



PENERAPAN VECTOR SPACE MODEL (VSM) PADA SISTEM PENCARIAN ARTIKEL ARKEOLOGI

Wa Ode Wahyuni Makmun^{*1}, Ika Purwanti Ningrum², Adha Mashur Sajiah³

^{1,2,3} Jurusan Teknik Informatika, Fakultas Teknik, Universitas Halu Oleo, Kendari

e-mail: ^{*1}wahyunimakmun1@gmail.com, ²ika.purwanti.n@uho.ac.id, ³adha.m.sajiah@uho.ac.id

Abstrak

Artikel ilmiah adalah tulisan yang berisi laporan sistematis mengenai hasil kajian atau hasil penelitian yang dilakukan oleh dosen, mahasiswa, peneliti dan ilmuwan. Jurusan Arkeologi Universitas Halu Oleo telah memiliki koleksi artikel Arkeologi. Namun, proses pencarian dokumen membutuhkan waktu yang relatif lama, karena banyaknya koleksi artikel yang dimiliki. Berdasarkan hal tersebut dibutuhkan suatu sistem pencarian yang dapat membantu menemukan dokumen yang dibutuhkan. Sistem Temu Kembali Informasi (*Information Retrieval*) adalah proses menemukan materi yang biasanya dokumen dari data yang tidak terstruktur yang memenuhi kebutuhan informasi dari sebuah kumpulan koleksi yang tersimpan pada komputer. Salah satu metode sistem temu kembali informasi yang cocok digunakan untuk menemukan dokumen artikel arkeologi yaitu *Vector Space Model* karena efektif dan memiliki ketepatan dalam mengurutkan dokumen berdasarkan *query* yang dicari. Dari hasil pengujian *precision* dan *recall* yang dilakukan pada *query* yang terdiri dari 1 kata, 2 kata, 3 kata, 4 kata, dan 5 kata. *Query* yang memiliki nilai rata-rata *recall* tertinggi yaitu *query* yang terdiri dari 4 kata sebesar 95,04% dan *query* yang memiliki nilai rata-rata *precision* tertinggi yaitu *query* yang terdiri dari 1 kata sebesar 98,125%.

Kata kunci; *Vector Space Model*, Perangkingan, Artikel Arkeologi, Sistem Pencarian

Abstract

Scientific articles are writings that contain systematic reports on the results of studies or research conducted by lecturers, students, researchers, and scientists. The Halu Oleo University Department of Archeology has a large collection of Archeology articles. However, the document search process takes a relatively long time, due to the large collection of articles we have. Based on this, a search system is needed that can help find the required documents. Information Retrieval System is the process of finding material, which is usually a document from unstructured data that meets the information needs of a collection stored on a computer. One method of information retrieval system that is suitable for finding archaeological article documents is the Vector Space Model because it is effective and has accuracy in sorting documents based on the query being searched. From the results of the precision and recall tests carried out on queries consisting of 1 word, 2 words, 3 words, 4 words, and 5 words. Queries that have the highest average recall value are queries consisting of 4 words at 95.04% and queries that have the highest average precision value, namely queries consisting of 1 word at 98.125%.

Keywords; *Vector Space Model*, *Ranking*, *Archeology Articles*, *Search System*



1. PENDAHULUAN

Mengembangkan tridharma perguruan tinggi merupakan tanggung jawab dari suatu perguruan tinggi. Salah satu tridharma perguruan tinggi yaitu penelitian. Dalam melakukan suatu penelitian seorang peneliti perlu untuk mencari artikel ilmiah yang berkaitan untuk dipelajari maupun sebagai pendukung penelitiannya. Artikel ilmiah adalah tulisan yang berisi laporan sistematis mengenai hasil kajian atau hasil penelitian yang dilakukan oleh dosen, mahasiswa, peneliti dan ilmuwan [1].

Jurusan Arkeologi Universitas Halu Oleo telah memiliki koleksi artikel Arkeologi. Namun, koleksi artikel tersebut masih tersimpan pada komputer secara manual dengan nama file yang tidak teratur dan artikel tersebut dikelola oleh dosen yang telah diberi tanggung jawab. Dengan itu, masih banyak kendala-kendala yang dihadapi seperti apabila dosen atau mahasiswa yang membutuhkan artikel sebagai referensi, harus meminta kepada dosen yang memiliki wewenang untuk mengelolanya. Kendala lain yang dihadapi yaitu proses pencarian dokumen membutuhkan waktu yang relatif lama, karena banyaknya koleksi artikel yang dimiliki serta harus membuka satu per satu dokumen untuk mendapatkan artikel yang berkaitan dengan penelitian selanjutnya. Berdasarkan hal tersebut dibutuhkan suatu sistem pencarian yang dapat membantu menemukan dokumen yang dibutuhkan.

Sistem Temu Kembali Informasi (*Information Retrieval*) adalah proses menemukan materi yang biasanya dokumen dari data yang tidak terstruktur yang memenuhi kebutuhan informasi dari sebuah kumpulan koleksi yang tersimpan pada komputer [2].

Dalam sistem temu kembali informasi, diperlukan perankingan hasil pencarian dokumen sesuai dengan tingkat relevansinya dengan kata kunci yang dimasukkan. *Vector Space Model* (VSM) merupakan salah satu model dalam sistem temu kembali informasi yang mampu melakukan perankingan terhadap dokumen – dokumen yang relevan terhadap kata kunci yang dimasukkan. *Vector Space Model* (VSM) adalah metode untuk mengetahui tingkat kesamaan (*similarity*) *term* dengan cara pembobotan *term*. *Vector Space Model* (VSM) merepresentasikan suatu dokumen dan *query*

dalam sebuah bentuk *vector* yang memiliki jarak (*magnitude*) dan arah (*direction*) [3].

Berdasarkan latar belakang yang telah diuraikan diatas, maka penulis mengambil topik penelitian dengan judul “Penerapan *Vector Space Model* (VSM) pada Sistem Pencarian Artikel Arkeologi. (Studi Kasus: Jurusan Arkeologi Universitas Halu Oleo)”.

2. METODE PENELITIAN

2.1 *Text Preprocessing*

Menurut [3], *text preprocessing* merupakan tahapan awal pada proses merepresentasikan koleksi dokumen kedalam bentuk tertentu untuk memudahkan dan mempercepat proses pencarian dan penemuan kembali dokumen yang relevan. Terdapat beberapa tahapan pada fase ini antara lain:

- a. *Case Folding* dan *Tokenization*, *Case folding* dilakukan untuk merubah huruf besar dari setiap kata diubah menjadi huruf kecil dan menghilangkan karakter selain huruf seperti angka dan tanda baca (delimiter). Sedangkan *tokenization* yaitu memecah semua kalimat pada isi dokumen menjadi kata per kata.
- b. *Filtering*, dilakukan dengan metode *stopword* yaitu menghilangkan semua kata yaitu kata sambung, kata depan, kata ganti, dan lain-lain.
- c. *Stemming*, dilakukan dengan cara menghilangkan semua imbuhan (*affixes*) baik yang terdiri dari awalan (*prefixes*), sisipan (*infixes*), akhiran (*suffixes*) dan *confixes* (kombinasi dari awalan dan akhiran) pada kata turunan. *Stemming* digunakan untuk mengganti bentuk dari suatu kata menjadi kata dasar.

2.2 *Indexing*

Indexing merupakan sebuah proses untuk melakukan pengindeksan terhadap kumpulan dokumen yang akan disediakan sebagai informasi kepada pengguna [4]. Pengindeksan dilakukan dengan membuat *inverted index* (atau disebut juga *inverted file*). *Inverted index* merupakan konversi dari dokumen asli yang mengandung sekumpulan kata ke dalam daftar kata (*term*) yang behubungan dengan dokumen terkait dimana kata – kata tersebut muncul.

2.3 Sistem Temu Kembali Informasi

Sistem temu kembali informasi (*Information retrieval*) adalah proses menemukan materi yang biasanya dokumen dari data yang tidak terstruktur yang memenuhi kebutuhan informasi dari sebuah kumpulan koleksi yang tersimpan pada komputer [2].

2.4 Vector Space Model (VSM)

Vector Space Model (VSM) merupakan model yang digunakan untuk mengukur kemiripan antara dokumen dan *query* yang mewakili setiap dokumen dalam sebuah koleksi sebagai sebuah titik dalam ruang [5]. Dalam metode *Vector Space Model* dihitung bobot dari setiap *term* yang terdapat dalam semua dokumen dan *query* dari *user*. *Term* adalah kata atau kumpulan kata yang merupakan ekspresi verbal dari suatu pengertian. Penentuan relevansi dokumen dengan *query* dipandang sebagai pengukuran kesamaan antara vektor dokumen dengan vektor *query*.

Proses perhitungan *vector space model* (VSM) melalui tahapan perhitungan *term frequency* (tf), *inverse document frequency* (idf), *term frequency inverse document frequency* (tf – idf), jarak *query* dan dokumen, pengukuran similaritas *query document (inner product)*, dan pengukuran *Cosine Similarity* (menghitung nilai *cosine* sudut antara dua vektor).

1. Menghitung bobot dokumen dengan *term – frequency* dan *inverse document frequency* (tf – idf)

Term frequency-inverse document frequency atau biasa sering disebut tf – idf adalah metode pembobotan kata dengan menghitung nilai tf dan juga menghitung kemunculan sebuah kata pada koleksi dokumen teks secara keseluruhan. *Inverse document frequency* (idf) adalah jumlah dokumen yang mengandung sebuah *term* didasarkan pada seluruh dokumen yang ada pada data set. Terdapat beberapa cara atau metode dalam melakukan pembobotan kata pada metode tf – idf yaitu melalui skema pembobotan *query* dan dokumen. Formula yang digunakan pada *term frequency* (tf) diberikan berdasarkan jumlah kemunculan suatu kata di dalam dokumen. *Inverse dokument frequency* (idf) dihitung dengan menggunakan Persamaan (1) :

$$idf_i = \log\left(\frac{N}{df_i}\right) \quad (1)$$

Dimana, N adalah total jumlah dokumen dalam koleksi, df_i = jumlah dokumen yang berisi *term* j, dan idf_i = frekuensi kemunculan suatu istilah dalam dokumen i. Dengan demikian rumus untuk tf – idf dapat dilihat pada Persamaan (2).

$$W_{ij} = tf_{ij} * idf_i \quad (2)$$

Setelah bobot (W) masing-masing dokumen diketahui, maka dilakukan proses sorting atau pengurutan dimana semakin besar nilai W, semakin besar tingkat similaritas dokumen tersebut terhadap kata kunci, demikian sebaliknya.

2. Menghitung Jarak Query dan Dokumen

Perhitungan jarak dokumen dan jarak *query* dilakukan setelah perhitungan bobot. Jika, Q adalah vektor *query* dan D adalah vektor dokumen, merupakan 2 buah vektor dalam ruang berdimensi-n, dan θ adalah sudut yang dibentuk oleh 2 vektor tersebut. Perhitungan jarak *query* dan dokumen menggunakan Persamaan (3).

$$Q \cdot D = |Q||D| \cos \theta \quad (3)$$

Dimana $Q \cdot D$ adalah hasil perkalian dalam (*inner product*) kedua vektor, sedangkan perhitungan jarak *query* menggunakan Persamaan (4) dan perhitungan jarak dokumen menggunakan Persamaan (5).

$$|Q| = \sqrt{\sum_{j=1}^n (W_{i,j})^2} \quad (4)$$

Dengan $|Q|$ adalah jarak *query* dan $W_{i,j}$ adalah bobot *query* dokumen ke – i.

$$|D_j| = \sqrt{\sum_{i=1}^n (W_{i,j})^2} \quad (5)$$

Dengan $|D_j|$ adalah jarak dokumen dan $W_{i,j}$ adalah bobot dokumen ke – i.

3. Menghitung Similaritas Query dan Dokumen

Setelah jarak dari tiap dokumen dan *query* didapatkan, dilakukan perhitungan similaritas antara *query* dengan dokumen. Perhitungan kesamaan kedua vektor menggunakan Persamaan (6).

$$Sim(Q, D_j) = \sum_{i=1}^n (W_{iq} \cdot W_{ij}) \quad (6)$$

Dengan W_{ij} adalah bobot term dalam dokumen, W_{iq} adalah bobot *query*, dan

$\text{Sim}(Q, D_j)$ adalah similaritas antara *query* dan dokumen.

4. Menghitung Cosine Similarity

Melalui *vector space model* dan tf-idf weighting akan didapatkan representasi nilai numerik dokumen sehingga dapat dihitung kedekatan antar dokumen. Semakin dekat dua vektor didalam suatu *vector space model*, maka semakin mirip dua dokumen yang diwakili oleh dua vektor tersebut. Kemiripan antar dokumen dapat dihitung menggunakan suatu fungsi ukuran kemiripan (*similarity measure*). Ukuran ini memungkinkan perankingan dokumen sesuai dengan kemiripannya atau relevansinya terhadap *query*.

Pengukuran cosine similarity (menghitung nilai cosine sudut antara dua vektor) menggunakan Persamaan (7) dan (8).

$$\text{Sim}(Q, D_j) = \frac{Q \cdot D_j}{|Q||D_j|} = \frac{1}{|Q||D_j|} \sum_{i=1}^n (W_{iq} \cdot W_{ij}) \quad (7)$$

Dari Persamaan (7) dapat dinormalisasi menjadi:

$$\text{Sim}(Q, D_j) = \frac{Q \cdot D_j}{|Q||D_j|} = \frac{\sum_{i=1}^n (W_{iq} \cdot W_{ij})}{\sqrt{\sum_{i=1}^n (W_{iq})^2} \cdot \sqrt{\sum_{i=1}^n (W_{ij})^2}} \quad (8)$$

Perhitungan similaritas menghasilkan bobot dokumen yang mendekati nilai 1 atau menghasilkan bobot dokumen yang lebih besar dibandingkan dengan nilai yang dihasilkan dari perhitungan *inner product*.

2.5 Recall dan Precision

Dalam mengukur efektivitas sistem temu kembali informasi biasanya digunakan *recall* dan *precision*. *Recall* adalah proporsi jumlah dokumen yang dapat ditemukan kembali oleh sebuah proses pencarian di sistem IR. *Precision* adalah proporsi jumlah dokumen yang ditemukan dan dianggap relevan untuk kebutuhan si pencari informasi. *Recall* (perolehan) berhubungan dengan kemampuan sistem untuk memanggil dokumen yang relevan. *Precision* (ketepatan) berkaitan dengan kemampuan sistem untuk tidak memanggil dokumen yang tidak relevan [6].

Lancaster merumuskan matriks *recall* dan *precision* yang dapat dilihat pada Tabel 1.

Tabel 1 Matriks *Recall* dan *Precision*
Lancaster

	Relevan	Tidak Relevan	Total
Ditemukan	a	b	a + b
Tidak Ditemukan	c	d	c + d
Total	a + c	b + d	a + b + c + d

Lalu, berdasarkan tabel tersebut, rumus *recall* dan *precision* dapat dilihat pada persamaan (9) dan (10).

$$\text{Recall} = \left[\frac{a}{a+c} \right] \times 100 \quad (9)$$

$$\text{Precision} = \left[\frac{a}{a+b} \right] \times 100 \quad (10)$$

3. HASIL DAN PEMBAHASAN

3.1 Data

Data yang digunakan pada penelitian ini yaitu artikel arkeologi yang ada di jurusan Arkeologi Universitas Halu Oleo. Data artikel arkeologi yang digunakan sebanyak 215 data. Untuk menghitung tingkat kemiripan antara *query* yang di-input-kan oleh *user* dengan dokumen pada *database* dihitung berdasarkan isi pdf.

3.2 Penerapan Metode

3.2.1 Perhitungan Manual

Contoh implementasi sederhana dari *vector space model* adalah sebagai berikut :

1. *Query*: alat tukar masyarakat papua
2. Dokumen 1 (D_1): Menurut masyarakat, dalam suatu pembuatan alat batu atau kapak batu hanya dapat dilakukan oleh orang-orang tertentu saja, dan bahkan dalam pembuatannya tidak boleh dilihat oleh orang lain ...
3. Dokumen 2 (D_2): Pemakaman dengan dikubur di dalam tanah merupakan yang lazim dilakukan oleh masyarakat di Papua setelah mengenal Injil, yang dilakukan sampai saat ini.
4. Dokumen 215 (D_{215}): Hubungan pertama adalah hubungan pusat kerajaan Sriwijaya dengan masyarakat pantai daerah bawahannya, dan kedua adalah hubungan penguasa kerajaan Sriwijaya dengan negara-negara tetangganya.

Setelah dilakukan tahap *case folding* dan *tokenizing*, *Filtering*, *Stemming*, dan *Indexing*. Selanjutnya tahap perhitungan tf-idf *query* dan

masing-masing dokumen yang dapat dilihat pada Tabel 2.

Tabel 2 Perhitungan Tf – Idf

Kata	Tf					df	N/df	log (N/df)
	Q	D ₁	D ₂	D _{..}	D ₂₁₅			
masyarakat	1	13	50	...	26	183	1,17	0,069
suatu	0	14	40	...	13	187	1,15	0,060
buat	0	10	6	...	1	167	1,28	0,107
alat	1	3	1	...	0	135	1,59	0,201
batu	0	192	2	...	8	174	1,23	0,089
kapak	0	6	0	...	0	35	6,14	0,788
orang	0	11	43	...	11	178	1,21	0,082
lihat	0	0	0	...	2	63	3,41	0,532
makam	0	2	1	...	17	55	3,91	0,592
kubur	0	2	33	...	4	93	2,31	0,363
tanah	0	4	17	...	3	179	1,20	0,079
lazim	0	0	1	...	1	30	7,17	0,855
papua	1	69	63	...	0	68	3,16	0,499
kenal	0	2	5	...	6	181	1,19	0,075
injil	0	0	0	...	0	7	30,71	1,487
hubung	0	4	6	...	4	184	1,17	0,069
pertama	0	0	0	...	0	2	107,5	2,031
pusat	0	12	0	...	5	163	1,32	0,120
raja	0	11	0	...	13	122	1,76	0,245
sriwijaya	0	0	0	...	0	21	10,24	1,010
pantai	0	0	0	...	1	92	2,34	0,369
daerah	0	0	20	...	4	180	1,19	0,075
bawah	0	0	0	...	0	59	3,64	0,561
dua	0	0	0	...	0	5	43	1,633
kuasa	0	0	1	...	2	103	2,09	0,320
negara	0	0	3	...	1	71	3,03	0,481
tetangga	0	0	2	...	0	17	12,65	1,102

Selanjutnya tahap perhitungan bobot query dan masing-masing dokumen yang dapat dilihat pada Tabel 3.

Tabel 3 Perhitungan Bobot

Kata	W				
	Q	D ₁	D ₂	D _{..}	D ₂₁₅
masyarakat	0,069	0,897	3,45	...	1,794
suatu	0	0,84	2,4	...	0,78
buat	0	1,07	0,642	...	0,107
alat	0,201	0,603	0,201	...	0
batu	0	17,088	0,178	...	0,712
kapak	0	4,728	0	...	0
orang	0	0,902	3,526	...	0,902
lihat	0	0	0	...	1,064
makam	0	1,184	0,592	...	10,064
kubur	0	0,726	11,979	...	1,452
tanah	0	0,316	1,343	...	0,237

lazim	0	0	0,855	...	0,855
papua	0,499	34,431	31,437	...	0
kenal	0	0,15	0,375	...	0,45
injil	0	0	0	...	0
hubung	0	0,276	0,414	...	0,276
pertama	0	0	0	...	0
pusat	0	1,44	0	...	0,6
raja	0	2,695	0	...	3,185
sriwijaya	0	0	0	...	0
pantai	0	0	0	...	0,369
daerah	0	0	1,5	...	0,3
bawah	0	0	0	...	0
dua	0	0	0	...	0
kuasa	0	0	0,320	...	0,64
negara	0	0	1,443	...	0,481
tetangga	0	0	2,204	...	0

Selanjutnya tahap perhitungan panjang query dan masing-masing dokumen yang dapat dilihat pada Tabel 4.

Tabel 4 Perhitungan Panjang Query dan Dokumen

Kata	Q	D _j				
	(W _{1q}) ²	(W _{D1}) ²	(W _{D2}) ²	(W _{D..})	(W _{D215})	
masyar akat	0,004761	0,8046	11,902 5	...	3,2184	
suatu	0	0,7056	5,76	...	0,6084	
buat	0	1,1449	0,4121	...	0,0114	
alat	0,040401	0,3636	0,0404	...	0	
batu	0	219,999	0,0316	...	0,5069	
kapak	0	22,3539	0	...	0	
orang	0	0,8136	12,432 6	...	0,8136	
lihat	0	0	0	...	1,1320	
makam	0	1,4018	0,3504	...	101,28 40	
kubur	0	0,5270	143,49 6	...	2,1083	
tanah	0	0,0998	1,8036	...	0,0561	
lazim	0	0	0,7310	...	0,7310	
papua	0,2490001	1185,49 3	988,28 4	...	0	
kenal	0	0,0225	0,1406	...	0,2025	
injil	0	0	0	...	0	
hubung	0	0,0761	0,1713	...	0,0761	
pertama	0	0	0	...	0	
pusat	0	2,0736	0	...	0,36	
raja	0	7,2630	0	...	10,144 2	
sriwijaya	0	0	0	...	0	

pantai	0	0	0	...	0,1362
daerah	0	0	2,25	...	0,09
bawah	0	0	0	...	0
dua	0	0	0	...	0
kuasa	0	0	0,102 4	...	0,4096
negara	0	0	2,082 2	...	0,2314
tetang ga	0	0	4,857 6	...	0
.
.
Total	0,790713	1161,5 564	978,9 432	...	3,0910 44
	$\sqrt{\sum_{j=1}^n (W_{iq})}$		$\sqrt{\sum_{i=1}^n (W_{ij})^2}$		
	0,889220	34,081 6	31,28 80	...	1,7581 36

Selanjutnya tahap perhitungan similaritas *query* dan masing-masing dokumen yang dapat dilihat pada Tabel 5.

Tabel 5 Perhitungan Similaritas *Query* dan Dokumen

Kata	<i>Sim(Q, D_j)</i>			
	$W_{iq} \cdot W_{D1}$	$W_{iq} \cdot W_{D2}$	$W_{iq} \cdot W_{D...}$	$W_{iq} \cdot W_{D215}$
masyarakat	0,0619	0,2380	...	0,1238
suatu	0	0	...	0
buat	0	0	...	0
alat	0,1212	0,0404	...	0
batu	0	0	...	0
kapak	0	0	...	0
orang	0	0	...	0
lihat	0	0	...	0
makam	0	0	...	0
kubur	0	0	...	0
tanah	0	0	...	0
lazim	0	0	...	0
papua	17,1811	15,6871	...	0
kenal	0	0	...	0
injil	0	0	...	0
hubung	0	0	...	0
pertama	0	0	...	0
pusat	0	0	...	0
raja	0	0	...	0
sriwijaya	0	0	...	0
pantai	0	0	...	0
daerah	0	0	...	0
bawah	0	0	...	0

dua	0	0	...	0
kuasa	0	0	...	0
negara	0	0	...	0
tetangga	0	0	...	0
.
	$\sum_{i=1}^n (W_{iq} \cdot W_{ij})$			
Total	17,4915	16,1198	...	0,1188

Sesuai dengan perhitungan menggunakan Persamaan (8) maka nilai *cosine* setiap dokumen didapat terlihat pada Tabel 6.

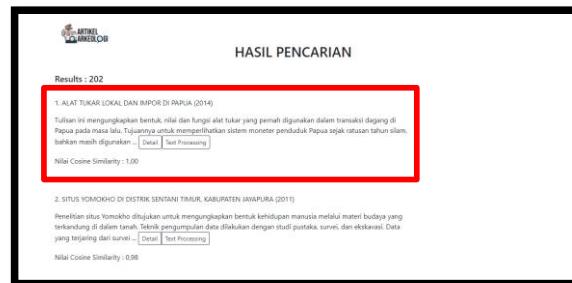
Tabel 6 Hasil Perhitungan Vector Space Model

<i>Sim(Q, D_j)</i>	D ₁	D ₂	D _{...}	D ₂₁₅
0,5771	0,5793	...	0,0760	

Dari hasil perhitungan *vector space model*, dapat kita simpulkan bahwa dokumen yang memiliki tingkat kemiripan tertinggi dan menghasilkan nilai bobot dokumen mendekati atau sama dengan 1, terdapat pada dokumen 168 (D₁₆₈) yang berjudul ‘Alat Tukar Lokal dan Impor di Papua’.

3.2.2 Perhitungan Sistem

Dari hasil perhitungan menggunakan sistem dapat dilihat bahwa dokumen yang memiliki tingkat kemiripan tertinggi dan menghasilkan nilai bobot dokumen mendekati atau sama dengan 1, yaitu dokumen yang berjudul ‘Alat Tukar Lokal dan Impor di Papua’. Adapun hasil perhitungan menggunakan sistem dapat dilihat pada Gambar 1.



Gambar 1 Hasil Perhitungan Sistem

3.3 Hasil Uji *Recall* dan *Precision*

Pengujian sistem dilakukan dengan menggunakan perhitungan *recall* dan *precision* seperti Persamaan (9) dan (10) Adapun matriks

recall dan *precision* Lancaster dari *query* ragam budaya papua dapat dilihat pada Tabel 7.

Tabel 7 Matriks Recall dan Precision
Lancaster

	Relevan	Tidak Relevan	Total
Ditemukan	202	0	202
Tidak Ditemukan	10	3	13
Total	212	3	215

Lalu, berdasarkan tabel tersebut, rumus *recall* dan *precision* pun menjadi:

$$\text{Recall} = \left[\frac{202}{(202+10)} \right] \times 100 = \left[\frac{202}{212} \right] \times 100 = 0,9528 \times 100 = 95,28$$

$$\text{Precision} = \left[\frac{202}{(202+0)} \right] \times 100 = \left[\frac{202}{202} \right] \times 100 = 1 \times 100 = 100$$

Berdasarkan perhitungan yang telah dilakukan dapat dilihat bahwa, *query* ragam budaya papua menghasilkan nilai *recall* sebesar 95,28%, hal itu dikarenakan sistem hanya mampu menampilkan 202 dokumen relevan dari 212 dokumen relevan. Sedangkan, nilai *precision* dihasilkan sebesar 100% dikarenakan sistem mampu menghasilkan dokumen sebanyak 202 dokumen dan semuanya relevan

4. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan mengenai metode *Vector Space Model* (VSM) pada sistem pencarian artikel arkeologi, maka diperoleh kesimpulan yaitu metode *Vector Space Model* (VSM) dapat diterapkan dalam sistem pencarian artikel arkeologi, *Vector space model* akan menghitung setiap dokumen didalam *database* yang memiliki kata yang sama dengan *query*, walau hanya ada satu saja kata yang sama, terlihat dari perangkingan artikel arkeologi yang ditampilkan, menghasilkan nilai *precision* dan *recall* yang dilakukan pada *query* yang terdiri dari 1 kata memiliki nilai rata-rata *recall* = 92,755% dan nilai rata-rata *precision* = 98,125%. Pada *query* yang terdiri dari 2 kata memiliki nilai rata-rata *recall* = 94,073% dan nilai rata-rata *precision* = 78,073%. Pada *query* yang terdiri dari 3 suku kata memiliki nilai rata-rata *recall* = 93,163% dan nilai rata-rata

precision = 97,361%. Pada *query* yang terdiri dari 4 kata memiliki nilai rata-rata *recall* = 95,04% dan nilai rata-rata *precision* = 95,034%. Pada *query* yang terdiri dari 5 suku kata memiliki nilai *recall* = 91,09% dan nilai *precision* = 91,72%. Dan saat diinputkan kalimat acak dan tidak bermakna, sistem tidak menampilkan data apapun.

5. SARAN

Saran yang perlu diperhatikan untuk pengembangan penelitian selanjutnya yaitu menggunakan metode lain selain *Vector Space Model*, karena pada metode ini untuk melakukan pencarian dari isi *full pdf*, masih kurang efektif dari segi waktu pencarinya.

DAFTAR PUSTAKA

- [1] G. Suryoputro, S. Riadi, and A. Sya'ban, *Menulis Artikel Untuk Jurnal Ilmiah*. 2012.
- [2] B. V. Christioko and A. F. Daru, "Sistem Temu Kembali Informasi Untuk Pencarian Judul Tugas Akhir Berbasis Kata Kunci," *J. Pengemb. Rekayasa dan Teknol.*, vol. 14, no. 2, p. 41, 2019, doi: 10.26623/jprt.v14i2.1226.
- [3] F. Sanjaya, "Pemanfaatan Sistem Temu Kembali Informasi dalam Pencarian Dokumen Menggunakan Metode Vector Space Model," *J. Inf. Technol.*, vol. 5, no. 2, pp. 147–153, 2018.
- [4] M. F. Somu, "Sistem Temu Balik Informasi Dokumen Tekstual Menggunakan Vector Space Model," no. 586, 2017.
- [5] V. B. Wicaksosno, R. Saptono, and S. W. Sihwi, "Analisis Perbandingan Metode Vector Space Model dan Weighted Tree Similarity dengan Cosine Similarity pada kasus Pencarian Informasi Pedoman Pengobatan Dasar di Puskesmas," *J. Teknol. Inf. ITS*, vol. 4, no. 2, p. 73, 2016, doi: 10.20961/its.v4i2.1768.
- [6] N. P. Lestari, "Uji Recall and Precision Sistem Temu Kembali Informasi OPAC Perpustakaan ITS Surabaya SKRIPSI," *Univ. Airlangga*, p. 1, 2016, [Online]. Available:

<http://journal.unair.ac.id/LN@uji-recall-and-precision-sistem-temu-kembali-informasi-opac-perpustakaan-its-surabaya-article-10825-media-136-category-8.html>.
