e-ISSN 2597-4963 dan p-ISSN 1858-4853 DOI : http://dx.doi.org/10.30872/jim.v14i1.1272

Pengujian Akurasi Sistem Rekomendasi Berbasis *Content-Based Filtering*

Wayan Gede Suka Parwita
Teknik Informatika, STMIK STIKOM Indonesia
Jalan Tukad Pakerisan No. 97, Denpasar-Bali, 80225
E-Mail : gede.suka@gmail.com

ABSTRAK

Rekomendasi dosen pembimbing dalam Sintesys memanfaatkan dokumen UPP dan dokumen publikasi atau penelitian calon dosen pembimbing sebagai dasar penentuan rekomendasi. Adapun dengan semakin banyaknya mahasiswa yang mengajukan UPP maka akan meningkatkan jumlah proses yang dilakukan sehingga perlu ditinjau pengaruh dari penerapan *stopword removal* sebagai salah satu proses dalam ektraksi dokumen. Penelitian dimulai dengan analisis sistem yang telah dibangun pada penelitian Parwita sebelumnya. Analisis ini memetakan komponen yang akan digunakan dalam pengujian sistem rekomendasi. Setelah analisis sistem, selanjutnya dilakukan pengumpulan data seperti *stopword*, rekomendasi dosen, dan dokumen penelitian dosen. Setiap dokumen penelitian dosen akan dibandingkan dengan UPP mahasiswa. Nilai tertinggi dari salah satu dokumen penelitian dosen akan digunakan sebagai nilai similaritas antara penelitian dosen dan UPP mahasiswa. Hasil pengujian merupakan nilai precission, recall, dan f-mesure yang digunakan dalam melakukan analisis hasil pengujian. Pengujian sistem rekomendasi dengan proses *stopword* menunjukkan nilai precission, recall, dan f-measure yang berbeda-beda untuk setiap *minimum similarity* yang ditetapkan. Nilai tertinggi pada sistem rekomendasi dengan proses *stopword* untuk precission didapatkan pada *minimum similarity* 30%, recall pada *minimum similarity* 5%, dan f-measure pada *minimum similarity* 20 % dan 25%.

Kata Kunci – *sistem rekomendasi; ekstraksi dokumen; pengujian information retrival;*

1. PENDAHULUAN

Sistem rekomendasi merupakan teknik dan software untuk menghasilkan usulan *item* yang akan dimanfaatkan oleh pengguna. "*Item*" merupakan istilah yang digunakan untuk menyatakan apa yang direkomendasikan oleh sistem kepada pengguna. Usulan tersebut dihasilkan berdasarkan berbagai proses pengambilan keputusan seperti barang apa yang akan dibeli, lagu apa yang ingin didengarkan, dan buku apa yang akan dibaca.

Dalam penentuan rekomendasi diperlukan data ketertarikan pengguna yang dinyatakan secara eksplisit dalam data peringkat *item* atau disimpulkan dengan menebak tindakan pengguna. Dalam rekomendasi dokumen, penentuan rekomendasi dapat dientukan dengan membandingkan kemiripan isi dari dokumen. Ekstraksi isi dokumen dilakukan dengan pemanfaatan metode ektraksi teks yang meliputi tokenisasi, *stemming*, dan *stopword removal*. *Stopword* merupakan kata-kata yang sering muncul atau dianggap tidak penting dalam dokumen. *Stopword* dalam ekstraksi teks akan dihilangkan dalam proses *stopword removal*. Proses ini juga bertujuan untuk mengurangi jumlah kata yang akan diproses selanjutnya.

Rekomendasi dosen pembimbing dalam Sintesys memanfaatkan dokumen UPP dan dokumen publikasi atau penelitian calon dosen pembimbing sebagai dasar penentuan rekomendasi. UPP dan dokumen penelitian atau publikasi ilmiah calon dosen pembimbing merupakan dokumen dengan klasifikasi khusus pada bidang ilmu teknologi informasi. Adapun dengan semakin banyaknya mahasiswa yang mengajukan UPP maka akan meningkatkan jumlah proses yang dilakukan.

Berdasarkan latar belakang di atas, maka dalam penelitian ini akan ditinjau pengaruh *stopword* dalam tingkat akurasi penentuan rekomendasi dokumen dengan klasifikasi dokumen yang digunakan adalah UPP dan dokemen publikasi atau penelitian calon dosen pembimbing.

2. TINJAUAN PUSAKA

A. Ekstraksi *Keyword*

Dalam dokumen ilmiah, *keyword* adalah kata pokok yang merepresentasikan masalah yang diteliti atau istilah-istilah yang merupakan dasar pemikiran dan dapat berupa kata tunggal atau gabungan kata. Similaritas *keyword* dokumen dapat digunakan untuk menentukan relevansi dokumen terhadap dokumen lain (Weiss, Indurkhya, Zhang, & Damerau, 2005). Automatic *keyword extraction system* memiliki tugas untuk mengidentifikasi kumpulan kata, frase kunci, *keyword*, atau segmen kunci dari sebuah dokumen yang dapat menggambarkan arti dari dokumen (Hulth, 2003). Tujuan dari ekstraksi otomatis adalah menekan kelemahan pada ekstraksi manual yang dilakukan manusia yaitu pada kecepatan, ketahanan, cakupan, dan juga biaya yang dikeluarkan.

Salah satu pendekatan yang dapat digunakan dalam automatic *keyword extraction* yaitu pendekatan tata bahasa. Pendekatan ini menggunakan fitur tata bahasa dari kata-kata, kalimat, dan dokumen. Metode ini memperhatikan fitur tata bahasa seperti bagian kalimat, struktur sintaksis, dan makna yang dapat menambah bobot. Fitur tata bahasa tersebut dapat digunakan sebagai penyaring untuk *keyword* yang buruk. Dalam ekstraksi *keyword* dengan pendekatan tata bahasa berbasis struktur sintaksis, ada beberapa tahap yang dilakukan yaitu

e-ISSN 2597-4963 dan p-ISSN 1858-4853 DOI : http://dx.doi.org/10.30872/jim.v14i1.1272

tokenisasi, *stopword removal*, *stemming*, dan pembobotan kata (Oelze, 2009).

B. Tokenisasi

Teks elektronik adalah urutan linear simbol (karakter, kata-kata atau frase). Sebelum dilakukan pengolahan, teks perlu disegmentasi ke dalam unit-unit linguistik seperti kata-kata, tanda baca, angka, alpha-numeric, dan lain-lain Proses ini disebut tokenisasi. Tokenisasi sederhana (*white space tokenization*) merupakan tokenisasi yang memisahkan kata berdasarkan karakter spasi, tab, dan baris baru (Weiss et al., 2005). Namun, tidak setiap bahasa melakukan hal ini (misalnya bahasa Cina, Jepang, Thailand). Dalam bahasa Indonesia, selain tokenisasi sederhana diperlukan juga tokenisasi yang memisahkan kata-kata berdasarkan karakter lain seperti "/" dan "-".

C. *Stopword Removal*

Stopword removal adalah pendekatan mendasar dalam preprocessing yang menghilangkan kata-kata yang sering muncul (*stopword*). Fungsi utamanya adalah untuk mencegah hasil proses selanjutnya terpengaruh oleh *stopword* tersebut. Banyak diantara *stopword* tersebut tidak berguna dalam Information Retrival (IR) dan text mining karena kata-kata tersebut tidak membawa informasi (seperti ke, dari, dan, atau). Cara biasa untuk menentukan apa yang dianggap sebagai *stopword* adalah menggunakan stoplist. Stoplist merupakan kumpulan kata atau kamus yang berisi daftar *stopword*. Langkah penghilangan *stopword* ini adalah langkah yang sangat penting dan berguna (Srividhya & Anitha, 2010).

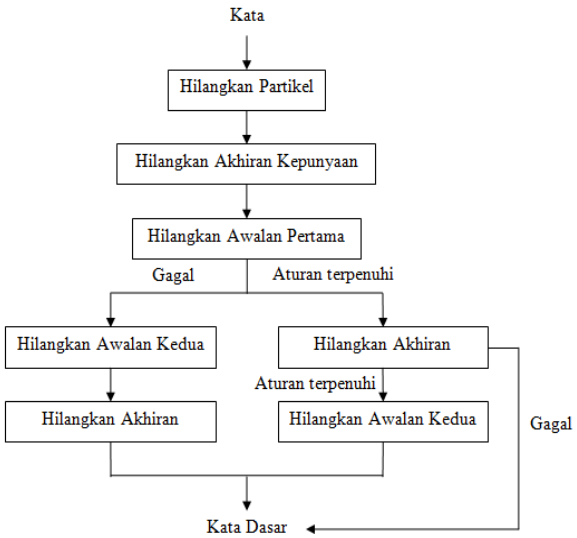
D. *Stemming*

Algoritma *stemming* adalah proses yang melakukan pemetaan varian morfologi yang berbeda dari kata-kata ke dalam kata dasar/kata umum (*stem*). *Stemming* berguna pada banyak bidang komputasi linguistik dan information retrieval (Lovins, 1968). Dalam kasus bahasa Indonesia, sejauh ini hanya ada dua algoritma untuk melakukan proses *stemming* yaitu algoritma yang dikembangkan oleh Nazief dan Adriani serta algoritma yang dikembangkan oleh Tala. Algoritma Nazief dan Adriani dikembangkan dengan menggunakan pendekatan confix stripping dengan disertai pemindaian pada kamus. Sedangkan *stemming* yang dikembangkan Tala menggunakan pendekatan yang berbasis aturan (*rule-based*).

Pengembangan Algoritma Tala didasarkan pada kenyataan bahwa sumber daya seperti kamus besar digital untuk bahasa mahal karena kurangnya penelitian komputasi di bidang linguistik. Maka, ada kebutuhan untuk algoritma *stemming* tanpa keterlibatan kamus. Algoritma Tala sendiri dikembangkan dari algoritma Porter stemmer yang dimodifikasi untuk bahasa Indonesia. Algoritma Tala menghasilkan banyak kata yang tidak dipahami. Ini disebabkan oleh ambiguitas dalam aturan morfologi Bahasa Indonesia. Dalam beberapa kasus kesalahan tidak mempegaruhi kinerja, tetapi dalam kasus lain menurunkan kinerja (Tala, 2003).

Algoritma Tala memproses awalan, akhiran, dan kombinasi keduanya dalam kata turunan. Walaupun dalam bahasa Indonesia terdapat sisipan, jumlah kata yang diturunkan menggunakan sisipan sangat sedikit. Karena hal tersebut dan juga demi penyederhanaan, sisipan akan diabaikan.

Algoritma Porter *stemmer* dibangun berdasarkan ide tentang akhiran pada bahasa Inggris yaitu kebanyakan merupakan kombinasi dari akhiran yang lebih sederhana dan lebih kecil. Beberapa perubahan dilakukan pada algoritma Porter *stemmer* agar sesuai dengan Bahasa Indonesia. Perubahan dilakukan pada bagian kumpulan aturan dan penilaian kondisi. Karena algoritma Porter *stemmer* hanya dapat menangani akhiran, maka perlu penambahan agar dapat menangani awalan, akhiran, dan juga penyesuaian penulisan dalam kasus dimana terjadi perubahan karakter pertama kata dasar. Gambar 1 menunjukkan langkah-langkah proses pada algoritma Tala.



Gambar 1. Algoritma Tala

Dalam Bahasa Indonesia, unit terkecil dari suatu kata adalah suku kata. Suku kata paling sedikit terdiri dari satu huruf vokal. Desain implementasi algoritma Tala belum dapat mengenali seluruh suku kata. Ini disebabkan karena adanya dua huruf vokal yang dianggap satu suku kata yaitu ai, au, dan oi. Kombinasi dua huruf vokal (terutama ai, oi) tersebut dapat menjadi masalah, apalagi jika berada pada akhir sebuah kata. Ini disebabkan oleh sulitnya membedakannya dengan kata yang mengandung akhiran –i. Hal ini menyebabkan kombinasi huruf vokal ai/oi akan diperlakukan seperti kata turunan. Huruf terakhir (-i) akan dihapus pada hasil proses *stemming*. Kebanyakan kata dasar terdiri dari minimal dua suku kata. Inilah alasan kenapa kata yang akan diproses memiliki minimal dua suku kata.

E. Pembobotan

Tahapan ini dilakukan dengan tujuan untuk memberikan suatu bobot pada term yang terdapat pada suatu dokumen. Term adalah satu kata atau lebih yang dipilih langsung dari corpus dokumen asli dengan menggunakan metode term-*extraction*. Fitur tingkat term, hanya terdiri dari kata-kata tertentu dan

e-ISSN 2597-4963 dan p-ISSN 1858-4853 DOI : http://dx.doi.org/10.30872/jim.v14i1.1272

ekspresi yang d*item*ukan dalam dokumen asli (Feldman & Sanger, 2006).

Dalam pengkategorian teks dan aplikasi lain di information retrieval maupun machine learning, pembobotan term biasanya ditangani melalui metode yang diambil dari metode pencarian teks, yaitu yang tidak melibatkan tahap belajar (Debole & Sebastiani, 2003). Ada tiga asumsi monoton yang muncul di hampir semua metode pembobotan dapat dalam satu atau bentuk lain yaitu (Zobel & Moffat, 1998):

a. Term yang langka tidak kalah penting daripada term yang sering muncul (asumsi IDF).
b. Kemunculan berkali-kali dari term pada dokumen tidak kalah penting daripada kemunculan tunggal (asumsi TF).
c. Untuk pencocokan term dengan jumlah pencocokan yang sama, dokumen panjang tidak lebih penting daripada dokumen pendek (asumsi normalisasi).

Bobot diperlukan untuk menentukan apakah term tersebut penting atau tidak. Bobot yang diberikan terhadap sebuah term bergantung kepada metode yang digunakan untuk membobotinya.

F. *Cosine Similarity*

Pendekatan *cosine similarity* sering digunakan untuk mengetahui kedekatan antara dokumen teks. Perhitungan *cosine similarity* dimulai dengan menghitung *dot product*. *Dot product* merupakan perhitungan sederhana untuk setiap komponen dari kedua vektor. Vektor merupakan representasi dari masing-masing dokumen dengan jumlah *term* pada masing-masing dokumen sebagai dimensi dari vektor (Manning, Ragahvan, & Schutze, 2009). Vektor ditunjukkan oleh notasi (1) dan (2). Hasil *dot product* bukan berupa vektor tetapi berupa skalar. Persamaan (3) merupakan perhitungan *dot product* dimana n merupakan dimensi dari vektor (Axler, Gehring, & Ribet, 1997).

$$\vec{a} = (a_1, a_2, a_3, \dots, a_n) \quad \dots\dots (1)$$

$$\vec{b} = (b_1, b_2, b_3, \dots, b_n) \quad \dots\dots (2)$$

$$\vec{a} \cdot \vec{b} = \sum_{i=1}^{n} a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n \quad \dots (3)$$

a_n dan b_n merupakan komponen dari vektor (bobot *term* masing-masing dokumen) dan n merupakan dimensi dari vektor. *Cosine similarity* merupakan perhitungan yang mengukur nilai *cosine* dari sudut antara dua vektor (atau dua dokumen dalam *vector space*). *Cosine similarity* dapat dilihat sebagai perbandingan antara dokumen karena tidak hanya mempertimbangkan besarnya masing-masing jumlah kata (bobot) dari setiap dokumen, tetapi sudut antara dokumen. Persamaan (4) dan (5) adalah notasi dari metode *cosine similarity* dimana $\|\vec{a}\|$ merupakan *Euclidean norm* dari vektor a dan $\|\vec{b}\|$ merupakan E*uclidean norm* vektor b (Han, Kamber, & Pei, 2012).

$$\vec{a} \cdot \vec{b} = \|\vec{a}\| \|\vec{b}\| \cos\theta \quad \dots\dots (4)$$

$$\cos\theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \|\vec{b}\|} \quad \dots\dots (5)$$

Dari notasi (5) dapat dibentuk persamaan matematika yang ditunjukkan oleh persamaan (6).

$$Similarity(x, y) = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2 \cdot \sum_{i=1}^{n} b_i^2}} \quad (6)$$

Dimana:
a_i : term ke-i yang terdapat pada dokumen a.
b_i : term ke-i yang terdapat pada dokumen b.

G. *Precision*

Precision bersama *recall* merupakan salah satu pengujian dasar dan paling sering digunakan dalam penentuan efektifitas *information retrival system* maupun *recommendation system*. *True positive* (tp) pada *information retrival* merupakan *item* relevan yang dihasilkan oleh sistem. Sedangkan *false positive* (fp) merupakan semua *item* yang dihasilkan oleh sistem. Sehingga dalam *information retrival*, *precision* dihitung dengan persamaan (7) (Manning et al., 2009).

$$Precision = \frac{tp}{tp + fp} = \frac{relevant\ item\ retrieved}{retrieved\ item} \quad .. (7)$$

Istilah *positive* dan *negative* mengacu pada prediksi yang dilakukan oleh sistem. Sedangkan istilah *true* dan *false* mengacu pada prediksi yang dilakukan oleh pihak luar atau pihak yang melakukan observasi. Pembagian kondisi tersebut dapat dilihat pada Tabel 1 (Manning et al., 2009).

Tabel 1. Pembagian kondisi hasil yang memungkinkan

	Relevant	*Nonrelevant*
Retrieved	*True positive* (tp)	*False positive* (fp)
Not retrieved	*False negative* (fn)	*True negative* (tn)

H. *Recall*

Recall digunakan sebagai ukuran dokumen yang relevan yang dihasilkan oleh sistem. *False negative* (fn) merupakan semua *item* relevan yang tidak dihasilkan oleh sistem. Dalam evaluasi *information retrival system*, *recall* dihitung dengan persamaan (8) (Manning et al., 2009).

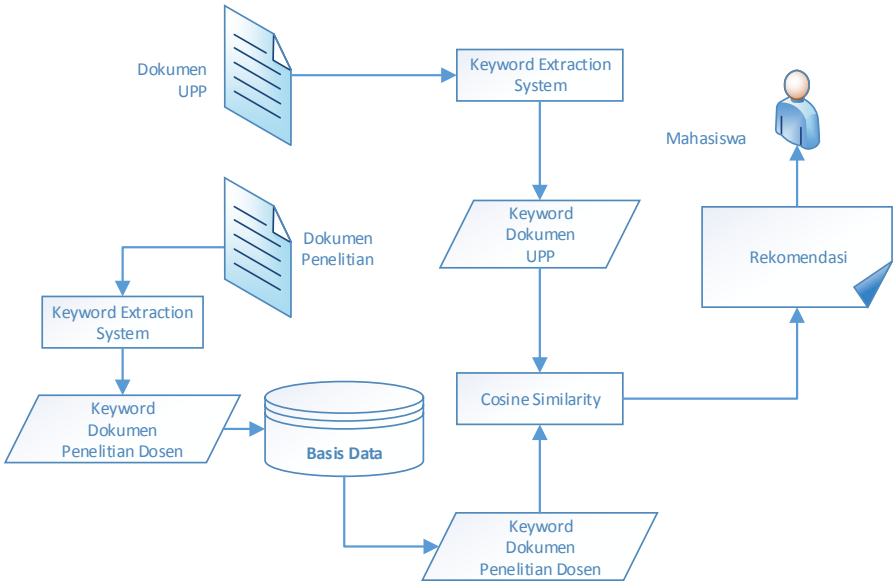
$$Recall = \frac{tp}{tp + fn} = \frac{relevant\ item\ retrieved}{relevant\ item} \quad .. (8)$$

I. F-Measure

F-measure merupakan nilai tunggal hasil kombinasi antara nilai *precision* dan nilai *recall*. F-measure dapat digunakan untuk mengukur kinerja dari *recommendation system* ataupun *information retrival system*. Karena merupakan rata-rata harmonis dari *precision* dan *recall*, F-measure dapat memberikan penilaian kinerja yang lebih seimbang. Persamaan (9) merupakan persamaan untuk menghitung F-measure (Jannach, Zanker, Felfernig, & Friedrich, 2011).

$$F_{measure} = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad \dots\dots (9)$$

e-ISSN 2597-4963 dan p-ISSN 1858-4853 DOI : http://dx.doi.org/10.30872/jim.v14i1.1272



Gambar 2. Gambaran umum sistem rekomendasi

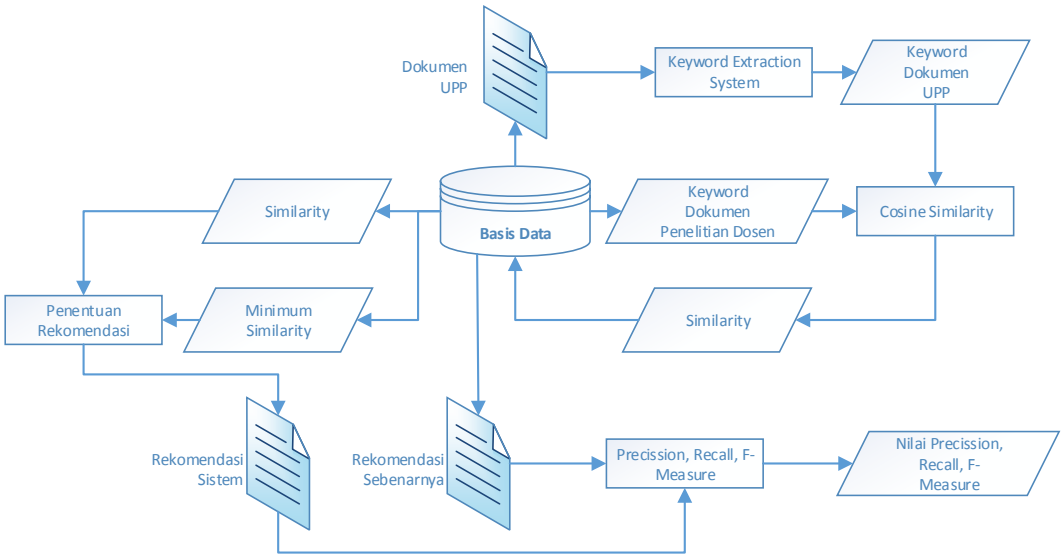
3. METODE PENELITIAN

A. Tahapan Pengujian

Penelitian dimulai dengan analisis sistem yang telah dibangun pada penelitian (Parwita et al., 2018). Analisis ini memetakan komponen yang akan digunakan dalam pengujian sistem rekomendasi. Setelah analisis sistem, selanjutnya dilakukan pengumpulan data seperti *stopword* yang diambil dari penelitian Tala (Tala, 2003), rekomendasi dosen, dan dokumen penelitian dosen. Implementasi pengujian dilakukan dengan membangun algoritma pengujian memanfaatkan komponen sistem rekomendasi yang telah dibangun.

B. Analisis Sistem

Pembangunan sistem rekomendasi disesuaikan dengan fasilitas yang ada pada STMIK STIKOM Indonesia. Dokumen UPP maupun penelitian dosen diektraksi untuk menemukan *keyword* yang ada pada setiap dokumen. Proses ekstraksi melalui beberapa tahap yaitu tokenisasi, *stopword removal*, *stemming* dan pembobotan. Proses *stopword removal* menggunakan *stopword* yang pada penelitian Tala (Tala, 2003) dan proses *stemming* akan memanfaatkan algoritma yang dikembangkan oleh Tala (Tala, 2003). *Keyword* dari dokumen penelitian dosen akan disimpan dalam basis data agar ekstraksi tidak dilakukan berulang-ulang. Secara umum, sistem yang akan dibangun ditunjukkan oleh Gambar 2. Pengembangan sistem akan dibagi menjadi 3 tahap, yaitu tahap pembangunan sistem ekstraksi *keyword* dengan menyesuaikan bentuk sistem



Gambar 3. Pengujian sistem rekomendasi

C. Implementasi Pengujian

Pengujian diimplementasikan dengan menggunakan komponen-komponen pada sistem ekstraksi dengan dokumen yang ada, pengembangan sistem rekomendasi dengan memanfaatkan hasil ekstraksi *keyword*.

e-ISSN 2597-4963 dan p-ISSN 1858-4853 DOI : http://dx.doi.org/10.30872/jim.v14i1.1272

rekomendasi yang telah dibangun. Hasil rekomendasi akan dibandingkan dengan rekomendasi nyata yang telah diberikan dalam bentuk kuesioner. Pengujian dilakukan dengan menghitung Precission, Recall, dan F-Measure antara hasil rekomendasi dengan rekomendasi sebenarnya. Pengujian rekomendasi yang dihasilkan sistem dimulai dengan mencari nilai similaritas antara setiap dokumen penelitian dengan masing-masing dokumen UPP. Pada saat dosen memiliki penelitian lebih dari 1 (satu), maka nilai similaritas tertinggi yang akan digunakan sebagai nilai similaritas dosen dengan mahasiswa. Gambar 3 merupakan gambaran umum dari pengujian yang dilakukan

D. Skenario Pengujian

Setiap dokumen penelitian dosen akan dibandingkan dengan UPP mahasiswa. Nilai tertinggi dari salah satu dokumen penelitian dosen akan digunakan sebagai nilai similaritas antara penelitian dosen dan UPP mahasiswa. Nilai similaritas digunakan untuk menentukan rekomendasi yang dihasilkan oleh sistem. Hasil rekomendasi sistem akan dibandingkan dengan hasil rekomendasi sesungguhnya yang didapatkan melalui kuesioner yang telah dilakukan. Perbandingan ini akan menghasilkan nilai *precission*, *recall*, dan f-measure masing-masing dosen. Nilai masing-masing dosen lalu dijumlahkan dan dibagi dengan jumlah dosen untuk mencari rata-rata nilai *precission*, *recall*, dan f-measure. Nilai rata-rata inilah yang akan digunakan untuk mengukur kualitas dari rekomendasi.

4. HASIL DAN PEMBAHASAN

Pengujian sistem rekomendasi dengan proses *stopword* menunjukkan nilai *precission*, *recall*, dan f-measure yang berbeda-beda untuk setiap *minimum similarity* yang ditetapkan. Tabel 2 merupakan nilai pengujian sistem rekomendasi dengan menggunakan proses *stopword removal*.

Tabel 2. Hasil pengujian sistem rekomendasi

Minimum similarity (%)	**Dengan *Stopword Removal***		
	Precission	**Recall**	**F-Measure**
5	0,0797	0,9870	0,1390
10	0,1142	0,8394	0,1855
15	0,1588	0,6987	0,2325
20	0,2882	0,6245	0,3202
25	0,3455	0,4816	0,3131
30	0,3903	0,3495	0,2782
35	0,2291	0,1908	0,1577
40	0,2402	0,1369	0,1245
45	0,2068	0,0991	0,1128

Ada perbedaan jika dibandingkan dengan hasil pengujian sistem rekomendasi tanpa proses *stopword removal*. Nilai tertinggi pada sistem rekomendasi tanpa proses *stopword* untuk *precission* didapatkan pada *minimum similarity* 25% mencapai 0,3506, *recall* pada *minimum similarity* 5% mencapai 0,9361, dan f-measure pada *minimum similarity* 15% mencapai 0,2965. Hasil pengujian sistem rekomendasi tanpa proses *stopword removal* dapat dilihat pada Tabel 3.

Tabel 3. Hasil pengujian tanpa proses *stopword removal*

***Minimum similarity* (%)**	**Tanpa *Stopword Removal***		
	Precission	**Recall**	**F-Measure**
5	0,0815	0,9361	0,1418
10	0,1319	0,7235	0,2015
15	0,2519	0,5689	0,2965
20	0,3091	0,3781	0,2745
25	0,3506	0,2275	0,2181
30	0,1909	0,1002	0,0984
35	0,1995	0,0608	0,0799
40	0,1136	0,0122	0,0205
45	0,0606	0,0103	0,0174

5. KESIMPULAN

Setelah melakukan pengujian terhadap sistem rekomendasi yang dibangun, maka dapat diperoleh kesimpulan yaitu nilai tertinggi sistem rekomendasi tanpa proses *stopword removal* untuk *precission* didapatkan pada *minimum similarity* 25% mencapai 0,3506, *recall* pada *minimum similarity* 5% mencapai 0,9361, dan f-measure pada *minimum similarity* 15% mencapai 0,2965.

Adapun jika dibandingkan dengan sistem rekomendasi dengan proses *stopword removal* didapatkan bahwa nilai sistem rekomendasi dengan proses *stopword removal* masih lebih unggul dibandingkan sistem rekomendasi tanpa proses *stopword removal*. Hal ini tampak pada nilai *precission*, *recall*, dan f-measure yang semua dicapai pada sistem rekomendasi dengan proses *stopword removal* yaitu *precission* dengan nilai 0,3903 pada *minimum similarity* 30%, *recall* 0,9870 pada *minimum similarity* 5%, dan f-measure 0,3202 pada *minimum similarity* 20%.

6. DAFTAR PUSTAKA

Axler, S., Gehring, F., & Ribet, K. (1997). Linear Algebra Done Right, Second Edition. *Computer Engineering*, xv, 251. https://doi.org/10.1007/b97662

Debole, F., & Sebastiani, F. (2003). Supervised term weighting for automated text categorization. In *Proceedings of the 2003 ACM symposium on Applied computing - SAC '03* (p. 784). https://doi.org/10.1145/952686.952688

Feldman, R., & Sanger, J. (2006). *The Text Mining Handbook*. https://doi.org/10.1017/CBO9780511546914

Han, J., Kamber, M., & Pei, J. (2012). *Data Mining: Concepts and Techniques*. *San Francisco, CA, itd: Morgan Kaufmann*. https://doi.org/10.1016/B978-0-12-381479-1.00001-0

Hulth, A. (2003). Improved Automatic Keyword Extraction Given More Linguistic Knowledge. *Language*, (2000), 216–223. https://doi.org/10.3115/1119355.1119383

e-ISSN 2597-4963 dan p-ISSN 1858-4853 DOI : http://dx.doi.org/10.30872/jim.v14i1.1272

Jannach, D., Zanker, M., Felfernig, A., & Friedrich, G. (2011). *Recommender systems: an introduction*. *Cambridge University Press*. https://doi.org/10.1017/CBO9780511763113

Lovins, J. B. (1968). Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, *11*(June), 22–31. Retrieved from http://journal.mercubuana.ac.id/data/MT-1968-Lovins.pdf

Manning, C. D., Ragahvan, P., & Schutze, H. (2009). *An Introduction to Information Retrieval*. *Information Retrieval*. https://doi.org/10.1109/LPT.2009.2020494

Oelze, I. (2009). Automatic Keyword Extraction for Database Search. *L3Sde*, 17–20. Retrieved from http://www.l3s.de/~demidova/students/thesis_oelze.pdf

Parwita, W. G. S., Swari, M. H. P., & Welda, W. (2018). Perancangan Sistem Rekomendasi Dokumen Dengan Pendekatan Content-Based Filtering. *CESS(Journal of Computer Engineering System and Science)*, *3*(1), 65–74.

Srividhya, V., & Anitha, R. (2010). Evaluating preprocessing techniques in text categorization. *International Journal of Computer Science and Application*, (2010), 49–51. Retrieved from http://www.sinhgad.edu/IJCSA-2012/pdfpapers/1_11.pdf

Tala, F. Z. (2003). A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia. *M.Sc. Thesis, Appendix D*. https://doi.org/10.22146/teknosains.26972

Weiss, S. M., Indurkhya, N., Zhang, T., & Damerau, F. J. (2005). *Text mining: Predictive methods for analyzing unstructured information*. *Text Mining: Predictive Methods for Analyzing Unstructured Information*. https://doi.org/10.1007/978-0-387-34555-0

Zobel, J., & Moffat, A. (1998). Exploring the similarity space. *ACM SIGIR Forum*. https://doi.org/10.1145/281250.281256