

PENERAPAN *LATENT SEMANTIC INDEXING* PADA SISTEM TEMU BALIK INFORMASI PADA UNDANG-UNDANG PEMILU BERDASARKAN KASUS

Agung Hasbi Ardiansyah¹, Kurnia Paranita Kartika², Saiful Nur Budiman³

^{1,2,3} Program Studi Teknik Informatika S1, Fakultas Teknologi Informasi
Universitas Islam Balitar, Jl. Majapahit No. 2-4, Kota Blitar, Indonesia
agunghasbi13@gmail.com

ABSTRAK

Laporan dugaan pelanggaran merupakan laporan yang disampaikan secara tertulis oleh pelapor kepada pengawas pemilu tentang dugaan terjadinya pelanggaran pemilu. Ketika mendapat temuan atau laporan dugaan kasus pelanggaran pemilu, pengawas pemilu akan melakukan klarifikasi dan pencarian bukti-bukti yang cukup sebelum menentukan temuan atau laporan tersebut termasuk kedalam pelanggaran atau tidak. Pada saat proses klarifikasi, pengawas pemilu mencari pasal yang kemungkinan dilanggar pada temuan atau laporan yang masuk. Banyaknya pasal rujukan untuk masing-masing kasus pada temuan atau laporan terkadang menghambat pekerjaan petugas pengawas pemilu, sehingga dibutuhkan sebuah alat bantu untuk mempercepat proses pencarian pasal berdasarkan kasus pelanggaran. Pada penelitian ini, sistem temu balik informasi digunakan untuk mencari pasal-pasal pada undang-undang nomor 10 tahun 2016 yang relevan pada suatu kasus berdasarkan deskripsi kasus. Pada penelitian ini digunakan metode *Latent Semantic Indexing* (LSI). LSI menggunakan teknik *Singular Value Decomposition* (SVD) untuk mereduksi dimensi. Pada penelitian ini digunakan 37 pasal, dan 4 kasus atau deskripsi pelanggaran sebagai *query*. Sistem menerima masukkan berupa *query* atau deskripsi kasus pelanggaran kemudian menghitung dan menentukan pasal yang terkait. Tingkat keberhasilan dari metode ini untuk menemukan hasil pencarian yang relevan dapat dilihat melalui besar 100% untuk *recall*, 70% untuk *precision* dan 82% untuk *f-measure*.

Kata Kunci: *latent semantic indexing*, sistem temu balik informasi, *cosine similarity*, *tf-idf*

1. PENDAHULUAN

Laporan dugaan pelanggaran adalah laporan yang disampaikan secara tertulis oleh pelapor kepada pengawas pemilu tentang dugaan terjadinya pelanggaran pemilu. Ketika mendapat temuan atau laporan dugaan kasus pelanggaran pemilu, pengawas pemilu akan melakukan klarifikasi dan pencarian bukti-bukti yang cukup sebelum menentukan temuan atau laporan tersebut termasuk kedalam pelanggaran atau tidak. Pada saat proses klarifikasi, pengawas pemilu mencari pasal yang kemungkinan dilanggar pada temuan atau laporan yang masuk. Banyaknya pasal rujukan untuk masing-masing kasus pada temuan atau laporan terkadang menghambat pekerjaan petugas pengawas pemilu, sehingga dibutuhkan sebuah alat bantu untuk mempercepat proses pencarian pasal berdasarkan kasus pelanggaran. Sistem temu balik informasi dapat menjadi jawaban atas permasalahan tersebut, metode *Latent Semantic Indexing* (LSI) merupakan sebuah metode pengindeksan pada sistem temu balik informasi yang menggunakan teknik *singular value decomposition* (SVD) untuk mengidentifikasi kesamaan makna kata berdasarkan pola dan hubungan antara istilah dan konsep yang terdapat pada kumpulan kata (Froud, Lachkar dan Ouatik, 2013).

Banyaknya penggunaan metode *Latent Semantic Indexing* pada sistem temu balik informasi memunculkan pertanyaan berapakah nilai *recall* dan nilai *precision* yang dihasilkan sistem temu balik informasi yang menggunakan metode LSI, sehingga membuat LSI banyak digunakan sebagai metode pada sistem temu balik informasi. Menurut Lee Pao dalam (Dwiyantoro, 2017) terdapat dua hal penting yang biasanya digunakan untuk mengukur kemampuan suatu sistem temu balik

informasi yaitu rasio atau perbandingan dari perolehan (*recall*), dan ketepatan (*precision*). Sistem temu balik informasi juga dapat dinyatakan efektif apabila mampu mendapatkan hasil pencarian dengan ketepatan (*precision*) yang tinggi sekalipun perolehannya rendah.

Dibuatnya sistem temu balik informasi pasal pada undang-undang pemilu berdasarkan kasus pelanggaran pemilu dengan menggunakan metode LSI bertujuan untuk mengukur nilai *recall*, dan nilai *precision* dari metode LSI. Dengan objek yang digunakan adalah pasal pada undang-undang pemilu sebagai dokumen yang akan diambil (*retrieve*) dari penyimpanan dan kasus pelanggaran pemilu yang sebelumnya telah di *preprocess* sebagai permintaan (*query*).

2. TINJAUAN PUSTAKA

2.1. Kajian Penelitian

Penelitian Hasan pada tahun 2018 merupakan penelitian tentang sistem temu balik informasi artikel kesehatan, aplikasi ini membantu penggunaannya dalam mencari artikel kesehatan. Penelitian tersebut menggunakan metode pembobotan kata TF.IDF dan untuk meningkatkan ketepatan hasil pencarian dari sistem, peneliti tersebut menggunakan metode LSI yang mempertimbangkan hubungan semantik antar kata. Hasil pengujian pada penelitian tersebut mendapatkan nilai *Mean Average Precision* (MAP) pada nilai *k* 10 sebesar 86% dan pada nilai *k* 15 sebesar 82.4%.

Penelitian Fawaz S. Al-Anzi dan Al-Anzi & AbuZeina (2017) meneliti tentang performa klasifikasi teks berbahasa arab menggunakan metode *singular value decomposition* (SVD) untuk

mengekstrak fitur tekstual berdasarkan LSI dan membandingkannya dengan metode klasifikasi terkenal lainnya. Penelitian tersebut menggunakan *corpus* sebanyak 4000 dokumen dari 10 topik (400 dokumen per masing-masing topik). Studi ini mengungkapkan bahwa metode klasifikasi yang menggunakan fitur LSI dengan nilai akurasi 82.50% yang secara signifikan mengungguli metode berbasis TF.IDF yang nilai akurasinya sebesar 67,25%.

2.2. Information Retrieval / Sistem Temu Balik Informasi

Menurut Ibrahim yang dikutip oleh (Wahyuddin, 2017) sistem temu balik informasi atau *information retrieval* merupakan proses yang berkaitan dengan penyajian, penyimpanan, pengambilan dan pencarian informasi yang berkaitan dengan kebutuhan informasi pengguna.

2.3. Text Preprocessing

Text Preprocessing adalah proses mempersiapkan teks untuk kemudian diproses lebih lanjut. Data teks atau karakter yang bersambung akan dipecah menjadi beberapa elemen yang bermakna (Hasan, 2018).

Berikut adalah tahapan pada *text preprocessing*:

1. *Case Folding* adalah proses mengubah semua huruf pada teks menjadi huruf kecil dan semua karakter selain huruf dihilangkan.
2. *Tokenizing* adalah proses pemisahan kalimat input menjadi token atau bagian tertentu. Sehingga proses ini dilakukan pemecahan teks yang berupa kalimat menjadi kata-kata.
3. *Filtering* adalah proses penghapusan kata-kata yang tidak penting pada data latih. Pada proses ini kita harus membuat atau mempunyai list kata-kata yang tidak penting atau yang biasa disebut dengan Stopword. Kata-kata hasil pemrosesan sebelumnya yaitu tokenizing, akan dicocokkan dengan data pada stopword. Jika kata yang diperiksa sama dengan kata yang terdaftar dalam daftar stopword, maka kata tersebut akan dihapus.
4. *Stemming* merupakan tahapan dimana dilakukan pencarian terhadap kata dasar dari tiap kata hasil *filtering*. *Stemming* mengubah berbagai bentuk kata ke dalam suatu bentuk kata dasar yang sama. Tahap ini dilakukan pemotongan atau penghilangan imbuhan seperti *prefix* (awalan) dan *suffix* (akhiran) serta *confix* (awalan dan akhiran).

2.4. Term Frequency – Inverse Document Frequency (TF-IDF)

TF adalah bobot suatu *term* pada suatu dokumen berdasarkan jumlah kemunculan *term* dalam dokumen tersebut (Yudho Baskoro, Ridok dan Tanzil Furqon, 2015). Untuk menghitung bobot TF digunakan persamaan sebagai berikut:

$$W_{d,t} = tf_{d,t} \quad (1)$$

Dimana $W_{d,t}$ adalah bobot TF, dan $tf_{d,t}$ adalah jumlah kemunculan t (*term*) dalam d (dokumen).

IDF merupakan logaritma rasio jumlah keseluruhan dokumen terhadap jumlah dokumen yang mengandung *term* terkait Darmawan & Wuriyanto dalam (Hasan, 2018). Untuk menghitung *Inverse Document Frequency* (IDF) digunakan persamaan sebagai berikut:

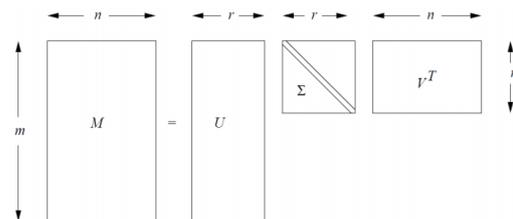
$$IDF_t = 1 + \log\left(\frac{D}{df}\right) \quad (2)$$

IDF_t merupakan bobot IDF, D adalah jumlah semua dokumen dalam koleksi dan df merupakan jumlah dokumen yang mengandung *term* t .

2.5. Latent Semantic Indexing (LSI)

Latent Semantic Indexing (LSI) merupakan sebuah metode pengindeksan pada sistem temu balik informasi yang menggunakan teknik *singular value decomposition* (SVD) untuk mengidentifikasi kesamaan makna kata berdasarkan pola dan hubungan antara istilah dan konsep yang terdapat pada kumpulan kata (Froud, Lachkar dan Ouatik, 2013).

Matrik M merupakan matrik frekuensi *term* pada dokumen. Berikut adalah ilustrasi SVD.



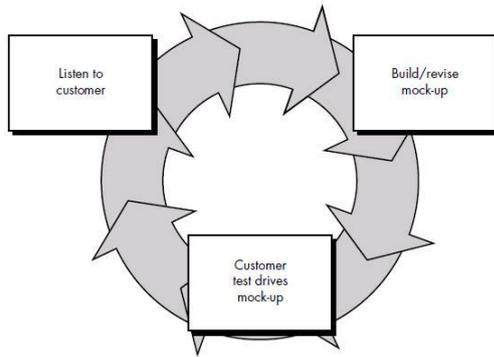
Gambar 1. Dekomposisi Matrik dengan SVD (Wahib dkk., 2015)

Dekomposisi SVD menghasilkan dua matriks orthogonal yaitu matrik U berukuran $m.k$ dan matrik V berukuran $n.k$. Sedangkan matrik Σ adalah matrik diagonal, semua elemen matrik selain elemen diagonalnya bernilai 0. Elemen diagonal dari matrik Σ disebut sebagai *singular values* dari matrik M Rajaraman & Ullman dalam (Wahib dkk., 2015).

3. METODE PENELITIAN

3.1. Metode Pengembangan

Penelitian ini menggunakan metode *prototype* untuk mengembangkan sistem.

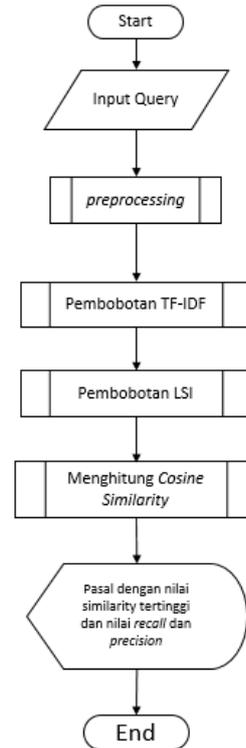


Gambar 2. Metode Prototype (Khosrow-Pour, 2005)

Berikut merupakan tahapan-tahapan pada metode *prototype*:

1. Pengumpulan kebutuhan. Pengumpulan kebutuhan meliputi kebutuhan pengguna, yaitu form untuk pencarian dan halaman untuk mengelola data pasal. Fungsi-fungsi pada sistem, meliputi fungsi untuk mengelola pasal, fungsi *preprocessing* data, fungsi pembobotan TF-IDF dan LSI, dan fungsi perhitungan kemiripan dengan *Cosine Similarity*. Input yang diperlukan, pada penelitian ini input-nya adalah pasal pada undang-undang pemilu dan *query* kasus pelanggaran pemilu. Output yang akan dihasilkan, pada penelitian ini hasil pencarian pasal dengan nilai *similarity* tertinggi.
2. Membangun *Prototype*. Membangun *Prototype* sistem antara lain desain tampilan, form *input* pasal dan *query* dan form *output* hasil pencarian.
3. Evaluasi *Prototype*. Evaluasi *Prototype* dilakukan untuk menyesuaikan keinginan pengguna setelah *prototype* dibangun.
4. *Coding*. Pada tahap ini *prototype* yang telah disetujui oleh pengguna mulai dibangun menjadi sistem aplikasi.
5. Menguji Sistem. Pengujian sistem bertujuan untuk mengukur kemampuan kinerja aplikasi. Pengujian juga dilakukan untuk mengukur kesesuaian rancangan dengan hasil aplikasi, dan relevansi hasil penelitian terhadap permasalahan yang diteliti.
6. Evaluasi Sistem. Sistem yang sudah dibuat akan dievaluasi oleh pengguna untuk mengetahui apakah sistem sudah sesuai atau masih perlu dilakukan perbaikan.
7. Penggunaan Sistem. Pengguna menggunakan sistem aplikasi yang telah dibuat.

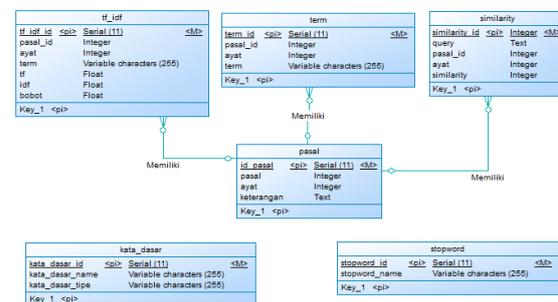
3.2. Flowchart



Gambar 3. Flowchart sistem pencarian pasal

Pada sistem temu balik informasi pasal ini pengguna terlebih dahulu memasukkan *query* di kotak pencarian yang merupakan deskripsi kasus pelanggaran. Sistem kemudian melakukan *preprocessing* pada *query*, selanjutnya dilakukan pembobotan TF-IDF dan pembobotan LSI. Hasil pembobotan dilakukan perhitungan *cosine similarity* untuk mendapatkan hasil pencarian pasal yang dengan *query*, kemudian diambil pasal yang memiliki nilai *similarity* tertinggi dan ditampilkan ke pengguna.

3.3. Entity Relationship Diagram (ERD)

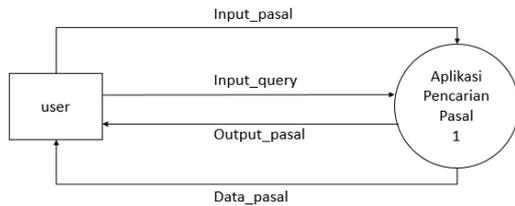


Gambar 4. ERD

Dalam perancangan sistem ini, terdapat 6 tabel, yaitu *tf_idf*, *term*, *similarity*, *pasal*, *kata_dasar*, dan *stopword*.

3.4. Data Flow Diagram (DFD) Level 0

Proses sistem digambarkan pada DFD level 0 bahwa pengguna dapat menginputkan data pasal dan query pencarian. Kemudian sistem akan mengembalikan data pasal dan hasil proses pencarian.



Gambar 5. DFD Level 0

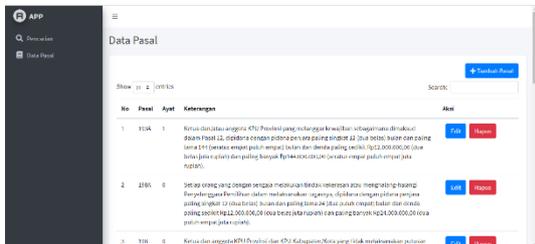
4. HASIL DAN PEMBAHASAN

4.1. Tampilan Sistem

Pada sistem temu balik informasi pasal ini terdapat 2 halaman yaitu halaman pencarian dan halaman data pasal.



Gambar 6. Halaman pencarian



Gambar 7. Halaman data pasal

4.2. Tahap text preprocessing

1. Case Folding

Pada tahap ini semua karakter huruf diubah menjadi huruf kecil.

Tabel 1. Hasil Case Folding

Sebelum Case Folding	Sesudah Case Folding
Setiap orang yang pada waktu pemungutan suara dengan sengaja melakukan perbuatan melawan hukum mengaku dirinya sebagai orang lain untuk menggunakan hak pilih	setiap orang yang pada waktu pemungutan suara dengan sengaja melakukan perbuatan melawan hukum mengaku dirinya sebagai orang lain untuk menggunakan hak pilih

2. Tokenizing

Pada *tokenizing* dilakukan penghilangan tanda baca dan pemotongan kata pada kalimat dalam dokumen.

Tabel 2. Hasil Tokenizing

Sebelum Tokenizing	Sesudah Tokenizing
setiap orang yang pada waktu pemungutan suara dengan sengaja melakukan perbuatan melawan hukum mengaku dirinya sebagai orang lain untuk menggunakan hak pilih	["setiap","orang","yang","pada","waktu","pemungutan","suara","dengan","sengaja","melakukan","perbuatan","melawan","hukum","mengaku","dirinya","sebagai","orang","lain","untuk","menggunakan","hak","pilih"]

3. Stopword Removal

Tahap selanjutnya adalah *Stopword Removal* adalah proses pembuangan *stopword* untuk mengurangi volume kata sehingga hasil pencarian menjadi lebih akurat karena *noise* akan direduksi.

Tabel 3. Contoh hasil Stopword Removal

Sebelum Stopword Removal	Sesudah Stopword Removal
["setiap","orang","yang","pada","waktu","pemungutan","suara","dengan","sengaja","melakukan","perbuatan","melawan","hukum","mengaku","dirinya","sebagai","orang","lain","untuk","menggunakan","hak","pilih"]	["orang","pemungutan","suara","sengaja","perbuatan","melawan","hukum","orang","hak","pilih"]

4. Stemming

Pada penelitian ini, *stemming* dilakukan menggunakan Stemmer Sastrawi. Sastrawi adalah *library* PHP untuk mengubah kata-kata dalam bahasa indonesia ke bentuk dasarnya.

Tabel 4. Contoh hasil Stemming

Sebelum Stemming	Sesudah Stemming
["orang","pemungutan","suara","sengaja","perbuatan","melawan","hukum","orang","hak","pilih"]	["orang","mungut","suar","sengaja","buat","lawan","hukum","orang","hak","pilih"]

Hasil *preprocessing* kemudian disimpan ke *database* untuk tahap selanjutnya yaitu pembobotan.

4.3. Pembobotan TF-IDF

Dengan menggunakan persamaan (1) didapatkan nilai TF dan persamaan (2) digunakan untuk mendapatkan nilai IDF. Bobot adalah hasil kali dari nilai TF dan IDF.

term	tf	idf	bobot
orang	2	1.27875	2.55751
mungut	1	1.46584	1.46584
suara	1	1.27875	1.27875
sengaja	1	1.14842	1.14842
buat	1	1.40369	1.40369
lawan	1	1.40369	1.40369
hukum	1	1.37566	1.37566
hak	1	1.67669	1.67669
pilih	1	1.37566	1.37566

Gambar 8. Hasil pembobotan TF-IDF

4.4. Pembobotan LSI

Semua *term* hasil *preprocessing* dari semua pasal dibentuk menjadi sebuah matrik (Matrik M). Baris merepresentasikan semua *term* unik pada semua pasal dan Kolom merepresentasikan pasal. Bobot hasil pembobotan TF-IDF dimasukkan ke matrik berdasarkan *term* dan pasalnya. Jika *term* tidak ada dalam pasal maka diberi nilai 0 pada pasal tersebut.

	Pasal 1	Pasal 2	Pasal 3	Pasal 4
orang	2.55751	1.37566	0.00000	2.60241
sengaja	1.14842	1.30132	0.00000	0.00000
laku	0.00000	1.60211	0.00000	1.27875
buat	1.40369	0.00000	0.00000	1.27875
lawan	1.40369	1.46584	0.00000	1.27875
hukum	1.37566	1.46584	0.00000	1.27875
palsu	0.00000	0.00000	0.00000	0.00000
data	0.00000	0.00000	0.00000	0.00000
daftar	0.00000	0.00000	0.00000	0.00000
pilih	1.37566	1.14842	0.00000	0.00000
pasal	0.00000	1.14842	0.00000	0.00000
anggota	0.00000	0.00000	1.04113	0.00000
pps	0.00000	0.00000	1.04113	0.00000

Gambar 9. Sampel Matrik M

Setelah matriks terbentuk, selanjutnya matriks tersebut akan di dekomposisi menggunakan SVD. Dan menghasilkan matriks U, S, V. Dalam kaitannya agar mempermudah perhitungan selanjutnya, matriks hasil SVD di lakukan pengurangan dimensi, sebanyak nilai *k* dengan tujuan untuk mempercepat proses. Prosesnya adalah mengambil *k* kolom pertama dari matriks U dan V. dan mengambil *k* baris dan kolom pertama pada matriks S. Sehingga matriks hasil SVD disebut matriks U_k, S_k, V_k^T .

U_k		S_k	
0.0336	0.0199	0.0255	0
0.167	0.057	0	0.04
-0.0745	0.0849		
-0.0282	0.009		
0.05	-0.0609		
0.0024	0.0167		
-0.009	0.0009		
0.008	0.0197		
0.0159	-0.0051		
0.0065	0.0659		
-0.0411	0.0777		
-0.0839	-0.0508		
0.1468	0.1433		
0.002	0.0076		
0.0563	0.0872		
-0.0007	0.1178		
0.0224	-0.0138		
0.0365	-0.012		
0.0013	0.0016		
-0.0056	-0.0309		
0.0219	0.0172		
-0.0423	0.0315		
0.0298	-0.0462		
0.1027	0.2962		
0.1185	-0.3062		
-0.0002	0.0389		
0.0464	0.0175		

V_k^T					
0.0252	-0.0412	0.0241	-0.0264	-0.0376	0.1125
0.0328	-0.1143	0.0043	0.0177	-0.0599	0.2574

Gambar 10. Sampel matrik hasil dekomposisi

4.5. Penghitungan Similarity

Pada tahap perhitungan *similarity*, perhitungan menggunakan *cosine similarity*. Perhitungan ini adalah antara *query* yang dimasukkan pengguna dengan masing-masing pasal yang didasarkan dengan matriks hasil pembobotan LSI.

Setelah *query* di proses di tahap *preprocessing* dan TF-IDF didapatkan matriks *query*.

0.00000
0.00000
4.20532
0.00000
0.00000
0.00000
1.27875
1.37566
0.00000
0.00000
2.27875
0.00000
0.00000

Gambar 11. Sampel Matrik Query

Kemudian matriks *query* dikalikan dengan matriks U_k , dan matriks S_k . Yang hasilnya adalah vektor *query*.

$$q = qU_k S_k \quad (3)$$

q =

0	0	4.20532	0	0	0	1.27875	1.37566	0	0	2.27875	0	0
---	---	---------	---	---	---	---------	---------	---	---	---------	---	---

0.0336	0.0199
0.167	0.057
-0.0745	0.0849
-0.0282	0.009
0.05	-0.0609
0.0024	0.0167
-0.009	0.0009
0.008	0.0197
0.0159	-0.0051
0.0065	0.0059
-0.0411	0.0777
-0.0839	-0.0508
0.1468	0.1433

0.0255	0
0	0.04

q =

-0.0104	0.02249
---------	---------

Vektor query yang dihasilkan kemudian dibandingkan dengan seluruh vektor pasal dalam matriks V^T_k . Jumlah kolom pada matriks V^T_k menginisialisasikan jumlah pasal yang terlibat, misalkan kolom pertama pada matriks V^T_k menginisialisasikan pasal pertama.

$$sim(q, p) = \frac{q+p}{|q||p|} \tag{4}$$

$$sim(q, p1) = \frac{(-0.010)(0.022)+(0.025)(0.383)}{\sqrt{(-0.010)^2+(0.022)^2} \sqrt{(0.025)^2+(0.383)^2}}$$

$sim(q, p1) = 0.47$

Berdasarkan perhitungan diatas hasil similarity query dengan pasal 1 adalah 0.44.

4.6. Pengujian Metode

Pada penelitian ini digunakan data berjumlah 37 pasal dan 4 kasus kejahatan yang terkait dengan pelanggaran yang diatur pada pasal pada UU No. 10 Tahun 2016 sebagai query, dengan menggunakan nilai k 20, 25, 30, dan 35.

Tabel 5. Hasil pengujian precision, recall, dan f-measure

No	Query	k	P	R	F-measure
1.	Q1.	20	0,3	1	0.46
		25	0,5	0,5	0.50
		30	0,5	0,5	0.50
		35	0,5	0,5	0.50
2.	Q2	20	0,1	1	0.18
		25	0,2	1	0.33
		30	0,2	1	0.33
		35	0,3	1	0.46
3.	Q3	20	0,4	1	0.57
		25	0,4	1	0.57
		30	0,4	1	0.57
		35	0,2	0,5	0.29
4.	Q4	20	0,7	0,5	0.58
		25	0,7	1	0.82
		30	0,7	1	0.82
		35	0,7	1	0.82

Berdasarkan hasil pengujian pada tabel diatas, nilai k 20 perolehan nilai precision, recall, dan f-measure yang didapat cukup rendah dengan nilai terendahnya pada query 2 yaitu 0,1, 1 dan 0,18.

karena nilai k mengurangi dimensi matrik sebanyak k, sehingga apabila nilai k semakin kecil maka data yang direduksi menjadi semakin besar, sehingga pada nilai k 20 ada banyak informasi yang terkena reduksi dimensi matrik.

Perolehan nilai precision, recall, dan f-measure mengalami peningkatan ketika nilai k semakin bertambah dengan nilai terbaik pada query 4 dengan nilai 0,7 untuk precision, 1 untuk recall, dan 0,82 pada f-measure. Namun pada query 3 mengalami penurunan nilai precision, recall, dan f-measure di nilai k 35, penambahan nilai k dapat mengurangi akurasi dari hasil pencarian, karena adanya penambahan noise pada data.

4.7. Pengujian Sistem

Pengujian sistem dilakukan oleh pengguna menggunakan instrumen berupa kuisioner. Hasil dari pengujian oleh pengguna Sistem Temu Balik Informasi Pasal Berdasarkan Kasus ditunjukkan pada Tabel 6.

Tabel 6. Hasil pengujian oleh pengguna

	Jumlah	Skor	Jumlah x Skor
STS	0	1	
TS	0	2	
RR	0	3	
S	51	4	204
SS	27	5	135
Total			339

Skor maksimal adalah skor apabila semua responden menjawab “Sangat Setuju” (SS) dengan skor 5. Sehingga Perhitungan skor maksimal adalah sebagai berikut:

Skor maksimal = total responden x jumlah pernyataan x 5. Dalam hal ini jumlah responden berjumlah 4 orang sehingga, Skor maksimal= 4 x 19 x 5 = 380.

Perhitungan persentase total skor menggunakan rumus:

Persentase total skor = Jumlah Skor Total / Skor Maksimal x 100%. = 339/380 x 100% = 89,21%.

Persentase hasil pengujian oleh pengguna adalah 89,21%. Setelah hasil pengujian ini kemudian dikonversi kedalam skala kualitatif didapatkan hasil “sangat tinggi”.

5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan perancangan, analisa, dan implementasi serta pengujian pada penerapan metode latent semantic indexing pada sistem temu balik informasi pasal pada undang-undang pemilu berdasarkan kasus pelanggaran pemilu, maka didapatkan kesimpulan:

1. Dengan menerapkan semua yang dibutuhkan untuk membangun metode Latent Semantic Indexing (LSI), seperti text processing, TF-

IDF, SVD dan *Cosine Similarity* dapat membangun sebuah sistem temu kembali informasi pada undang-undang pemilu berdasarkan kasus pelanggaran pemilu.

2. Tingkat keberhasilan dari sistem temu balik informasi dengan metode LSI ini dapat dilihat melalui besarnya nilai uji kinerja sistem yang menghasilkan nilai *recall* sebesar 100%, *precision* sebesar 70 dan *f-measure* sebesar 82%. Pengujian oleh pengguna menghasilkan persentase sebesar 89,21 % (sangat tinggi).

5.2. Saran

Berikut adalah saran yang peneliti berikan untuk penelitian kedepannya:

1. Pada penelitian ini masih menggunakan pasal dari UU no 10 tahun 2016 saja, untuk penelitian selanjutnya bisa ditambahkan undang-undang atau peraturan lain. Misal dari Perbub atau Perwali.
2. Karena jumlah pasal yang diproses banyak maka menambah lama waktu komputasinya. Pengembangan dan optimasi waktu komputasi yang cepat perlu dilakukan pada penelitian selanjutnya.

DAFTAR PUSTAKA

- [1] Al-Anzi, F.S. dan AbuZeina, D. (2017). Toward an enhanced Arabic text classification using cosine similarity and Latent Semantic Indexing. *Journal of King Saud University - Computer and Information Sciences*, 29 (2), 189–195. Tersedia pada <https://doi.org/10.1016/j.jksuci.2016.04.001>.
- [2] Dwiyantoro, D. (2017). Sistem Temu Kembali Dengan Keyword (Deskriptif Menggunakan Recall Dan Precision Pada Judul, Subjek Opac Perpustakaan Universitas Gadjah Mada). *Khizanah al-Hikmah: Jurnal Ilmu Perpustakaan, Informasi, dan Kearsipan*, 5 (2), 164–175. Tersedia pada <https://doi.org/10.24252/kah.v5i2a4>.
- [3] Froud, H., Lachkar, A. dan Ouatik, S.A. (2013). Arabic Text Summarization Based on Latent Semantic Analysis to Enhance Arabic Documents Clustering. *International Journal of Data Mining & Knowledge Management Process*, 3 (1), 79–95. Tersedia pada <https://doi.org/10.5121/ijdkp.2013.3107>.
- [4] Hasan, M. ismail. (2018). Information Retrieval System artikel kesehatan menggunakan Pembobotan TF.IDF dan Latent Semantic indexing. (*Doctoral dissertation, Universitas Islam Negeri Maulana Malik Ibrahim*).
- [5] Khosrow-Pour, M. (2005). *Encyclopedia of Information Science and Technology (5 Volumes)*. Idea Group Reference.
- [6] Wahib, A. dkk. (2015). Perangkingan Dokumen Berbahasa Arab Menggunakan Latent Semantic Indexing. *Jurnal Buana Informatika*, 6 (2), 83–92. Tersedia pada <https://doi.org/10.24002/jbi.v6i2.411>.
- [7] Wahyuddin. (2017). Pemanfaatan sistem temu balik informasi di dinas perpustakaan dan kearsipan kabupaten barru skripsi. (*Doctoral dissertation, Universitas Islam Negeri alauddin makassar*).
- [8] Yudho Baskoro, S., Ridok, A. dan Tanzil Furqon, M. (2015). Pencarian Pasal Pada Kitab Undang-Undang Hukum Pidana (Kuhp) Berdasarkan Kasus Menggunakan Metode Cosine Similarity Dan Latent Semantic Indexing (Lsi). *Journal of Enviromental Engineering and Sustainable Technology*, 2 (2), 83–88. Tersedia pada <https://doi.org/10.21776/ub.jeest.2015.002.02.4>.