Peta Pikiran Otomatis Teks Berbahasa Indonesia Menggunakan *Word Co-occurrence* dan Bobot Kalimat

Automatic Indonesian Text Mind Map Using Word Co-ocurrence and Sentence's Weight

Dika Atrariksa [1], Dede Rohidin[2], Gia Septiana[3]

[1,2,3,] Departemen Teknik Informatika Universitas Telkom, Bandung
[1]atrariksa@gmail.com, [2]ddr@ittelkom.ac.id , [3]gia@ittelkom.ac.id

Abstrak

Pada penelitian ini dibuat sistem yang men-generate peta pikiran yang isi cabang-cabangnya merupakan kata kunci hasil ekstraksi dengan menggunakan word co-occurrence statistical information. Penelitian ini menyajikan perbandingan antara peta pikiran hasil sistem dengan pembobotan kalimat dan peta pikiran hasil sistem tanpa pembobotan kalimat. Penggunaan pembobotan kalimat adalah untuk melihat pengaruhnya terhadap kata kunci-kata kunci cabang-cabang peta pikiran.

Sistem mampu men-generate peta pikiran dari sebuah dokumen, baik dengan maupun tanpa pembobotan kalimat. Sistem tanpa pembobotan kalimat menghasilkan peta pikiran dengan rata-rata jumlah kata kunci penting cabang utama sebesar 75% dan rata-rata jumlah cabang anak relevan dengan cabang utamanya 37,52%. Sedangkan, sistem dengan pembobotan kalimat menghasilkan peta pikiran dengan rata-rata jumlah kata kunci penting cabang utama sebesar 70,83% dan rata-rata jumlah cabang anak yang penting dan relevan dengan cabang utamanya 34,61%.

Kata Kunci : peta pikiran, word co-occurrence, pembobotan kalimat, ekstraksi kata kunci

Abstract

In this study, a mind map generator has been built. The branches of mind map are filled with keywords that produced by using word co-occurrence statistical information. This study presents the results of a comparison between mind map produced by system with sentence weighting and mind map produced by system without sentence weighting. The use of sentence weighing is to see its effect on branches of mind map.

The system is able to generate a mind map from a document, either with or without sentence weighting. System without sentence weighting generates a mind map with the average number of important keywords as main branch by 75% and the average number of important keywords as children branches and relevant to the main branch by 37.52%. Whereas, the other system has 70.83% and 34.61%..

Keyword : mind map, word co-occurrence, sentence weighting, keyword extraction

1. Pendahuluan

Peta pikiran adalah cara termudah untuk menempatkan informasi ke otak dan mengambil informasi ke luar otak[1]. Peta pikiran memiliki struktur alami yang memancar dari pusat[1]. Sebuah peta pikiran dibuat dengan garis lengkung, symbol, kata, dan gambar yang sesuai dengan satu rangkaian aturan yang sederhana, mendasar, alami, dan sesuai dengan cara kerja otak[1].

Rancang Bangun Sistem Dream Share – Aplikasi Pembuat Peta Pikiran Berbasis Pembelajaran Kolaborasi adalah riset terkait tetntang pembuatan sistem yang mampu membangkitkan peta pikiran dari dokumen yang dilakukan oleh Yohanda Mandala (2013). Riset tersebut menghasilkan peta pikiran dengan kesesuaian cabang-cabang sebesar 52,69%[12].

Pada penelitian kali ini akan dibuat sistem yang mampu men-generate peta pikiran dari teks. Penelitian ini bertujuan mencari kombinasi pembobotan kalimat yang paling baik. Selain itu, pada penelitian ini dicari parameter untuk submodul ekstraksi kata kunci yang hasilnya dijadikan sebagai cabang-cabang peta pikiran. Dan, penelitian ini juga bertujuan untuk menganalisis pengaruh pembobotan kalimat terhadap cabang-cabang peta pikiran.

Penelitian lebih lanjut sangat diharapkan, dimana dokumen yang digunakan tidak hanya dokumen tunggal, tetapi multi dokumen. Dan, dicari metode lain dalam pemilihan cabang anak agar relevansi dengan cabang utamanya semakin meningkat.

2. Landasan Teori

2.1 Mind Map Generator

Pada penelitian ini akan dibuat sistem yang mampu men-generate peta pikiran dari suatu dokumen dengan dan tanpa pembobotan kalimat. Perbedaan proses antara keduanya adalah kalimat yang menjadi input langkah pembuatan peta pikiran. Sistem tanpa pembobotan kalimat memproses seluruh kalimat dari teks yang diinputkan. Sistem dengan pembobotan kalimat memproses kalimat hasil pembobotan kalimat.

Berikut adalah langkah-langkah yang diimplementasikan untuk membangun peta pikiran. Pertama, kata kunci untuk cabang utama dicari dengan input semua kalimat. Kedua, kalimat-kalimat yang memuat suatu cabang utama diambil. Ketiga, bersihkan kalimat-kalimat tersebut dari semua kata-kata cabang utama kecuali cabang utama yang sedang diproses. Keempat, dari kalimat-kalimat tersebut dicari kata kunci untuk mengisi cabang-cabang anak bagi cabang utama yang sedang diproses. Kelima, ulangi langkah dua dan seterusnya sebanyak jumlah cabang utama.

Sistem men-*generate* maksimal delapan cabang utama dan maksimal lima cabang anak untuk setiap cabang utama. Formula untuk mengkalkulasi hasil sistem dalam men-*generate* cabang utama adalah :

— (1)

dengan adalah rata-rata kata penting cabang utama, adalah jumlah kata penting cabang utama, dan adalah jumlah cabang utama yang dihasilkan sistem. Formula untuk mengkalkulasi hasil sistem dalam men-*generate* cabang anak adalah :

— (2)

dengan adalah rata-rata kata penting cabang anak yang relevan dengan cabang utamanya, adalah jumlah kata penting cabang anak dan relevan dengan cabang utamanya, dan adalah jumlah cabang anak yang dihasilkan sistem.

2.2 Peta Pikiran

Peta pikiran adalah cara termudah untuk menempatkan informasi ke otak dan mengambil informasi ke luar otak[1]. Peta pikiran memiliki struktur alami yang memancar dari pusat[1]. Sebuah peta pikiran dibuat dengan garis lengkung, symbol, kata, dan gambar yang sesuai dengan satu rangkaian aturan yang sederhana, mendasar, alami, dan sesuai dengan cara kerja otak[1].
Berikut di bawah ini langkah membuat mind map (sumber : thinkbuzan.com/how-to-mind-map).

1. Buat sebuah ide pusat
   Ide pusat adalah awal dari peta pikiran yang akan dibuat dan merepresentasikan topik yang akan dieksplorasi. Ide pusat harus berada di tengah-tengah peta pikiran.
2. Tambahkan cabang-cabang dari ide pusat
   Langkah selanjutnya adalah membuat cabang-cabang utama dari ide pusat. Cabang-cabang ini adalah tema kunci.
3. Tambahkan keywords pada cabang-cabang
   Tambahkan kata kunci pada setiap cabang. Dan kata kunci sebaiknya terdiri dari satu kata.
4. Beri warna yang berbeda pada setiap cabang
   Warnai cabang-cabang utama dengan warna yang berbeda. Dan warna cabang utama lebih dalam dibanding warna cabang selanjutnya.
5. Masukkan gambar
   Gambar memiliki muatan informasi yang lebih banyak dibandikan dengan satu kata.

2.3 Automated Keyword Extraction Using Word Cooccurrence

Pada penelitian ini, dua kata saling co-occur bila muncul bersamaan di suatu kalimat tanpa memperhatikan urutan kemunculannya[3]. Dan,

derajat bias co-occurrence suatu kata dengan frequent terms dapat digunakan sebagai sebuah indikator kepentingan kata[3]. Sehingga, word co-occurrence statistical information diimplementasikan pada keyword extraction. Ini mengacu pada penelitian Matsuo (2003) tentang "Keyword Extraction from a Single Document using Word Co-occurrence Statistical Information" menunjukkan bahwa algoritma ekstraksi kata kunci dengan word co-occurrence buatannya dapat mendeteksi kata kunci "tersembunyi" dan masih dapat dibandingkan dengan tf-idf untuk ekstraksi kata kunci (keyword) pada dokumen tunggal dan tanpa korpus.

2.3.1 Clustering

Metode *clustering* yang diimplementasikan adalah *Jensen-Shannon Divergence* (JSD) dan *Mutual Information* (MI). *clustering* dilakukan untuk mengelompokkan apakah dua kata yang sedang diproses ada di *cluster* yang sama. Berikut ini adalah formula JSD[3],

( ) – * ((|) (|))
((|)) ((|))+ (3)

dengan ( ) dan (|) ( ) ( ). Dan formula MI didefinisikan[3],

(|) ( ) (4)

dengan adalah semua kata unik yang lolos *remove stopword*.

Threshold yang digunakan kedua metode sudah didapatkan yaitu 0.85 * Log(2) untuk JSD dan 9.1 * Log(2) untuk MI.

2.3.2 *Chi Square*

Chi Square yang digunakan adalah *pearson's chi squared test*. Formula ini secara umum diaplikasikan untuk mengevaluasi seberapa besar perbedaan data yang diamati antara set yang muncul secara kebetulan. Implementasi pada penelitian ini menggunakan formula yang telah diimprove oleh Matsuo. Formula tersebut sebagai berikut[3] :

( )
Σ {(( ))} {(( )
)}

(5)

dengan , adalah total kata yang *co-occur* dengan . Dan, ( ) adalah frekuensi setiap kata yang co-occur dengan .

Chi square di sini digunakan untuk menghitung index bias *co-occurrence* setiap kata dan bukan untuk tes hipotesis, sama seperti yang dilakukan Matsuo.

2.4 Pembobotan Kalimat

Pada penelitian ini, pembobotan kalimat menggunakan fitur posisi kalimat dalam paragraf, fitur kemiripan kalimat dengan judul, dan fitur kemunculan cue words.

Penggunaan fitur posisi kalimat mengacu struktur paragraf deduktif. Paragraf adalah unit terkecil sebuah karangan yang terdiri dari kalimat pokok atau gagasan utama dan kalimat penjelas atau gagasan penjelas[5]. Paragraf deduktif adalah paragraf yang kalimat utamanya terletak pada awal paragraf[5].

Dengan struktur paragraf deduktif, jika ada paragraf yang terbentuk dari empat kalimat, maka kalimat pertama memiliki nilai 4/4, kalimat kedua memiliki nilai 3/4, kalimat ketiga memiliki nilai 2/4, dan kalimat keempat memiliki nilai ¼ yang kemudian akan dikalikan dengan bobot fitur posisi kalimat.

Selanjutnya, penggunaan fitur kemiripan kalimat dengan judul didasari penelitian Aristoteles (2011) tentang "Pembobotan Fitur Pada Peringkasan Teks Bahasa Indonesia Menggunakan Algoritme Genetika". Penelitian tersebut menyebutkan bahwa fitur kalimat yang menyerupai judul adalah fitur dengan bobot tertinggi.

2.5 Rouge

ROUGE adalah singkatan dari Recall Oriented Understudy for Gisting Evaluation[14]. Rouge mengukur untuk secara otomatis menentukan kualitas ringkasan dengan membandingkannya dengan yang lain (ideal) ringkasan yang dibuat oleh manusia[14].

Rouge yang digunakan dalam mengukur kualitas hasil ekstraksi kalimat pada penelitian ini adalah Rouge-N. Rouge-N adalah n-gram calon ringkasan dan satu set ringkasan referensi (pembanding)[14]. Di bawah ini formula Rouge-N[14] :

Σ * +Σ ( )

Σ * +Σ ( )

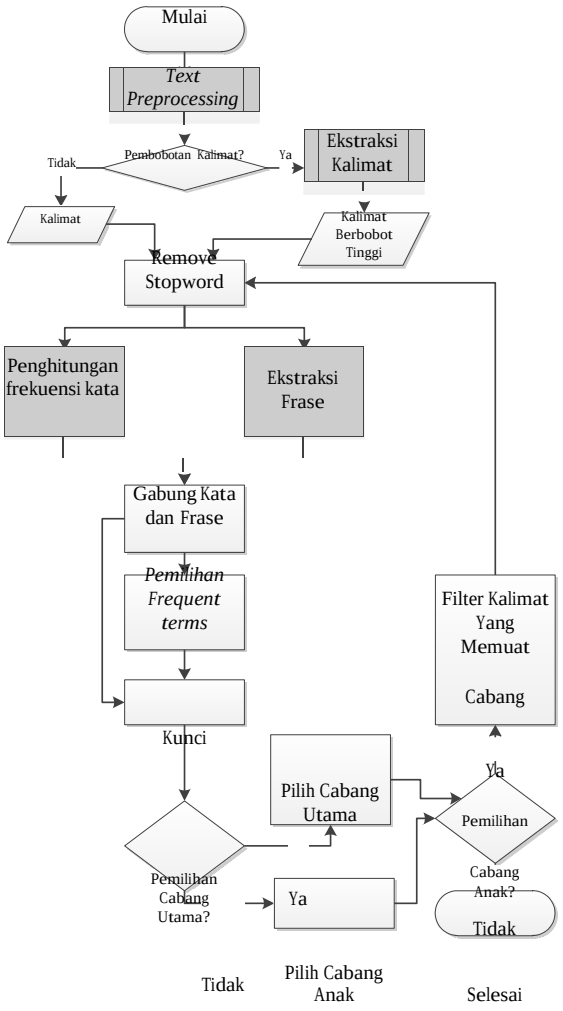
(6)

3. Perancangan Sistem

3.1 Gambaran Umum Sistem

Secara umum, Peta Pikiran Otomatis dibagi menjadi tiga modul utama, yaitu modul Ekstraksi Kalimat, modul Ekstraksi Kata Kunci, dan modul Pemilihan Cabang. Sistem tanpa pembobotan kalimat mendapatkan kalimat langsung dari hasil text preprocessing. Sedangkan, sistem dengan pembobotan kalimat mendapatkan kalimat dari modul ekstraksi kalimat. Modul Ekstraksi Kalimat mengekstrak kalimat sejumlah maksimal 50% dari total jumlah kalimat dari teks input dengan memanfaatkan pembobotan kalimat berdasarkan posisi, kemiripan kalimat dengan judul, dan kemunculan cue words.

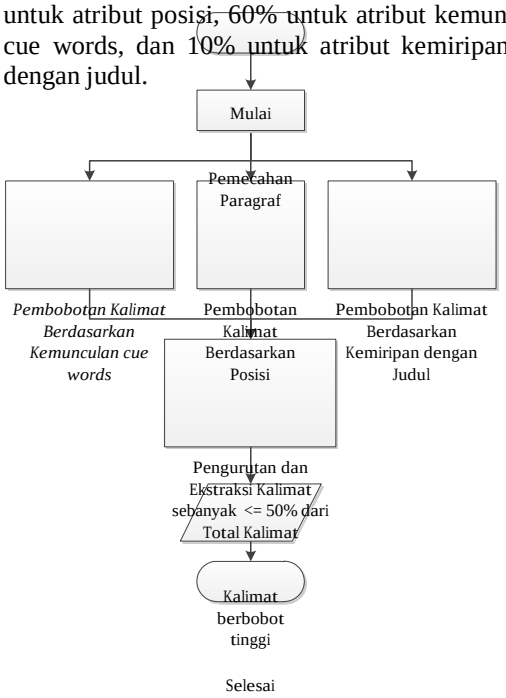
Di bawah ini adalah gambaran umum sistem.



Gambar 3.1 Diagram alur sistem.

3.2 Modul Ekstraksi Kalimat

Setiap kalimat memiliki tiga atribut nilai bobot. Atribut tersebut memiliki nilai maksimal 30% untuk atribut posisi, 60% untuk atribut kemunculan cue words, dan 10% untuk atribut kemiripan kata dengan judul.

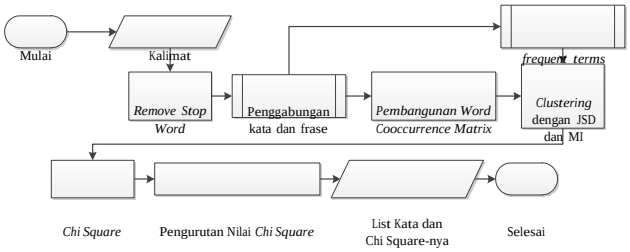


Gambar 3.2 Modul Ekstraksi Kalimat

3.3 Modul Ekstraksi Kata Kunci

Modul ini menghitung nilai *chi square* setiap kata berdasarkan bias *co-occurrence* setiap kata dengan *frequent terms* yang dipilih. Kalimat yang menjadi input terlebih dahulu diproses oleh *Pre-keyword extraction*. *Pre-keyword extraction* dibagi menjadi proses penghitungan frekuensi kata, ekstraksi frasa, penggabungan kata dan frasa, dan pemilihan frequent terms. Proses Penghitungan Frekuensi Kata hanya menghitung kata-kata yang lolos remove stopword.

Proses ekstraksi frasa mengekstrak frasa dengan mencari dan menghitung kata-kata yang berdampingan di setiap kalimat (*word sequence*). Frasa yang diekstrak hanya bermuatan dua sampai tiga kata dengan kemunculan minimal empat kali dalam teks. frasa. Frasa yang dicari terbatas pada frasa yang tidak diperluas. Sehingga, frasa yang memuat stopword list tidak akan terdeteksi.



Gambar 3.3 Modul Ekstraksi Kata Kunci

Selanjutnya, semua kata /frase yang diperoleh menjadi bahan pembangun word co-occurrence matrix. Proses pembangunan word co-occurrence matrix dilakukan dengan cara menghitung dua kata yang muncul bersamaan dengan mengabaikan urutan atau posisi kemunculan dua kata tersebut dalam kalimat.

proses *Clustering* dilakukan untuk mengelompokkan frequent term yang dihasilkan dari modul penghitungan TF di modul sebelumnya. Proses ini memanfaatkan Jensen-Shannon Divergence (JSD) dan Mutual Information (MI).

atau MI. Clustering dengan menggunakan JSD dan

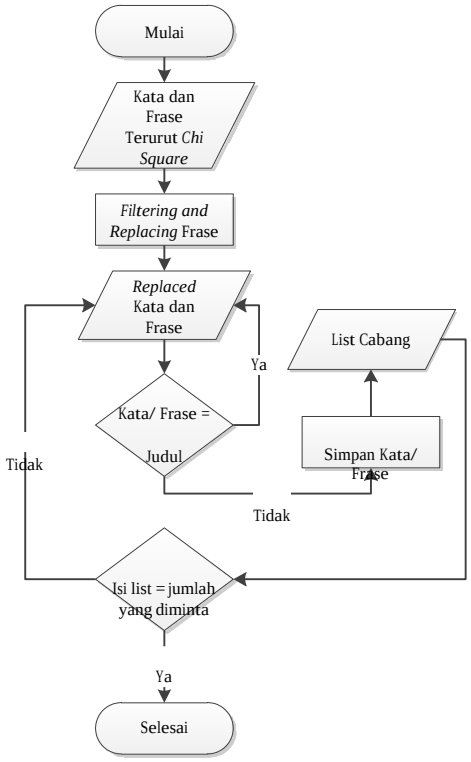
menjadi lebih baik (Matsuo, 2003).

Berikutnya, proses penghitungan nilai Chi Square setiap kata. Setelah setiap kata mempunyai nilai Chi Square, kata dan frasa diurut berdasarkan nilai Chi Square dari yang tertinggi sampai yang paling rendah.

3.4 Modul Pemilihan Cabang

Modul pemilihan cabang digunakan pada proses pemilihan cabang utama dan proses pemilihan cabang anak. Pada pemilihan atau ekstraksi cabang-cabang peta pikiran, kata-kata dan frasa (word sequence) dengan nilai Chi Square tertinggi dicek apakah ada frasa yang sama persis
dengan judul atau tidak. Bila ada, frasa tersebut tidak dipilih untuk mengisi cabang utama,

meskipun memiliki nilai Chi Square tinggi. Pada pemilihan cabang utama, bila kandidat cabang berupa frasa dan pembentuknya di awal keluaran hasil ekstraksi kata kunci (nilai chi square kata dan frasa tersebut berada di segmen teratas), maka frasa yang diekstrak menjadi cabang. Hal ini dilakukan karena bila suatu frasa dan kata-kata pembentuknya berada pada urutan teratas hasil ekstraksi kata kunci, frasa tersebutlah yang dibahas dokumen input sistem.



Gambar 3.4 Modul Pemilihan Cabang.

Proses filtering and replacing frasa tidak mengubah urutan keseluruhan, namun hanya menempatkan frasa pada posisi kata pembentuknya yang teratas dan menghilangkan kata-kata pembentuknya.

Segmentasi urutan kata kunci dilakukan untuk menentukan jumlah iterasi yang dibutuhkan dalam filtering and replacing frasa dari list kandidat kata kunci. List kandidat kata kunci dibagi menjadi lima segmen. Segmen 1 teratas, segmen 2 atas, segmen 3 medium, segmen 4 bawah, segmen 5 paling bawah. Sehingga, filtering and replacing frasa pada modul pemilihan cabang mencari frasa pada range iterasi sebanyak list kandidat kata kunci div 5.

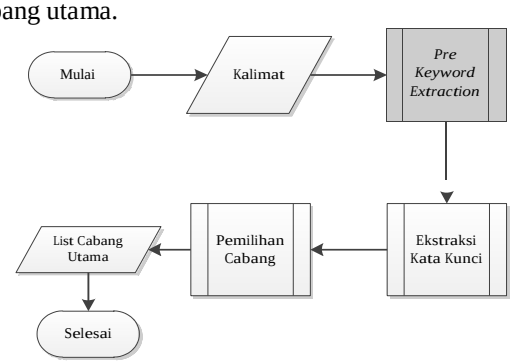
Proses filtering and replacing frasa pada proses pemilihan cabang sebetulnya adalah improvisasi karena kata pembentuk word sequence sama-sama memiliki nilai chi square yang besar sehingga muncul dengan urutan berdekatan dengan posisi komponennya. Hal tersebut akan membingungkan user dalam memahami maksud dari peta pikiran yang cabang-cabangnya di suatu level yang sama (terutama cabang utama) berisi word sequence (frasa) dan komponennya. Penggunaan segmentasi dalam menghasilkan iterasi bermaksud agar word sequence yang didapat tetap berkualitas karena masih dalam range segmen teratas. Urutan word sequence pada saat ekstraksi disesuaikan dengan urutan didapatnya word sequence pada list kandidat kata kunci. Sehingga, apabila ada word sequence di urutan 11 dan 13, maka word sequence urutan 11 yang pertama dicek untuk ditempatkan pada kata pembentuknya.

Contoh lain : kata “perintah” ada di posisi 5, kata “kalimat” di posisi 7, kata “berita” ada di posisi 8, word sequence “kalimat berita” ada di posisi 11, dan word sequence “kalimat perintah” ada di posisi 13. Maka, “kalimat perintah” akan me- replace kata “perintah” dan menempati posisi 5. Kemudian, kata “kalimat” dibuang karena sudah diwakili oleh “kalimat perintah” yang ada di posisi lebih awal. Kata “berita” naik posisi ke posisi 7. Kemudian, “kalimat berita” me-replace kata “berita” di posisi 7.

Dengan kata lain, urutan antar komponen frasa dan urutan antar word sequence sangat diperhatikan. Dan, urutan antar urutan komponen pembentk frasa yang lebih dulu diperhatikan, kemudian urutan frasa (word sequence) diperhatikan. Lampiran B menampilkan tabel-tabel berisi urutan kata kunci hasil sistem tanpa dan dengan filtering and replacing frasa.

3.4.1 Proses Pemilihan Cabang Utama

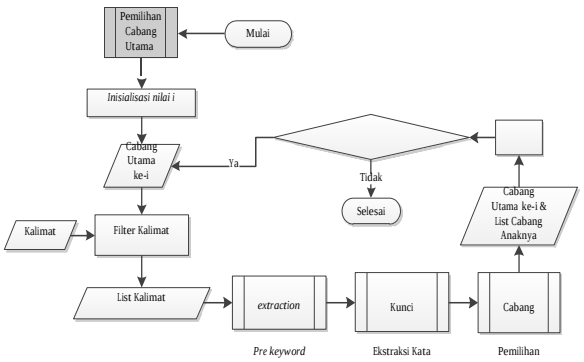
Modul pemilihan cabang utama mengekstraksi cabang utama peta pikiran. Input modul adalah list kandidat kata kunci yang sudah diurut berdasarkan nilai chi square. Di bawah ini flowchart pemilihan cabang utama.



Gambar 3.5 Modul Pemilihan Cabang Utama

3.4.2 Proses Pemilihan Cabang Anak

Selanjutnya, proses pemilihan cabang anak. Pada proses ini, cabang utama yang telah didapat pada proses sebelumnya menjadi input untuk me-list kalimat-kalimat (output modul Pembobotan Kalimat) yang memuat kata/frasa cabang utama tersebut. Kemudian, kalimat-kalimat tersebut menjadi input modul ekstraksi kata kunci untuk mengekstrak kata kunci untuk cabang anak. Lebih jelasnya perhatikan flow chart proses pemilihan cabang anak di bawah ini.



Gambar 3.6 Proses Pemilihan Cabang Anak.

Proses Filter Kalimat menyaring kalimat yang memuat cabang utama ke-i dan menghapus kata yang sama dengan cabang utama selain cabang ke-i pada kalimat tersebut. Kemudian, kalimat tersebut diproses di modul Ekstraksi Kata Kunci yang menghasilkan list kata kunci. List kata kunci tersebut diproses di modul pemilihan cabang untuk memilih cabang anak. Setelah didapat list cabang anak untuk cabang utama ke-i, cek iterasi dengan jumlah cabang utama. Bila iterasi yang diperoleh sudah melebihi jumlah cabang utama, maka proses selesai. Bila iterasi kurang dari sama dengan jumlah cabang utama, proses dilanjutkan ke iterasi berikutnya.

4. Analisis Hasil Pengujian

4.1 Pengujian Ekstraksi Kalimat (Pembobotan Kalimat)

Training modul pembobotan kalimat dilakukan dengan membandingkan antara esktraksi kalimat hasil sistem dan abstrak paper. Paper yang diuji adalah paper KNIF[17]. Abstrak dan ekstraksi kalimat hasil sistem di-remove stopword sebelum dibandingkan. Pembanding keduanya adalah Rouge-2-Gram yang sudah diimplementasikan pada bahasa Java[15]. Recall tertinggi hasil perhitungan Rouge menunjukkan kombinasi fitur pembobotan kalimat terbaik.

Nilai *recall* individual setiap fitur dengan CR 50% adalah 33% untuk kemunculan fitur kemunculan cue words, 38% untuk fitur kemiripan kalimat dengan judul, dan 36% untuk fitur posisi kalimat. Gabungan penggunaan semua fitur dengan kombinasi 30%,60%,10% secara berurutan untuk fitur posisi kalimat, fitur kemiripan kalimat dengan judul, dan fitur kemunculan *cue words* menghasilkan recall 41,15%.

4.2 Pengujian Ekstraksi Kata Kunci

4.2.1 Penentuan Jumlah *Frequent terms*

Penentuan jumlah *frequent terms* pada modul dilakukan karena langkah pertama dari algoritma yang digunakan untuk men-generate peta pikiran adalah mengekstraksi kata kunci yang dijadikan cabang utama. Pada penelitian ini diekstrak delapan kata kunci yang akan berperan sebagai cabang utama. Oleh karena itu, harus dicari jumlah frequent terms yang menghasilkan urutan kata kunci yang stabil. Kestabilan dilihat berdasarkan rata-rata panjang urutan kata kunci yang sama antara parameter 10%, parameter 20%, dan parameter 30%. Parameter dengan rata-rata panjang urutan kata kunci yang sama terbesar dianggap paling reliable.

Penentuan parameter ini sekaligus menguji apakah implementasi ekstraksi kata kunci sudah benar atau tidak. Penentuan jumlah frequent terms dilakukan dengan mengamati urutan kata kunci yang telah diurutkan oleh nilai chi square¬-nya tanpa terpengaruh oleh clustering. Urutan kata kunci yang diamati cukup kata kunci cabang utama saja.

Tabel 4-1 Hasil Penghitungan Urutan Kata Kunci Sistem

Paper	Parameter			Paper	Parameter		
	10%	20%	30%		10%	20%	30%
1	1	6	6	16	5	0	5
2	11	11	10	17	3	6	6
3	4	8	8	18	7	4	7
4	11	11	5	19	6	14	14
5	5	7	7	20	17	27	27
6	5	19	19	21	1	23	23
7	11	17	17	22	4	4	1
8	2	16	16	23	4	4	1
9	6	11	11	24	10	13	13
10	11	12	12	25	4	20	20
11	15	15	15	26	4	10	10
12	7	8	8	27	8	9	9
13	16	16	5	28	10	2	10
14	7	7	3	29	5	19	19
15	3	7	7	30	7	20	20
				Total	210	346	334

Dari tabel di atas, parameter 10% memiliki total panjang urutan kata kunci yang sama sebanyak 210 kata kunci. Rata-tata panjang urutan kata kunci yang sama untuk parameter ini adalah 210/30 = 7.0 kata kunci. Parameter 20% memiliki total panjang urutan kata kunci yang sama sebanyak 346 kata kunci. Rata-tata panjang urutan kata kunci yang sama untuk parameter ini adalah 346/30 = 11,53 kata kunci. Parameter 30% memiliki total panjang urutan kata kunci yang sama yang sama sebanyak 334 kata kunci. Rata-tata panjang urutan kata kunci yang sama untuk parameter ini adalah 334/30 = 11,13 kata kunci. Dari hasil tersebut, maka parameter 20% lah dipilih untuk digunakan oleh sistem.

4.2.2 *Balancing Clustered Frequent terms*

Dalam pengujian tahap ini, hasil yang pertama dilihat adalah rata-rata "Out of Range" yang bernilai paling rendah (pada penelitian ini adalah 0). Ini karena bila rata-rata "Out of Range" bernilai lebih dari 0, maka ada *cluster* yang berisi lebih dari lima *frequent terms* yang berarti *clustering* tidak seimbang. Selanjutnya, barulah mencari parameter yang memiliki rata-rata "In Range" paling tinggi di antara parameter yang menghasilkan rata-rata "Out of Range" bernilai 0. Parameter yang memenuhi kedua kriteria tersebut dianggap yang paling optimal dalam *balancing cluster*.

Hasil pengujian *clustering* yang menggunakan JSD, rata-rata "Out of Range" bernilai 0 dimiliki parameter bernilai 0,85 dan 0,85 juga menghasilkan "In Range" tertinggi. Sehingga, *threshold* JSD = 0.85 * Log(2).

Hasil pengujian *clustering* yang menggunakan MI, rata-rata "Out of Range" bernilai 0 dimiliki parameter bernilai 9 sampai 9,1. Dan, parameter 9,1 adalah batas atas parameter yang menghasilkan "In Range" tertinggi. Sehingga, *threshold* MI = 9.1 * Log(2).

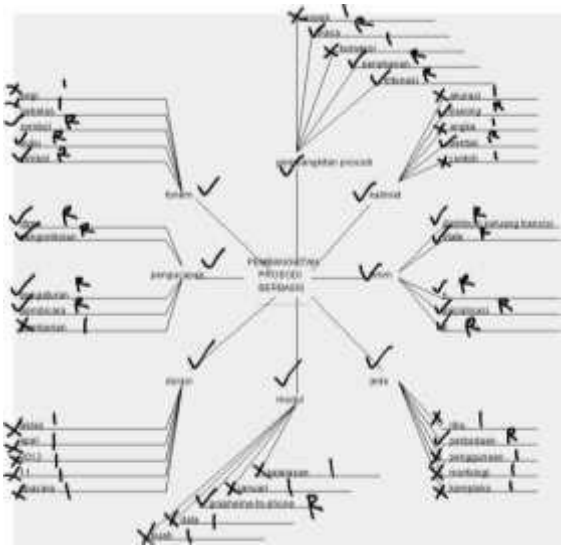
4.2.2 Hasil Pembandingan Kata Kunci

Pembandingan kata kunci hasil sistem dilakukan setelah penentuan jumlah *frequent terms* dan *balancing cluster*. Perbandingan dilakukan dengan membandingkan antara kata kunci hasil sistem dan kata kunci pada paper. Ini dilakukan untuk menguji modul ekstraksi kata kunci dalam menghasilkan kata kunci.

Dari hasil pembandingan kata kunci hasil sistem dengan kata kunci pada paper, sistem mampu mengekstrak kata kunci dengan rata-rata presisi sebesar 0.38 dan *recall* sebesar 0.47.

4.3 Pengujian Peta Pikiran (Mind Map)

Pengujian tahap ini melibatkan *author* paper (penulis) dalam penilaian hasil sistem. Pengujian dilakukan dengan men-*generate* peta pikiran dari teks paper. Kemudian, penulis paper tersebut memberikan penilaian terhadap peta pikiran yang dihasilkan sistem. Aspek yang dinilai adalah kepentingan kata kunci pada cabang yang dihasilkan, baik cabang utama, maupun cabang anak. Aspek selanjutnya adalah relevansi antara cabang-cabang anak dengan cabang utamanya. Peta pikiran hasil sistem yang diperiksa setiap penulis untuk setiap teks paper ada dua, yaitu peta pikiran dengan pembobotan kalimat dan peta pikiran tanpa pembobotan kalimat. Para penulis tidak tahu mana peta pikiran yang dengan pembobotan maupun yang tanpa pembobotan kalimat. Di bawah ini tabel hasil penilaian para author.



Gambar 4.1 Hasil Screen Shot Peta Pikiran yang telah diperiksa Author**.**

Tabel-tabel di bawah ini adalah statistic hasil perhitungan kata penting dan relevan yang telah dibandingkan dengan setiap jumlah cabang peta pikiran dari setiap teks paper input.

Tabel 4-2 Hasil Penilaian Sistem Dengan Pembobotan Kalimat

No	Pembobotan					
	Rata-rata Kata Penting di cabang		Rata-rata kata di Cabang Anak			
			Penting		Tidak Penting	
	Utama (%)	Anak (%)	r (%)	i (%)	r (%)	i (%)
1	100.00	47.50	47.50	0.00	0.00	52.50
2	50.00	24.24	15.15	9.09	0.00	75.76
3	100.00	52.50	40.00	12.50	7.50	40.00
4	62.50	37.50	37.50	0.00	0.00	62.50
5	62.50	32.50	32.50	0.00	5.00	62.50
6	50.00	35.00	35.00	0.00	0.00	65.00
Rata-rata	**70.83**	38.21	**34.61**	3.60	2.08	59.71

Tabel 4-3 Hasil Penilaian Sistem Tanpa Pembobotan Kalimat

No	Tanpa Pembobotan					
	Rata-rata Kata Penting di cabang		Rata-rata kata di Cabang Anak			
			Penting		Tidak Penting	
	Utama (%)	Anak (%)	r (%)	i (%)	r (%)	i (%)
1	100.00	37.50	37.50	0.00	0.00	62.50
2	87.50	42.50	37.50	5.00	2.50	55.00
3	75.00	48.72	38.46	10.26	12.82	38.46
4	62.50	42.50	42.50	0.00	0.00	57.50

5	25.00	29.17	29.17	0.00	29.17	41.67
6	100.00	40.00	40.00	0.00	0.00	60.00
Rata-rata	**75.00**	40.06	**37.52**	2.54	7.41	52.52

Dari dua tabel di atas rata-rata kata penting di cabang utama sistem dengan pembobotan kalimat ( ) adalah 70,83%. Dan, rata-rata kata penting di cabang utama sistem tanpa pembobotan kalimat ( ) adalah 75%. Dengan hasil ini, kedua kondisi sistem sudah menghasilkan cabang utama yang cukup memuaskan. Sistem tanpa pembobotan kalimat dinilai lebih baik karena memiliki rata-rata yang lebih tinggi dari rata-rata hasil sistem dengan pembobotan.

Cabang-cabang utama hasil kedua sistem memiliki perbedaan pada kata kuncinya dan urutan kemunculan kata kuncinya. Hal ini terjadi karena pembobotan kalimat mengestrak 50% kalimat yang digunakan untuk diproses menjadi peta pikiran. Dari fakta tersebut ada dua hal yang mempengaruhi hasil kata kunci pada peta pikiran. Pertama, jumlah kalimat hilang sebesar 50%. Kedua, karena jumlah kalimat berkurang, *terms* yang diproses pun ikut berkurang.

Dengan hilangnya kalimat sebesar 50%, maka sesuai formula perhitungan *chi square* dalam ekstraksi kata kunci, nilai *chi square* kata akan terpengaruh. Ini karena, frekuensi kata (*frequent terms* dan *terms*) akan berkurang yang berpengaruh pada *expected frequency* dan *observed frequency*.

Jika suatu kata yang sedang dicari nilai *chi square*-nya (*current term*) kehilangan kalimat, maka yang terjadi adalah kata tersebut memiliki peluang menjadi tidak *co-occur* dengan suatu *frequent terms*. Dan, nilai nw (total kata pada dimana *current term* muncul) pun akan sangat terkikis. Sehingga, apabila dua hal tersebut terjadi, maka nilai *chi square* pasti lebih kecil dari kata yang *co-occur* dengan semua *frequent terms*. Dan, ini berpengaruh terhadap urutan kemunculan cabang.

Berkurangnya kalimat mengakibatkan berkurangnya *terms*. Seperti kita tahu, pembobotan kalimat tidak mempedulikan kata-kata apa saja yang akan hilang bersama dengan kalimat yang tidak terpilih. Jadi, meskipun diasumsikan kalimat-kalimat hasil pembobotan kalimat adalah kalimat-kalimat penting, ada kemungkinan kalimat yang tidak masuk di 50% adalah juga kalimat penting yang memuat *frequent terms* dan *terms* lain yang tidak ada pada list kalimat. Sedangkan, pada penelitian kali ini hanya *compression rate* 50% yang diuji untuk sistem dengan pembobotan kalimat. Sehingga, *compression rate* yang tepat untuk sistem dengan pembobotan kalimat disarankan untuk dicari pada penelitian berikutnya.

Pada rata-rata kata penting cabang anak yang relevan dengan cabang utamanya ( ), sistem tanpa pembobotan kalimat memiliki rata-rata 37,52%. Lebih besar dari hasil sistem dengan pembobotan kalimat yang hanya 34,61%. Dari hasil tersebut, sistem tanpa pembobotan sedikit lebih baik dari sistem tanpa pembobotan dalam menghasilkan cabang-cabang anak.

Dua hal yang mempengaruhi hasil perhitungan *chi square* cabang utama yang telah dijelaskan sebelumnya, juga mempengaruhi hasil perhitungan *chi square* di cabang anak. Ini karena proses pengekstraksian cabang anak adalah sama dengan proses ekstraksi cabang utama. Perbedaan adalah pada kalimat yang menjadi input proses.

Hal lain yang juga mempengaruhi hasil cabang anak adalah urutan kemunculan cabang utama yang list kalimat terfilternya beririsan dengan list kalimat cabang utama yang lain. Sehingga, cabang utama yang urutan kemunculannya lebih awal akan mengklaim cabang anak.

Dari hasil pengujian juga ada cabang anak hasil sistem tidak berjumlah 40. Padahal sistem di-*run* dalam kondisi untuk menghasilkan lima cabang anak untuk delapan cabang utama. Hal ini terjadi karena karakteristik *word statistical information* dalam mengekstraksi kata kunci dan algoritma yang diimplementasikan sistem dalam men-*generate* cabang anak.

Matsuo (2003) menyebutkan bahwa algoritma *keyword extraction* buatannya mampu mengekstrak suatu kata kunci meskipun frekuensi kemunculan kata tersebut kecil. Hal ini kemudian berkaitan dengan algoritma dalam men-*generate* cabang anak yang mencari dan memproses kalimat yang memuat *current* cabang utama. Dan, hal tersebut juga berkaitan dengan langkah cabang-cabang utama yang lebih dahulu muncul mengklaim kata kunci dari kalimat tersebut.

Frekuensi kemunculan kata cabang utama kecil berarti kalimat yang memuat cabang utama tersebut juga hanya sedikit. Sehingga, apabila kalimat-kalimat yang memuat suatu cabang utama ke-i semuanya beririsan dengan kalimat-kalimat cabang-cabang utama lain yang lebih dahulu muncul, maka besar sekali kemungkinan kata-kata dari kalimat yang memuat cabang utama ke-i sudah diklaim oleh cabang-cabang utama lain yang muncul lebih

dahulu. Sehingga, cabang utama ke-i memiliki cabang anak kurang dari lima cabang atau bahkan tidak memiliki cabang anak sama sekali.

5. Kesimpulan

Pembobotan kalimat dengan kombinasi nilai bobot (30%, 60%, 10%) secara berurut untuk fitur posisi kalimat, fitur kemiripan kalimat dengan judul, dan fitur kemunculan *cue words* menghasilkan *recall* sebesar 41,15% pada *compression rate* 50%. Ketiga fitur tersebut dapat digunakan dalam ekstraksi kalimat (pembobotan kalimat). Dari ketiga fitur pembobotan kalimat, fitur kemiripan kalimat dengan judul adalah fitur yang paling berpengaruh dalam pembobotan kalimat. Dan fitur ini juga pemilik *recall* tertinggi dalam pembobotan kalimat secara *individual*.

Word co-occurrence Statistical Information dapat digunakan untuk mengkalkulasi ranking kata kunci berdasarkan nilai bias (co-occur atau tidaknya dengan *frequent terms*) setiap *term*. Dengan kata lain, *Word co-occurrence statistical information* dapat digunakan sebagai dasar untuk ekstraksi kata kunci. Jumlah *frequent terms*, *threshold clustering* JSD dan *threshold clustering* MI pada modul ekstraksi kata kunci hasil *training* secara berurutan adalah 20% * total *running terms*, 0.85 * Log(2), dan 9.1 * Log(2). Hasil pengujian ekstraksi kata kunci memiliki presisi sebesar 0.38 dan *recall* sebesar 0.47.

Sistem mampu menghasilkan peta pikiran dari dokumen tunggal. Namun, keterkaitan antara cabang utama dengan cabang-cabang anaknya masih belum baik. Jumlah *frequent terms* berpengaruh pada nilai *chi square* pada proses ekstraksi kata kunci. Semakin banyak jumlah *frequent terms* yang *co-occur* dengan suatu kata, semakin besar nilai *chi square* kata tersebut.

Pembobotan kalimat secara tidak langsung berpengaruh pada peta pikiran yang dihasilkan. Ini karena pembobotan kalimat hanya menggunakan maksimal 50% dari total kalimat yang berpengaruh terhadap atribut *expected* dan *observed frequency* dalam mengukur bias *co-occurrence* setiap kata.

Untuk sistem yang dengan pembobotan kalimat, peta pikiran yang dihasilkan memiliki rata-rata jumlah kata kunci yang penting di cabang utama adalah 70,83%. Rata-rata jumlah cabang anak yang penting dan relevan dengan cabang utamanya adalah 34,61%. Untuk sistem tanpa pembobotan kalimat, peta pikiran yang dihasilkan memilki rata-rata jumlah kata kunci yang penting di cabang utama adalah 75%. Rata-rata jumlah cabang anak yang penting dan relevan dengan cabang utamanya adalah 37,52%. Dengan hasil ini, sistem tanpa pembobotan kalimat masih lebih baik dari sistem dengan pembobotan kalimat dalam men-*generate* peta pikiran dari teks.

6. Saran

Saran yang muncul dari penelitian ini adalah :

1. Sistem pada penelitian lebih lanjut diharapkan dapat men-*generate* lebih dari dua level cabang peta pikiran.
2. Mencari metode untuk meningkatkan keterkaitan antara cabang utama dan cabang anak.
3. Mencari metode lain untuk men-*generate word sequence* (frasa) agar frasa seperti “kata kunci” dan “lalu lintas” bisa didapatkan.
4. Perlu dicari *compression rate* yang tepat untuk sistem dengan pembobotan kalimat yang lebih baik dari CR 50% dalam menghasilkan peta pikiran.

Daftar Pustaka

[1] Buzan, Tony. 2005. The Ultimate Book of Mind Maps, Alih bahasa : Susi Purwoko, Jakarta : Penerbit Gramedia.

[2] Hovy, Eduard. 2005. The Oxford Handbook of Computational Linguistics. Chapter 32 Text Summarization.

[3] Matsuo, Yutaka. 2003. *Keyword Extraction from a Single Document using Word Co-occurrence Statistical Information*. Tokyo.

[4] Aristoteles. 2011. Pembobotan Fitur Pada Peringkasan Teks Bahasa Indonesia Menggunakan Algoritme Genetika. Bogor.

[5] Durachman, Memen. Pertmuan V Paragraf: Pengertian, Jenis, Syarat Pembentukan, dan Metode Pengembangannya. UPI. http://file.upi.edu/Direktori/FPBS/JUR._PEND._BHS._DAN_SASTRA_INDONESIA/196306081988031-MEMEN_DURACHMAN/Paragrafx.pdf diakses pada jam 19.09 WIB tanggal 22 April 2015.

[6] Pratama, Indra. 2013. Analisis dan Implementasi Perbandingan Stemming Menggunakan Algoritma Ahmad Yusoff Sembok dengan Jelita Asian pada Pencarian Ayat Al-Quran yang Terkait Hadits. Bandung.

[7] Wikipedia. Pearson’s Chi Squared Test. http://en.wikipedia.org/wiki/Pearson's_chi-squared_test diakses pada jam 15.50 tanggal 15 April 2015.

[8] Adikara, Putra. Kamus Kata Dasar dan Stopword List Bahasa Indonesia. http://hikaruyuuki.lecture.ub.ac.id/kamus-kata-dasar-dan-stopword-list-bahasa-

indonesia/ diakses pada jam 0.19 WIB tanggal 7 Mei 2015.

[9] Adikara, Putra. *Stopword List* Bahasa Indonesia. http://static.hikaruyuuki.com/wp-content/uploads/stopword_list_tala.txt diakses pada jam 0.19 WIB tanggal 7 Mei 2015.

[10] Hulth, Anette. 2003. *Improved Automatic Keyword Extraction Given More Linguistic Knowledge*. Swedia.

[11] Giarlo, Michael. 2005. *A Comparative Analysis of Keyword Extraction Techniques*. Rutgers, The State University of New Jersey.

[12] Mandala, Yohanda. 2013. Rancang Bangun Aplikasi Dream Share – Pembuat Peta Pikiran. http://digilib.its.ac.id/public/ITS-paper-19954-5108100196-paper.pdf Diakses pada jam 3.00 WIB tanggal 21 April 2015

[13] Pereira, Fernando. 1999. *Similarity-Based Models of Word Co-occurrence Probabilities*. Boston.

[14] Lin, Chin-Yew. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. Barcelona, Spain.

[15] Krapivin, Eugene. 2014. JRouge-Java Rouge Implementation. https://bitbucket.org/nocgod/jrouge/wiki/Home diakses pada 16 Mei 2015.

[16] Zulkifli. 2015. Pembobotan Fitur Ekstraksi Pada Peringkasan Teks Bahasa Indonesia Menggunakan Algoritma Genetika. Bandung.

[17] KNIF. 2013. Prosiding Konferensi Nasional Informatika. http://knif.itb.ac.id/knif/prosiding.pdf diakses pada 9 Mei 2015.

[18] Cunningham. 2005. Mindmapping: Its Effects on Student Achievement in High School Biology. The University of Texas at Austin.

[19] Holland, Brian. 2004. An Investigation into The Concept of Mind Mapping and The Use of Mind Performance.