i],
11.        2); //(a^2)
12.        magnitude2 += Math.pow(docVector2[i],
13.        2); //(b^2)

```

```

10.    magnitude1 =
   Math.sqrt(magnitude1); //sqrt(a^2)
11.    magnitude2 =
   Math.sqrt(magnitude2); //sqrt(b^2)
12.    if (magnitude1 != 0.0 | magnitude2
   != 0.0) {
13.        cosineSimilarity = dotProduct
   / (magnitude1 * magnitude2) * 100;
14.    } else {
15.        return 0.0; }
16.    return cosineSimilarity ;}

```



Gambar 5 User Interface TF-IDF dan Cosine Simmilarity

Gambar 5 merupakan potongan program untuk perhitungan *cosine similarity*, dimana pada program tersebut dilakukan inisialisasi variable, kemudian *looping* untuk perhitungan *dotproduct* yaitu perkalian vektor dokumen dan *magnitude* atau Panjang vektor dokumen. Setelah itu dilakukan perhitungan akar Panjang vektor dokumen kemudian perhitungan *cosine similarity*.

Gambar 7 merupakan *user interface* hasil perhitungan TF-IDF dan *Cosine Similarity*. Terdapat tabel daftar nama jurnal beserta persentase rekomendasi. Jurnal dengan persentase tertinggi adalah jurnal yang memiliki tingkat kemiripan paling tinggi diantara jurnal yang ada pada sistem. Nilai persentase dihasilkan dari perhitungan TF-IDF dan *Cosine Similarity* antara publikasi ilmiah yang diinputkan *user* dengan publikasi ilmiah yang tersimpan pada sistem. Nilai persentase dibulatkan dua angka dibelakang koma.

D. Pengujian Reduksi Publikasi Ilmiah

Pengujian reduksi publikasi ilmiah bertujuan untuk mengetahui maksimal dimensi kata yang dapat direduksi oleh *Frequent Itemset Mining*. Nilai *minimum support* sangat mempengaruhi reduksi. Semakin besar nilai *minimum support* maka semakin sedikit fitur kata yang terpilih. Hal tersebut menyebabkan reduksi dimensi kata semakin besar. Sebelum melakukan pengujian reduksi publikasi ilmiah, akan dilakukan pengujian untuk mengetahui maksimal nilai *minimum support* yang dapat digunakan dalam pemilihan fitur kata. Tujuannya agar mengetahui nilai *minimum support* terbesar yang dapat mereduksi dimensi publikasi ilmiah secara maksimal. Tabel pengujian maksimal nilai *minimum support* terlampir pada Lampiran C, tabel C.1.

Tabel 2 merupakan tabel hasil pengujian maksimal *minimum support* yang dapat digunakan sebagai nilai yang

mempengaruhi besarnya reduksi dimensi publikasi ilmiah. Pada tabel terdapat jumlah *itemset* yang merupakan jumlah kalimat pada *publikasi* ilmiah. Jumlah fitur kata yang terpilih adalah banyak kata yang terpilih dan memenuhi jumlah *minimum* kata yang sama. Misalnya publikasi ilmiah P1, fitur kata yang terpilih dari jumlah *itemset* sebanyak 329 adalah 12 kata dengan *minimum support* 0,1. Jika menggunakan *minimum support* 0,2 maka tidak ada fitur kata yang terpilih untuk publikasi ilmiah P1.

Dari tabel 2 dapat diperoleh rata-rata maksimal *minimum support* adalah sebesar 0,118. Nilai 0,118 kemudian dibulatkan menjadi 0,1 yang akan digunakan sebagai nilai *minimum support* untuk pengujian reduksi publikasi ilmiah. Tabel hasil pengujian reduksi publikasi ilmiah terlampir pada tabel 2

Tabel 2 merupakan tabel hasil pengujian reduksi publikasi ilmiah menggunakan nilai *minimum support* 0,1. Terdapat jumlah seluruh kata yaitu jumlah keseluruhan kata yang telah melalui proses *preprocessing*. Jumlah kata *frequent* adalah banyak kata yang terpilih dan memenuhi jumlah *minimum* kata yang sama. Total kata *frequent* adalah total keseluruhan kata yang *frequent*. Persentase reduksi adalah persentase pengurangan dimensi publikasi ilmiah, diperoleh dari persentase total kata yang tidak *frequent* terhadap jumlah seluruh kata.

TABLE II
HASIL PENGUJIAN REDUKSI PUBLIKASI ILMIAH

No	Publikasi Ilmiah	Jumlah Seluruh Kata	Jumlah Kata Frequent	Total Kata Frequent	Persentase Reduksi
1	P1	2694	12	544	79,81%
2	P2	6175	2	246	96,02%
3	P3	2220	9	342	84,59%
4	P4	3644	8	563	84,55%
5	P5	4141	6	424	89,76%
6	P6	9188	6	871	90,52%
7	P7	5703	6	637	88,83%
8	P8	4663	6	586	87,43%
9	P9	4141	6	424	89,76%
10	P10	4235	5	392	90,74%
11	P11	2166	8	379	82,50%
12	P12	3011	10	445	85,22%
13	P13	6764	4	641	90,52%
14	P14	2558	17	796	68,88%
15	P15	7467	8	1422	80,96%
16	P16	2462	6	249	89,89%
17	P17	7402	2	442	94,03%
18	P18	2725	6	377	86,17%
19	P19	4489	12	1150	74,38%
20	P20	1581	6	326	79,38%
21	P21	2951	13	974	66,99%
22	P22	2489	4	206	91,72%
23	P23	1482	7	408	72,47%
24	P24	4080	5	425	89,58%
25	P25	1817	4	234	87,12%
26	P26	7806	5	991	87,30%
27	P27	5827	3	539	90,75%
28	P28	1920	15	449	76,61%
29	P29	1717	10	299	82,59%
30	P30	1530	13	383	74,57%
31	P31	1213	8	168	86,15%

E. Pengujian Pemilihan Fitur Kata

Pemilihan fitur kata pada sistem menggunakan metode *Frequent Itemset Mining*. Pengujian pemilihan fitur kata bertujuan untuk menguji kebenaran fitur kata yang terpilih. Pengujian ini membandingkan hasil *tools Rapid Miner* dengan hasil sistem. Pengujian ini menggunakan beberapa publikasi ilmiah yang dianggap mampu merepresentasikan hasil pemilihan fitur kata. Alasan tidak menguji keseluruhan publikasi ilmiah yang berjumlah 100 disebabkan proses pengujian yang cukup panjang. Publikasi ilmiah yang digunakan sebanyak 7 buah dianggap sudah mewakili proses pengujian karena menggunakan nilai *minimum support* yang bervariasi. Tabel pengujian pemilihan fitur kata dengan *Rapid Miner* dan sistem terlampir pada tabel 3.

Tabel 3 merupakan tabel pengujian pemilihan fitur kata dengan *Rapid Miner* dan sistem. Pada tabel terdapat jumlah *itemset* merupakan jumlah data transaksi atau jumlah baris pada publikasi ilmiah. Jumlah minimal kata yang *frequent* adalah jumlah minimal yang harus dipenuhi oleh fitur kata yang terpilih. Kolom kata adalah daftar fitur kata yang terpilih. Kolom jumlah adalah jumlah setiap fitur kata pada dokumen.

TABLE I
HASIL PENGUJIAN PEMILIHAN FITUR KATA

Publikasi Ilmiah	Minimum Support	Jumlah Itemset	Jumlah Minimal Kata yang Frequent	Fitur Kata dengan Rapid Miner		Fitur Kata dengan Sistem	
				Kata	Jumlah	Kata	Jumlah
P34	0.07	1159	82	docum	129	docum	129
				measur	103	measur	103
				similar	104	similar	104
				data	107	data	107
				cluster	85	cluster	85
				featur	85	featur	85
P35	0.2	365	73	paper	121	paper	121
				topic	117	topic	117
				research	92	research	92
				set	79	set	79
P36	0.1	309	31	categorize	113	categorize	113
				comput	72	comput	72
				paper	65	paper	65
				docum	57	docum	57
				scienc	45	scienc	45
				agent	55	agent	55
				acm	45	acm	45
				us	33	us	33
P37	0.1	459	46	base	31	base	31
				similar	138	similar	138
				docum	113	docum	113

				seman t	76	seman t	76
				word	52	word	52
				vector	48	vector	48
				cluster	51	cluster	51
P38	0.2	217	44	search	84	search	84
				engin	65	engin	65
				docum	55	docum	55
				retriev	53	retriev	53

Berdasarkan hasil dari tabel 3 diperoleh kesamaan hasil perhitungan *cosine similarity* secara manual dengan menggunakan perangkat *rapid miner* dan perhitungan dengan system yang dibangun. Sehingga dapat disimpulkan bahwa system yang dibangun telah menerapkan algoritma yang benar.

V. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, dapat disimpulkan bahwa penerapan *Frequent Itemset Mining (FIM)* berhasil melakukan pemilihan fitur kata pada publikasi ilmiah yang dapat mereduksi rata-rata 80,17% menggunakan minimum support 0,1. Rekomendasi publikasi ilmiah menggunakan FIM sebagai pemilihan fitur kata terbukti dapat menghemat waktu proses TF-IDF dan Cosine Similarity sebesar 69,7 menit menggunakan 100 publikasi ilmiah uji. Proses reduksi dimensi terbukti mempengaruhi akurasi hasil relevansi.

UCAPAN TERIMA KASIH

Penulis mengucapkan terimakasih kepada institusi Politeknik Caltex Riau yang telah mendanai penelitian ini.

DAFTAR PUSTAKA

- [1] I. M. D. P. Asana, I. K. A. G. Wiguna, K. J. Atmaja, and I. P. A. Sanjaya, "FP-Growth Implementation in Frequent Itemset Mining for Consumer Shopping Pattern Analysis Application," *J. Mantik*, vol. 4, no. 3, pp. 2063–2070, 2020.
- [2] M. Shahpoori and M. Alimoradi, "Document Classification with Frequent Itemset Mining".
- [3] A. Sukma, B. Zaman, and E. Purwanti, "Information Retrieval Document Classification with K-Nearest Neighbor," *Rec. Libr. J.*, vol. 1, no. 2, pp. 129–138, 2015.
- [4] C. Goller, J. Löning, T. Will, and W. Wolff, "Automatic Document Classification-A thorough Evaluation of various Methods.," *ISI*, vol. 2000, no. 2, pp. 145–162, 2000.
- [5] D. N. R. P. S. Audia Nurul Asyfa, "Identifikasi Kinerja Perusahaan Berdasarkan Laporan Keuangan Menggunakan Algoritma K-NN," *Jurnal Aksara Komputer Terapan*, p. 8, 2016.
- [6] A. N. Asyfa, D. Nurmalasari, and R. P. Sari, "Identifikasi Kinerja Perusahaan Berdasarkan Laporan Keuangan Menggunakan Algoritma K-NN," *J. Aksara Komput. Terap.*, vol. 5, no. 1, 2016.
- [7] M. Ainiyah, D. Nurmalasari, and W. Nengsih, "Visualisasi Data Teks Food Reviews Menggunakan Frequent Itemset Mining," *J. Aksara Komput. Terap.*, vol. 6, no. 2, 2017.
- [8] L. Tanjaya, A. Wibowo, and D. Nurmalasari, "Sistem Pengelompokan E-Journal Berdasarkan Abstrak Menggunakan Text Mining dan K-Means Clustering," *J. Aksara Komput. Terap.*, vol. 5, no. 1, 2016.
- [9] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011.
- [10] R. Feldman and J. Sanger, *The text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge university press, 2007.
- [11] E. Muningsih, H. M. Nur, F. F. D. Imaniawan, V. R. Handayani, and F. Endiarto, "Comparative Analysis on Dimension Reduction Algorithm of Principal Component Analysis and Singular Value Decomposition for Clustering," in *Journal of Physics: Conference Series*, 2020, vol. 1641, no. 1, p. 012101.