

Otomatisasi Peringkasan Teks Pada Dokumen Hukum Menggunakan Metode *Latent Semantic Analysis*

Imam Fahrur Rozi¹, Kadek Suarjuna Batubulan², Millenia Rusbandi³

^{1,2,3}Teknik Informatika, Teknologi Informasi, Politeknik Negeri Malang
¹imam.rozi@polinema.ac.id, ²kadeksuarjuna87@gmail.com, ³rmllenia00@gmail.com

Abstrak

Pada masa sekarang, jumlah kriminalitas di Indonesia cukup banyak. Banyaknya jumlah kriminalitas di Indonesia akan berdampak pada jumlah dokumen hukum yang akan ditangani oleh aparat penegak hukum. Sehingga solusi yang dibutuhkan yaitu dengan membuat sebuah ringkasan pada dokumen yaitu nota pembelaan yang berbentuk PDF dengan tujuan untuk meminimalisir waktu yang dibutuhkan dalam memahami dokumen tersebut. Dalam hal meringkas teks, metode yang dapat digunakan yaitu algoritma LSA (*Latent Semantic Analysis*). Algoritma tersebut digunakan untuk menguraikan atau menganalisa makna yang masih tersembunyi dari suatu bahasa guna memperoleh informasi yang penting. Beberapa tahapan metode LSA yaitu pembobotan kata pada dokumen menggunakan metode TF-IDF, metode SVD (*Singular Value Decomposition*) untuk menghitung kesamaan kata antar kalimat dan menghitung panjang *length* pada matriks yang diperoleh dari perhitungan metode SVD. Dari pengujian terhadap 10 dokumen yang diringkas oleh pakar, diperoleh hasil *precision*, *recall*, *f-measure* dan *accuracy* secara berurutan pada peringkasan teks otomatis dengan metode *Latent Semantic Analysis* untuk *compression rate* 75% yaitu 53%, 27%, 35% dan 71% lalu untuk *compression rate* 50% yaitu 54%, 56%, 55% dan 75%, serta untuk *compression rate* 25% yaitu 51%, 79%, 61% dan 75%. Berdasarkan hasil penelitian dan pengujian yang telah dilakukan dapat ditarik kesimpulan bahwa Metode *Latent Semantic Analysis* dapat digunakan untuk melakukan peringkasan dokumen hukum.

Kata Kunci: peringkasan otomatis, dokumen hukum, *Latent Semantic Analysis*

1. Pendahuluan

Selama periode tahun 2015–2017, jumlah kasus kriminalitas di Indonesia cenderung fluktuatif (tidak stabil). Berdasarkan data Kepolisian Republik Indonesia (Polri), total jumlah kasus kriminalitas pada tahun 2015 sebanyak 352.936 kasus. Pada tahun 2016, kasus semakin meningkat menjadi sebanyak 357.197 kasus. Sedangkan pada tahun 2017, mengalami penurunan menjadi 336.652 kasus (Badan Pusat Statistik, 2018). Dari data tersebut, rata-rata jumlah kriminalitas periode tahun 2015-2017 sebesar 348.928 kasus. Banyaknya jumlah kriminalitas di Indonesia akan berdampak pada jumlah dokumen hukum yang akan ditangani oleh aparat penegak hukum.

Dalam pemahaman dokumen hukum, aparat penegak hukum seperti advokat, hakim, maupun jaksa harus membaca keseluruhan dokumen dimana akan memerlukan waktu yang cukup lama. Kapasitas yang dimiliki manusia untuk mengonsumsi informasi sangat terbatas. Oleh karena itu, sangatlah penting untuk menyaring informasi yang hanya berguna dari jumlah data tidak terstruktur. Sehingga salah satu solusi yang dibutuhkan yaitu dengan membuat sebuah ringkasan pada dokumen hukum dimana dokumen tersebut berbentuk pdf. Dokumen hukum yang akan

diringkas berupa nota pembelaan dan bagian yang diolah yaitu pendahuluan. Dilakukannya peringkasan pada bagian tersebut dikarenakan berbentuk paragraf dan dapat diringkas. Sehingga kalimat-kalimat yang tidak diperlukan akan dihilangkan.

Ringkasan adalah teks yang dihasilkan dari sebuah kumpulan teks yang mengandung informasi penting dengan panjang teks tidak lebih dari setengah teks aslinya (Radev, 2002). Dalam dunia komputer, peringkasan juga dikenal dengan Peringkasan teks Otomatis atau ATS (*automatic text summarization*). Peringkasan teks adalah pembuatan bentuk yang lebih singkat dari suatu teks dengan memanfaatkan aplikasi yang dijalankan dan dioperasikan pada komputer (Nugraha, 2008). Dengan adanya ringkasan tersebut, dapat mempermudah aparat penegak hukum dalam memahami dokumen tanpa harus membaca keseluruhan isi dokumen.

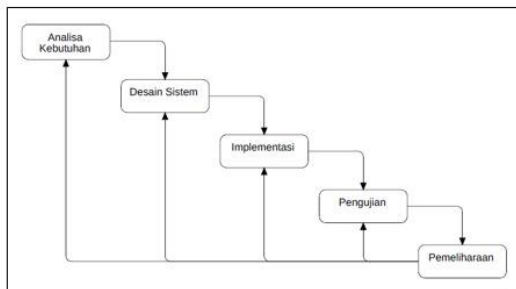
Pada penelitian yang telah dilakukan, umumnya menggunakan objek berupa artikel dan berita. Sedangkan untuk dokumen hukum, masih belum banyak dilakukan. Oleh karena itu, dalam penelitian ini mengusulkan peringkasan teks pada dokumen hukum dengan menggunakan metode *Latent Semantic Analysis* (LSA) diharapkan dapat menghasilkan

ringkasan kalimat dengan baik dan sesuai dengan isi dari dokumen hukum tersebut.

2. Metodologi Penelitian

Dalam sebuah penelitian agar lebih cepat dan tepat dalam mendeskripsikan solusi dan mengembangkan perangkat lunak juga hasilnya mudah dikembangkan dan dipelihara, maka pengembangan perangkat lunak memerlukan suatu metodologi khusus. Metode perangkat lunak merupakan suatu kerangka kerja yang digunakan untuk menstrukturkan, merencanakan, dan mengendalikan proses pengembangan suatu perangkat lunak (Ahmad, 2019).

Metode perangkat lunak terdiri dari beberapa jenis. Setiap metodenya memiliki karakteristik yang berbeda. Sehingga metode ini dipilih berdasarkan kebutuhan sistem. Metode yang digunakan pada penelitian ini yaitu metode *waterfall*. Metode ini dilakukan dengan pendekatan yang sistematis, langkah demi langkah yang dilalui harus diselesaikan satu per satu (tidak dapat meloncat ke tahap berikutnya) dan berjalan secara berurutan, oleh karena itu disebut *waterfall* (Hidayat, n.d.). Metode ini memiliki 5 tahapan yaitu analisis kebutuhan, desain sistem, implementasi, pengujian dan pemeliharaan (Sommerville, 2011) yang ditunjukkan pada Gambar 1.



Gambar 1 Metode *Waterfall*

2.1 Metode Pengolahan data

Metode pengolahan data yang ada sangat lah banyak jumlahnya. Akan tetapi, metode-metode tersebut tidak dapat digunakan secara sembarangan. Umumnya tidak ada metode yang lebih unggul dibandingkan dengan metode lainnya, karena pada dasarnya setiap metode pasti memiliki kelebihan dan kekurangan. Sehingga kita hanya perlu memilih metode mana yang tepat untuk digunakan sesuai dengan kondisi data yang dimiliki. Di bawah ini merupakan langkah-langkah pengolahan data yang digunakan pada penelitian:

2.1.1 Text Preprocessing

Sebelum melakukan analisis, *preprocessing* harus dilakukan terlebih dahulu untuk melakukan normalisasi kata-kata yang tidak diperlukan. Pada tahapan ini, data tekstual akan diubah menjadi teks agar dapat diolah oleh sistem. Penelitian ini menggunakan dokumen sebagai inputan awal. Proses yang digunakan antara lain:

a. Pembentukan Kalimat

Pembentukan kalimat yaitu pemecahan teks dokumen menjadi kumpulan kalimat berdasarkan *delimiter*.

b. Case Folding

Case folding merupakan perubahan huruf pada kalimat menjadi huruf kecil (*lowercase*) dan penghilangan karakter yang tidak valid seperti tanda baca.

c. Tokenizing

Pada proses ini, kalimat tersebut dipecah kembali menjadi beberapa kata tunggal penyusunnya berdasarkan spasi.

d. Stopword Removal

Stopword removal adalah proses penghilangan kata-kata yang tidak merepresentasikan isi dokumen.

e. Stemming

Stemming adalah proses pengembalian kata tunggal yang memiliki imbuhan menjadi kata dasar.

2.1.2 TF-IDF

Term frequency (TF) adalah pengukuran yang paling sederhana dalam metode pembobotan. Pada metode ini, masing-masing term diasumsikan mempunyai proporsi kepentingan sesuai jumlah kemunculan dalam teks dokumen. *Term frequency* dapat memperbaiki nilai *recall* pada *information retrieval*, tetapi tidak selalu memperbaiki nilai *precision* (Tokunaga & Iwayama, 1994). Hal ini disebabkan *term* yang *frequent* cenderung muncul di banyak teks, sehingga *term* tersebut memiliki kekuatan.

Inverse document frequency (IDF) adalah metode pembobotan *term* yang lebih condong (fokus) untuk memperhatikan kemunculan *term* pada keseluruhan kumpulan teks. Pada IDF, *term* yang jarang muncul pada keseluruhan koleksi teks dinilai lebih berharga. Nilai kepentingan tiap term diasumsikan berbanding terbalik dengan jumlah teks yang mengandung term tersebut (Tokunaga & Iwayama, 1994).

Term frequency inverse document frequency (TF-IDF) adalah metode pembobotan yang menggabungkan metode TF dan IDF. Metode ini

diusulkan oleh Salton (G, 1989) sebagai sebuah kombinasi metode yang dapat memberikan performansi yang lebih baik, khususnya dalam memperbaiki nilai *recall* dan *precision* (Tokunaga & Iwayama, 1994). Berikut ini merupakan perhitungannya:

$$TF.IDF = TF * (\log \frac{N}{DF} + 1) \tag{1}$$

Keterangan:

TF = Jumlah *term* tersebut

N = Total dokumen

DF = Jumlah dokumen yang mengandung suatu *term*

Hasil dari perhitungan TF-IDF ditunjukkan pada Tabel 1 di bawah ini.

Tabel 1 Matriks Hasil TF-IDF

TF-IDF		
1.176091	1.176091	0
1.477121	0	0
1.477121	0	0
1.477121	0	0
1.477121	0	0
1.477121	0	0
0	1.477121	0
0	1.477121	0
0	1.477121	0
0	1.477121	0
0	1.176091	1.176091
0	1.477121	0
0	1.477121	0
0	0	1.477121
0	0	1.477121

2.1.3 Latent Semantic Analysis

Latent Semantic Analysis (LSA) menurut bahasa terbagi atas beberapa kata yang penting yaitu *latent* dan *semantic*, *latent* yang memiliki arti tersembunyi atau sesuatu yang masih belum terlihat, sedangkan *semantic* berasal dari bahasa Yunani “*semantics*” yang berarti memberi tanda, penting atau cabang linguistik yang mempelajari arti dan makna dari suatu bahasa, kode atau jenis representasi lainnya.

Dari pengertian dapat ditarik kesimpulan bahwa LSA adalah menguraikan atau menganalisa makna yang masih tersembunyi dari suatu bahasa, kode atau jenis representasi lainnya, guna memperoleh informasi

yang penting. Pembobotan kata diperoleh dengan menggunakan metode TF-IDF. Berikutnya, Kesamaan kata dan kalimat diperoleh dengan cara menggunakan *Singular Value Decomposition* (SVD), di mana SVD mempunyai kapasitas untuk mereduksi *noise*, sehingga dapat meningkatkan hasil akurasi pada ringkasan (Peter & Kp, 2009).

$$A = USV^T \tag{2}$$

A adalah matriks dokumen yang mewakili kalimat atau kata yang dikenal dengan matriks Amn. Dari perhitungan TF-IDF, dilakukan pembentukan matriks A dengan cara memasukkan nilai TF-IDF. Hasil Matriks A ditunjukkan pada Tabel 2 di bawah ini.

Tabel 2 Matriks A

Kalimat 1	Kalimat 2	Kalimat 3
12,29262666	1,38319065	0
1,38319065	15,85770451	1,38319065
0	1,38319065	5,746965052

Selanjutnya dilakukan perhitungan menggunakan metode *Singular Value Decomposition* (SVD). Seperti pada persamaan (2) akan didapat matriks U mendeskripsikan matriks *orthogonal* $m \times m$ yang dikenal dengan istilah *Left Singular Vector*. *Right Singular Vektor* (V) merupakan matriks *orthogonal* $n \times n$ yang diperoleh dari *eigen vector* matriks $A^T A$, sedangkan matriks diagonal S dihasilkan dari *eigen value* matriks $A^T A$ yang diakarkan. Hasil Matriks V yang telah di *transpose* ditunjukkan pada Tabel 3 di bawah ini

Tabel 3 Matriks V^T

Kalimat 1	Kalimat 2	Kalimat 3
0,310572622	0,942769607	0,121367684
-0,950134866	0,304126009	0,068928272
0,028072398	-0,136722469	0,990211547

Tahapan selanjutnya adalah menghitung *length* dari setiap baris dari matriks V^t dengan menggunakan persamaan (3). Baris-baris pada kalimat yang mempunyai nilai *length* yang tinggi akan dijadikan sebagai ringkasan.

$$S_k = \sqrt{\sum_j^n (V^t)_{ij}^2 \cdot S_{jj}^2} \tag{3}$$

Keterangan :

S_k = panjang nilai setiap baris

$(V^t)_{ij}^2$ = nilai matriks V^t posisi ke- ij
 S_{jj}^2 = nilai diagonal matrik S posisi ke- jj
 i = bilangan bulat positif
 j = bilangan bulat positif

S_k adalah panjang vektor k pada kalimat yang dimodifikasi oleh laten vektor. n adalah jumlah ruang dimensi baru. Hasil dari length terbesar pada setiap dokumen kalimat akan dijadikan ringkasan. Hasil dari perhitungan dapat dilihat pada Tabel 4 di bawah ini.

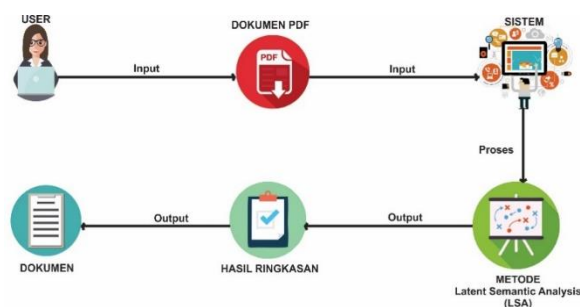
Tabel 4 Hasil Perhitungan S_k

Kalimat ke-	Hasil Skor
1	1,575819146
2	1,954901801
3	1,668110957

Didapatkan hasil bahwa kalimat 2 akan dijadikan sebagai ringkasan dikarenakan memiliki hasil skor tertinggi.

2.2 Arsitektur Sistem

Pada penelitian ini diterapkan tahap *preprocessing* pada dokumen hukum terlebih dahulu. Selanjutnya dilakukan peringkasan dokumen menggunakan metode *Latent Semantic Analysis*. Tahapan terakhir yaitu memilih kalimat berdasarkan skor peringkasan tertinggi. Hasil dari peringkasan yaitu berupa dokumen dengan format pdf. Arsitektur sistem penelitian ini ditunjukkan pada Gambar 2.



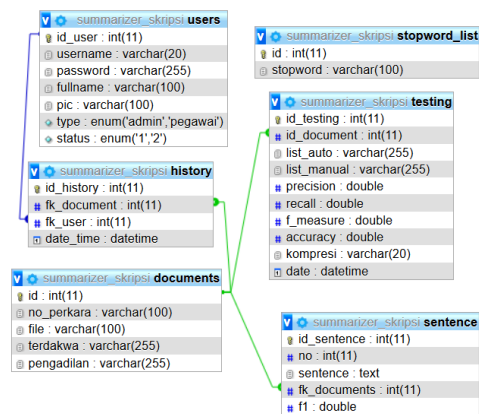
Gambar 2 Arsitektur Sistem

3. Hasil dan Pembahasan

Setelah dilakukan analisis dan perancangan terhadap sistem yang akan dibangun, selanjutnya adalah tahap implementasi.

3.1 Hasil Implementasi Basis Data

Database merupakan salah satu komponen penting dalam sistem informasi, karena merupakan dasar dalam menyediakan informasi. *Database* sendiri adalah kumpulan dari item data yang saling berhubungan satu dengan yang lainnya yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, yang kelak dapat dimanfaatkan kembali dengan cepat dan mudah (Minarni & Susanti, 2014). Pada proses pembangunan sistem, *database* diimplementasikan dengan nama *summarizer_skripsi* yang memiliki 6 tabel diantaranya yaitu: *users*, *history*, *documents*, *sentence*, *testing* dan *stopword_list*. Hal tersebut dapat dilihat pada Gambar 3.

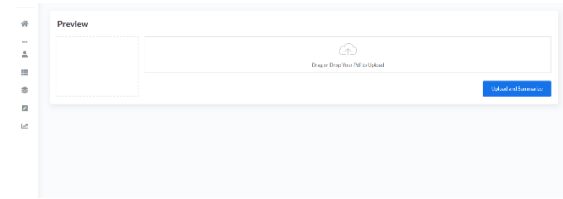


Gambar 3 Database

Pada tabel *user* memiliki 5 *fields* yaitu *id_user*, *username*, *password*, nama dan tipe *user*. Tabel ini berfungsi untuk menyimpan pengguna yang dapat mengakses *website* tersebut. Tabel dokumen memiliki 5 *fields* yaitu *id*, *no_perkara*, *file*, *terdakwa*, *pengadilan*. Tabel ini berfungsi menyimpan dokumen yang telah diringkaskan. Pada tabel *sentence* memiliki 5 *field* yaitu *id_sentence*, *no*, *sentence*, *fk_documents* dan *f1*. *Field* *fk_documents* pada tabel *sentence* memiliki relasi dengan *id* yang ada pada tabel *documents*. Tabel ini berfungsi menyimpan kalimat yang telah dilakukan *text preprocessing* berasal dari dokumen yang diringkaskan. Tabel *stopword_list* memiliki 2 *fields* yaitu *id* dan *stopword*. Tabel ini berfungsi untuk menyimpan *stopword* atau kata umum (*common words*) yang biasanya muncul dalam jumlah besar dan dianggap tidak memiliki makna. *Stopword* ini akan digunakan pada tahapan *preprocessing*. Tabel *testing* adalah tabel yang berfungsi untuk menyimpan hasil pengujian dokumen yang diringkaskan oleh sistem dengan pakar. Pada *field* *id_documents* memiliki relasi dengan *id* pada tabel *documents*. Tabel *history* berfungsi untuk menyimpan riwayat peringkasan yang dilakukan oleh *user*.

3.2 Hasil Implementasi Sistem

Sistem yang dibuat memiliki tampilan dan fungsi seperti Gambar 4 di bawah ini :



Gambar 4 Unggah Dokumen

Pada gambar di atas, dapat dilihat bahwa terdapat form untuk mengunggah dokumen yang akan diringkas. Selanjutnya pengguna dapat menekan tombol *upload and summarize* maka hasil peringkasan akan keluar secara otomatis.

Hasil peringkasan berdasarkan perhitungan metode yang digunakan yaitu *Latent Semantic Analysis*. Sebelum pengimplementasian metode, dilakukan *preprocessing* dan perhitungan bobot antar kata dengan menggunakan metode *TF-IDF* seperti Gambar 5 di bawah ini:

Gambar 5 Metode

3.3 Hasil Pengujian

Pengujian pada penelitian ini dilakukan dengan membandingkan hasil peringkasan yang dilakukan oleh sistem dan pakar dengan menggunakan *precision*, *recall*, *f-score* dan *accuracy*.

a. *Precision*

Precision adalah berapa banyak dokumen yang berhasil diambil oleh sistem. Di bawah ini merupakan rumus perhitungan *precision*.

$$\frac{tp}{tp+fp} \tag{4}$$

b. *Recall*

Recall ialah kemampuan untuk mengambil peringkat teratas yang sebagian besar relevan (benar). Di bawah ini merupakan rumus perhitungan *recall*.

$$\frac{tp}{tp+fn} \tag{5}$$

c. *F-Measure*

F-measure digunakan untuk mengukur kualitas *recall* dan *precision*. Di bawah ini merupakan rumus perhitungan *precision*.

$$\frac{2 \times (Precision \times Recall)}{(Precision + Recall)} \tag{6}$$

d. *Accuracy*

Accuracy merupakan salah satu perhitungan evaluasi yang mengkombinasikan antara kalimat yang terpilih oleh manusia, kalimat yang tidak terpilih oleh manusia, dan kalimat hasil pengambil sistem. Di bawah ini merupakan proses perhitungan *accuracy*.

$$\frac{tp + tn}{tp + fp + fn + tn} \tag{7}$$

Dimana :

tp = jumlah kalimat yang berhasil diekstrak sistem sesuai dengan kalimat yang diekstrak manusia

fp = jumlah kalimat yang diekstrak sistem tetapi tidak terdapat dalam kalimat yang diekstrak manusia

fn = jumlah kalimat yang diekstrak manusia tetapi tidak terdapat dalam kalimat yang diekstrak sistem.

tn = jumlah kalimat yang tidak diekstrak manusia dan tidak diekstrak sistem.

3.3.1 Pengujian Compression Rate 25%

Pada Tabel 5 merupakan hasil pengujian peringkasan otomatis dengan peringkasan manual oleh pakar pada *compression rate* sebesar 25%.

Tabel 5 Pengujian *compression rate* 25%

No	Dokumen	Precision	Recall	F-measure	Accuracy
1	DOK-1	0,5	0,625	0,556	0,704
2	DOK-2	0,6	0,833	0,698	0,787
3	DOK-3	0,522	0,75	0,615	0,741
4	DOK-4	0,571	0,889	0,696	0,794
5	DOK-5	0,543	0,76	0,633	0,75
6	DOK-6	0,25	0,571	0,348	0,634
7	DOK-7	0,611	0,917	0,733	0,814
8	DOK-8	0,429	0,75	0,545	0,722
9	DOK-9	0,571	1	0,727	0,813
10	DOK-10	0,467	0,778	0,583	0,73
Rata-rata		0,5064	0,7873	0,6134	0,7489
%		51	79	61	75

3.3.2 Pengujian Compression Rate 50%

Pada Tabel 6 merupakan hasil pengujian peringkasan otomatis dengan peringkasan manual oleh pakar pada *compression rate* sebesar 50%.

Tabel 6 Pengujian *compression rate* 50%

No	Dokumen	Precision	Recall	F-measure	Accuracy
1	DOK-1	0,571	0,5	0,533	0,72
2	DOK-2	0,625	0,556	0,588	0,754
3	DOK-3	0,533	0,5	0,516	0,722
4	DOK-4	0,556	0,556	0,556	0,75
5	DOK-5	0,609	0,56	0,583	0,753
6	DOK-6	0,364	0,571	0,444	0,722
7	DOK-7	0,583	0,583	0,583	0,756
8	DOK-8	0,6	0,75	0,667	0,813
9	DOK-9	0,5	0,5	0,5	0,733
10	DOK-10	0,5	0,556	0,526	0,735
Rata-rata		0,5441	0,5632	0,5496	0,7458
%		54	56	55	75

3.3.3 Pengujian *Compression Rate* 75%

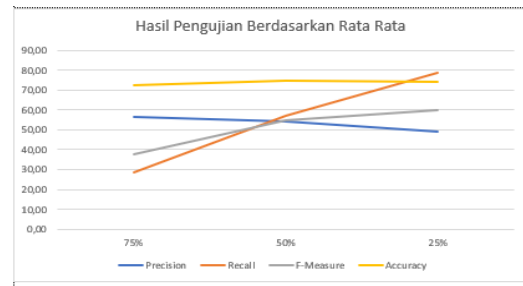
Pada Tabel 7 merupakan hasil pengujian peringkasan otomatis dengan peringkasan manual oleh pakar pada *compression rate* sebesar 75%.

Tabel 7 Pengujian *compression rate* 75%

No	Dokumen	Precision	Recall	F-measure	Accuracy
1	DOK-1	0,667	0,25	0,364	0,696
2	DOK-2	0,75	0,333	0,462	0,736
3	DOK-3	0,5	0,25	0,333	0,686
4	DOK-4	0,2	0,111	0,143	0,625
5	DOK-5	0,833	0,4	0,541	0,77
6	DOK-6	0,6	0,429	0,5	0,806
7	DOK-7	0,333	0,167	0,222	0,65
8	DOK-8	0,5	0,25	0,333	0,733
9	DOK-9	0,5	0,25	0,333	0,714
10	DOK-10	0,4	0,222	0,286	0,688
Rata-rata		0,5283	0,2662	0,3517	0,7104
%		53	27	35	71

3.4 Pengujian berdasarkan rata-rata

Pada Gambar 6 merupakan grafik hasil pengujian *compression rate* 75%, 50% dan 25% berdasarkan rata – rata *precision*, *recall*, *f-measure* dan *accuracy*.



Gambar 6 Pengujian berdasarkan rata-rata

Berdasarkan grafik di atas dapat diketahui bahwa *compression rate* memengaruhi nilai *precision*, *recall* dan *f-measure*. Semakin kecil *compression rate* maka nilai *precision* akan mengalami penurunan. Hal itu terjadi dikarenakan *compression rate* yang rendah menghasilkan lebih banyak kalimat peringkasan pada sistem tetapi tidak termasuk ringkasan menurut pakar. Selanjutnya untuk nilai *recall* akan terus meningkat sejalan dengan menurunnya *compression rate* dikarenakan pada *compression rate* yang rendah akan lebih banyak menghasilkan kalimat yang muncul pada peringkasan oleh sistem yang juga merupakan ringkasan menurut pakar. Pada nilai *f-measure* juga akan mengalami peningkatan apabila *precision* atau nilai *recall* mengalami kenaikan karena nilai *f-measure* merupakan nilai hubungan antara nilai *precision* dan *recall* itu sendiri. Sedangkan untuk *accuracy*, tidak mengalami perubahan yang drastis.

4. Kesimpulan dan Saran

Berdasarkan hasil penelitian dan pengujian yang telah dilakukan dapat ditarik kesimpulan bahwa Metode *Latent Semantic Analysis* dapat digunakan untuk melakukan peringkasan dokumen hukum. Selanjutnya, Pengujian dengan menggunakan *compression rate* berbeda dapat memengaruhi nilai *precision*, *recall*, *f-measure* dan *accuracy*. Hasil *precision*, *recall*, *f-measure* dan *accuracy* secara berurutan pada peringkasan teks otomatis dengan metode *Latent Semantic Analysis* untuk *compression rate* 75% yaitu 53%, 27%, 35% dan 71% lalu untuk *compression rate* 50% yaitu 54%, 56%, 55% dan 75%, serta untuk *compression rate* 25% yaitu 51%, 29%, 61% dan 75%.

Untuk penelitian selanjutnya, Sistem tidak hanya dapat melakukan ekstrak teks dari dokumen berformat pdf tetapi dapat juga meng ekstrak dengan format lain seperti doc/docx. Lalu untuk proses ekstrak teks dapat dilakukan secara dinamis. Apabila terdapat perubahan pada format pengambilan teks pada dokumen, dapat diganti melalui menu aplikasi tanpa harus mengganti pengkodean sistem. Peringkasan otomatis ini juga disarankan untuk dikembangkan menjadi peringkasan

secara abstraktif dengan metode yang berbeda agar dapat dibandingkan tingkat keakurasiannya.

5. Daftar Pustaka

- Ahmad, A. G. (2019, Maret). *Metodologi Pengembangan Perangkat Lunak*. Retrieved from Medium: <https://medium.com/adhub-team/software-development-e025afeb69e4>
- Badan Pusat Statistik. (2018). *Statistik Kriminal 2018*. Badan Pusat Statistik.
- G, S. (1989). *Automatic Text Processing : The Transformation, Analysis, and Retrieval of s.l.:Reading: Addison-Wesley*.
- Gupta, V., & Lehal, G. S. (2010). A Survey of Text Summarization Extractive. *JOURNAL OF EMERGING TECHNOLOGIES IN WEB INTELLIGENCE, VOL. 2, NO. 3*.
- Hidayat, C. (n.d.). *Pengertian Metode Waterfall dan Tahap-Tahapnya*. Retrieved from RanahResearch: <https://ranahresearch.com/metode-waterfall/>
- Hovy, E. (2005). Automated Text Summarization. *Handbook of Computation Linguistics OxfordUniversityPress*.
- Mandar, G., & Gunawan. (2017). Peringkasan dokumen berita Bahasa Indonesia menggunakan Cross Latent Semantic Analysis. *Jurnal Ilmiah Teknologi Sistem Informasi 3 (2) 94-104*.
- Minarni, & Susanti. (2014). Sistem Informasi Inventory Obat Rumah Sakit Umum Daerah (RSUD) Padang.
- N, D., & K, G. J. (2015). Kannada text summarization using Latent Semantic Analysis. *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*.
- Nindito, H. (2016, Desember 15). *Teori Text Mining dan Web Mining*. Retrieved from sis.binus.ac.id: <https://sis.binus.ac.id/2016/12/15/teori-text-mining-dan-web-mining/>
- Peter, R., & Kp, S. (2009). Evaluation of SVD and NMF Methods for Latent Semantic Analysis. *International Journal of Recent Trends in Engineering , Vol 1, No. 3*.
- Radev, D. R. (2002). Introduction to the special issue on summarization. *Journal*, 399-408.
- Sommerville, I. (2011). In *Software Engineering (Ninth Edition)* (pp. 30-31). United States of America: Pearson Education, Inc. Publishing as addison-wesley.
- Tinaliah, & Elizabeth, T. (2018). Perbandingan Hasil Deteksi Plagiarisme Dokumen dengan Metode Jaro-Winkler Distance dan Metode Latent Semantic Analysis. *Jurnal Teknologi dan Sistem Komputer, 6(1)*.
- Tokunaga, T., & Iwayama, M. (1994). *Text Categorization based on Weighted Inverse Document Frequency*. Tokyo, Japan: Tokyo Institute of Technology.

