Visualisasi dan Penelusuran Keterhubungan Subjek Karya Ilmiah sebagai Upaya Layanan Perpustakaan dalam Mendukung Inovasi dan Invensi Pengetahuan

Firman Ardiansyah
Magister Teknologi Informasi untuk Perpustakaan, Institut Pertanian Bogor, Bogor
f.ardiansyah@apps.ipb.ac.id

Badollahi Mustafa
Perpustakaan Institut Pertanian Bogor, Bogor
mus@ipb.ac.id

Chusnul Arifin Rusuandar
Departemen Ilmu Komputer, Institut Pertanian Bogor, Bogor
chusnul.arifin21@gmail.com

Abstrak:

Pencarian dan penelusuran karya ilmiah menggunakan subjek pada dokumen merupakan cara yang efektif dalam mendapatkan informasi penelitian. Upaya pencarian dan penelusuran akan menjadi lebih interaktif jika ditambahkan visualisasi informasi dan keterhubungan antarsubjek. Visualisasi informasi mampu menyuguhkan interaksi yang efisien dan efektif untuk beragam pekerjaan kognitif pemustaka seperti menganalisis, meringkas dan menarik kesimpulan atas informasi yang didapat. Sementara keterhubungan antarsubjek mampu membawa pemustaka memahami keterkaitan topik yang mengantarkan pada munculnya ide-ide baru. Pada kajian ini telah terbangun sebuah sistem penelusuran dokumen berdasarkan subjek dan memvisualisasikan keterhubungan antaristilah subjeknya. Kajian ini bertujuan menjabarkan proses pembangunan sistem dimulai dari pengelolaan data, pengindeksan, penghitungan peluang antaristilah, perancangan visualisasi, dan perancangan antarmuka pencarian dan penelusuran berbasis visual. Selain itu, kajian ini akan merinci infrastruktur dan teknologi yang digunakan hingga tata cara penggunaan sistem bagi perpustakaan yang ingin mengembangkan layanannya. Dengan kajian ini diharapkan perpustakaan dapat meningkatkan layanan kepada pemustaka dalam mengembangkan pengetahuan sehingga mendukung terciptanya iklim inovasi dan invensi di Indonesia sesuai dengan Tujuan Pembanguan Berkelanjutan.

Kata kunci: sistem penelusuran, keterhubungan subyek, visualisasi informasi.

PENDAHULUAN

Pencarian (*searching*) dan penelusuran (*browsing*) dokumen merupakan kegiatan yang paling sering terjadi di perpustakaan. Dalam proses mendapatkan informasi, pencarian dan penelusuran dokumen pada suatu koleksi merupakan 2 hal yang berbeda. Pencarian adalah proses memasukkan *query* ke dalam suatu *search engine* yang menghasilkan dokumen-dokumen yang sudah diurutkan berdasarkan tingkat kecocokan dengan *query*. Penelusuran adalah proses melihat suatu halaman secara berurutan dengan menggunakan *hyperlink* (Olston & Chi 2003). Pencarian dan penelusuran dokumen merupakan kegiatan yang relatif mudah dilakukan untuk mendapatkan suatu informasi. Akan tetapi, keduanya memiliki keterbatasan.

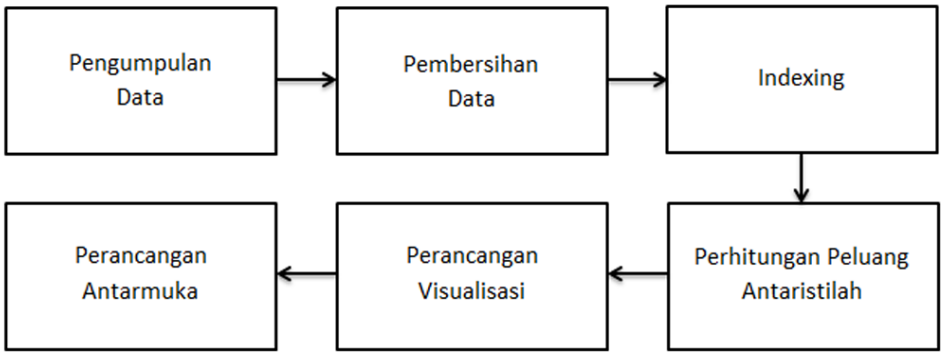
Menurut Manber *et al.* (2007), pencarian terkadang sulit dilakukan oleh pengguna yang belum menentukan *query* yang akan dimasukkannya, sedangkan penelusuran dapat membuat konten dalam koleksi menjadi lebih hidup sehingga pengguna yang belum menentukan *query* akan menjadi tertuntun namun membutuhkan waktu yang lama dalam mendapatkan informasi.

Proses pencarian berdasarkan subjek atau topik pada *metadata* dokumen merupakan cara yang efektif jika pengguna ingin mencari dokumen yang sesuai dengan topik yang diinginkan. Hasil pencarian dan penelusuran tersebut akan lebih interaktif jika ditambahkan visualisasi informasi. Menurut Chase *et al.* (1998), visualisasi dapat mendukung interaksi yang efisien dan efektif untuk beragam pekerjaan kognitif pengguna seperti menganalisis, meringkas dan menarik kesimpulan atas informasi yang didapat. Akan tetapi, kebutuhan pengguna dalam mencari infomasi semakin meluas dan rumit sehingga visualisasi sederhana tidak cukup untuk mengatasi masalah tersebut. Terkadang informasi mengenai keterhubungan antardokumen tidak terilustrasikan dengan baik sehingga pengguna sulit untuk menelusuri informasi dari sebuah dokumen ke dokumen lainnya (Kboubi *et al.* 2010). Oleh karena itu, perlu dibuat sistem yang memudahkan pengguna untuk mendapatkan informasi yang diinginkan dengan menelusuri ketehubungan antarsubjek.

Pada kajian ini, sistem yang dibangun adalah untuk penelusuran subjek bukan untuk pencarian sehingga sistem tidak perlu melakukan pemeringkatan relevansi hasil kueri. Selain itu, perhitungan peluang bersyarat dan pembuatan daftar istilah untuk kueri dilakukan pada saat proses pembuatan koleksi data (*indexing*). Jadi, pada saat pengguna memasukkan kueri untuk titik awal penelusuran, sistem hanya menampilkan hasil visualisasi berdasarkan nilai peluang yang sudah dihitung sebelumnya. Kajian ini berfokus pada proses membangun sistem penelusuran dokumen berdasarkan subjek yang dipisahkan berdasarkan frase dan menampilkan visualisasi keterhubungan istilahnya. Keterhubungan antaristilah dilihat dari kemunculan bersama pada setiap dokumen dan dihitung peluang kemunculan bersama antaristilah tersebut.

METODE

Secara ringkas, sistem ini dibangun berdasarkan metode yang dapat dilihat pada Gambar 1. Data yang digunakan sebagai contoh adalah *metadata* tesis IPB tahun 1963 sampai tahun 2013. Data diambil dari Perpustakaan IPB dalam bentuk data CDS/ISIS (*Computerised Documentation Service/Integrated Set Information Systems*). Format *metadata* di dalam data CDS/ISIS tersebut mengikuti aturan format INDOMARC (Indonesia-*Machine Readable Cataloging*). Data dalam format CDS/ISIS tersebut nantinya akan diekspor ke dalam format XLS (*Microsoft Excel Spreadsheet*) sebelum dilakukan proses pembersihan data.



Gambar 1 Metode pembangunan sistem visualisasi dan penelusuran keterhubungan antarsubjek

Pembersihan data dilakukan untuk membenahi data hasil ekspor dari bentuk data CDS/ISIS menjadi XLS. Data hasil ekspor akan memiliki *field* yang tidak terisi. Data yang sepeti ini disebut *bad-formed*. Data yang akan digunakan pada sistem ini harus berupa data XLS yang *well-formed*, yaitu data yang memiliki *field* yang lengkap atau terisi semua. Selain bagian *field* yang lengkap, data yang *well-formed* harus memiliki jumlah istilah yang cukup pada *field* subjek di dalam data XLS tersebut. Dalam kajian ini, ditetapkan jumlah subjek minimum dalam satu dokumen ialah dua subjek. Langkah selanjutnya adalan melakukan proses *indexing*.

Proses *indexing* dilakukan untuk membuat daftar istilah yang diambil dari *field* subjek pada data XLS. Untuk mengambil istilah-istilah tersebut, diperlukan proses *parsing* pada data XLS menggunakan *script* PHP ke dalam *database* MySQL. Setelah proses *parsing* data dilakukan, didapat kumpulan istilah subjek *metadata* tesis. Kumpulan istilah subjek tersebut akan digunakan untuk membuat *inverted index*. *Inverted index* adalah *index* yang selalu memetakan dari suatu istilah (*terms*) ke dalam bagian dokumen di mana istilah tersebut berada (Manning *et al.* 2008). *Index* tersebut dapat digunakan untuk menunjukkan dokumen yang direpresentasikan oleh kumpulan istilah tersebut.

Setelah dilakukan proses *indexing*, dilakukan proses penghitungan peluang antaristilah. Proses dimulai dengan menghitung peluang masing-masing istilah terhadap jumlah dokumen secara keseluruhan. Proses perhitungannya sesuai dengan persamaan (1).

$$P(f_a) = \frac{\sum F(A)}{F(S)} \quad (1)$$

dengan $\sum F(A)$ adalah jumlah istilah a dalam kumpulan istilah keseluruhan dan $F(S)$ adalah jumlah dokumen secara keseluruhan. Setelah itu dilakukan perhitungan banyaknya istilah satu dengan istilah yang lainnya yang berada dalam satu subjek *metadata* tesis.

Setelah itu dilakukan penghitungan peluang antaristilah menggunakan persamaan penghitungan peluang bersyarat sebagai berikut:

$$P(f_a|f_b) = \frac{P(f_a \cap f_b)}{P(f_b)} \quad (2)$$

dengan

$$P(f_a \cap f_b) = \frac{\sum (f_a, f_b)}{F(S)} \quad (3)$$

Peluang antaristilah dihitung untuk menunjukkan seberapa dekat satu istilah dengan istilah lainnya. Misalkan istilah a dan istilah b memiliki peluang yang besar, maka dapat dikatakan bahwa istilah a dekat dengan istilah b atau banyak tesis IPB yang memiliki subjek dengan pasangan istilah tersebut. Setelah perhitungan peluang selesai, dilakukan perancangan visualisasi.

Perancangan visualisasi dimulai dengan memilih bentuk dasar dari visualisasinya, contohnya *graph*, *tree*, atau *vector*. Pada kajian ini, bentuk dasar yang dipilih adalah *graph* dengan bentuk seperti sinar matahari. *Graph* tersebut mempunyai dua *node*, yaitu *parent* dan *child*. *Node parent* merupakan *node* yang merepresentasikan istilah yang akan ditelusuri keterhubungannya sedangkan *node child* merupakan *node* yang merepresentasikan istilah yang mempunyai keterhubungan dengan istilah yang ditelusuri. Visualisasi dilakukan dengan

mempertimbangkan besaran nilai peluang dari istilah yang akan ditelusuri. Perbedaan besaran nilai peluang dapat digambarkan dengan dengan membedakan warna atau ukuran *node child*. Setelah dilakukan perancangan visualisasi dilakukan proses penambahan fitur interaksi dan perancangan antarmuka.

Penambahan fitur interaksi dalam antarmuka dapat memudahkan pengguna dalam menggunakan aplikasi. Contoh interaksi yang akan terjadi selama penggunaan aplikasi adalah proses pemilihan istilah yang akan ditelusuri ketehubungannya (*node parent*) dan proses penelusuran keterhubungan antaristilah melalui hasil visualisasi. Fitur interaksi yang dibuat berupa *search box* untuk pencarian istilah pada daftar istilah atau penambahan animasi pada hasil visualisasi.

HASIL DAN PEMBAHASAN

Pengumpulan Data

Data yang digunakan sebagai contoh dalam kajian ini adalah *metadata* tesis IPB tahun 1963 sampai dengan 2013 yang diambil dari perpustakaan pusat IPB. Data tersebut berjumlah 12.109 dalam bentuk data CDS/ISIS yang dapat dilihat menggunakan program Winisis. Format penulisan data tersebut sudah mengikuti format INDOMARC. Gambar 2 merupakan contoh format INDOMARC dari data tersebut. Format INDOMARC dalam data CDS/ISIS tersebut memiliki beberapa *tag* yang menunjukkan kode atribut dari suatu data. Sebagai contoh *tag* 245 merupakan judul dari tesis dan *tag* 100 merupakan penulis utama tesis. Dalam penelitian ini, *tag-tag* yang akan digunakan adalah *tag* 245 yang merupakan judul tesis, *tag* 102 yang merupakan penulis utama tesis, *tag* 260 yang merupakan tahun terbit tesis, *tag* 300 yang merupakan atribut tambahan seperti jumlah halaman dari tesis, *tag* 710 yang merupakan departemen asal penulis utama, serta *tag* 650 yang merupakan subjek dari tesis dan yang akan menjadi fokus kajian ini.

```
260:    ^c2006
103:    ^aCharly Danny Heatubun
 35:    ^aJBUNIPB
 80:    ^a582.545
 99:    ^lT^a582.545^bHEA m
100:    ^aHeatubun, Charly Danny
102:    ^aHeatubun^bCharly Danny
245:    ^aA Monograph palm genus cyrtostachys blume (Arecaceae)^cCharly Danny Heatubun
300:    ^axiii, 66 lembar^bil.^c30 cm.
502:    ^aInstitut Pertanian Bogor, 2006
650:    ^aPalmae^xPalms
695:    ^aAnalisis filogenetika
695:    ^aMonofiletik
695:    ^aPhytogeography
695:    ^aPollen morphology
695:    ^aPhylogenetic analysis
695:    ^aCyrtostachys bakery heatubun
695:    ^aCyrtostachys barbata heatubun
710:    ^aInstitut Pertanian Bogor^bPascasarjana^cBiology
```

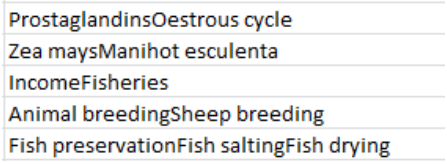
Gambar 2 Format INDOMARC data CDS/ISIS

Pembersihan Data

Kumpulan data CDS/ISIS yang sudah dikonversi ke dalam format XLS masih mengandung *record* data yang tidak bersih atau kurang sesuai dengan yang diinginkan. Oleh karena itu, perlu dilakukan proses pembersihan data, yaitu menghilangkan *record* data yang hanya memiliki 1 subjek, memperbaiki *record* data yang mengalami kesalahan *parsing* data saat proses konversi dari CDS/ISIS, dan menghilangkan tanda baca. Data awal yang dikonversikan dari CDS/ISIS berjumlah 12.109 *record* data. Setelah menghilangkan *record*

data yang hanya memiliki 1 subjek, data keseluruhan menjadi 9.304 *record* data. Selanjutnya, dilakukan perbaikan terhadap 1.499 *record* data yang mengalami kesalahan *parsing* data saat proses konversi. Gambar 3 menunjukkan contoh *record* data yang mengalami kesalahan *parsing* data.

Dari Gambar 3, dapat dikatakan bahwa *record* data yang mengalami kesalahan *parsing* data adalah *record* yang memiliki *camel case* (karakter *lowercase* dan *uppercase* yang berdekatan) pada datanya. *Camel case* disini menunjukkan bahwa jumlah subjek dari *record* data tersebut lebih dari 1 subjek, misalnya pada *record ProstaglandinsOestrous cycle* yang memiliki 2 subjek, yaitu *Prostaglandins* dan *Oestrous cycle*. Dalam kajian ini, *record* yang digunakan adalah *record* yang mengandung 2 subjek. Oleh karena itu, *record* tersebut perlu diperbaiki dengan cara mengambil subjek pertama dan subjek lainnya dihilangkan. Setelah *record* data tersebut diperbaiki, dilakukan penghilangan terhadap tanda baca pada *record* data. Setelah itu, data sudah dapat dianggap bersih dan siap untuk diolah. Jumlah data bersih secara



Gambar 3 Contoh *record* data yang mengalami kesalahan *parsing* data

keseluruhan berjumlah 9.304 *record* data.

Indexing

Proses *indexing* dimulai dengan mengimpor data XLS ke dalam *database* MySQL. Setelah itu, dibuat daftar istilah berdasarkan subjek dan dihitung frekuensinya. Perhitungan frekuensi dilakukan untuk mengetahui jumlah setiap pasang istilah yang ada. Selain itu, perhitungan dilakukan untuk memeriksa istilah yang mengandung anomali yang mungkin terjadi akibat kesalahan pengetikan atau perbedaan persepsi pustakawan yang memberikan subjek tesis. Tabel 1 dapat menunjukkan anomali tersebut.

Dapat dilihat dari Tabel 1, terdapat 2 pasangan istilah yang hanya berjumlah 1, sedangkan pasangan istilah lainnya berjumlah 4 sehingga dapat dikatakan bahwa 2 pasangan istilah tersebut mengandung anomali. Hal ini dikarenakan jumlah tesis kedua istilah tersebut hanya berjumlah 1 yang sangat memungkinkan adanya suatu kesalahan dalam pengetikan atau perbedaan persepsi pustakawan dalam memberi subjek tesis. Di lain sisi, tesis yang mengandung subjek *agricultural economics* dan *agribusiness* dianggap tidak ada anomali karena berjumlah 4 yang artinya sulit dikatakan mengandung kesalahan. Akan tetapi, sesungguhnya tesis-tesis tersebut bersubjek sama. Karena adanya anomali, sistem menganggap istilah subjek tersebut menjadi berbeda. Oleh karena itu, perlu dilakukan tindakan pembenahan istilah subjek yang mengandung anomali.

Tabel 1 Contoh istilah yang mengandung anomali

Subjek1	Subjek2	Jumlah
agricultural economic	agribusiness	1
agricultural economics	agribusiness	4
agricultural economy	agribusiness	1

Tindakan pembenahan istilah tersebut dapat dilakukan dengan dua pilihan, yaitu dengan memperbaiki istilah yang mengandung anomali menjadi bentuk istilah subjek yang sesuai atau membuat ketiga istilah tersebut menjadi seragam. Idealnya, tindakan yang paling tepat adalah

memperbaiki istilah yang mengandung anomali menjadi bentuk istilah subjek yang sesuai, namun diperlukan suatu pedoman atau kamus istilah untuk membandingkan bentuk yang sesuai dengan bentuk yang tidak sesuai tadi. Karena keterbatasan ini, maka dipilih tindakan yang kedua, yaitu membuat ketiga istilah tersebut menjadi seragam. Cara sederhana yang dapat dilakukan untuk menyeragamkan ketiga istilah tersebut adalah dengan memenggal imbuhan pada istilah tersebut. Salah satu algoritme yang dapat menghilangkan imbuhan adalah algoritme Porter Stemmer. Oleh karena itu, maka dilakukan proses *stemming* pada daftar istilah yang sudah dibuat. Tabel 2 merupakan hasil *stemming* dari istilah-istilah pada Tabel 1.

Tabel 2 Contoh istilah yang sudah dikenai *stemming*

No	Subjek1	Subjek2
1	agricultur econom	agribusi
2	agricultur econom	agribusi
3	agricultur econom	agribusi
4	agricultur econom	agribusi
5	agricultur economi	agribusi

Pada Tabel 2 dapat terlihat kekurangan dari algoritme Porter Stemmer, yaitu pada istilah nomor 5. Istilah pada Subjek1 yang mulanya *agricultural economy* berubah menjadi *agricultural economi* bukan *agricultural econom* sehingga istilah nomor 5 menjadi berbeda dengan yang istilah lainnya. Karena kekurangan ini, maka perlu dilakukan *thresholding* jumlah istilah yang bertujuan untuk mengurangi istilah-istilah yang kurang sesuai seperti istilah *agricultural economi*. Dalam penelitian ini, nilai *threshold* perlu diperhatikan karena akan mempengaruhi hasil dari daftar istilah dan waktu komputasi dari proses *indexing*.

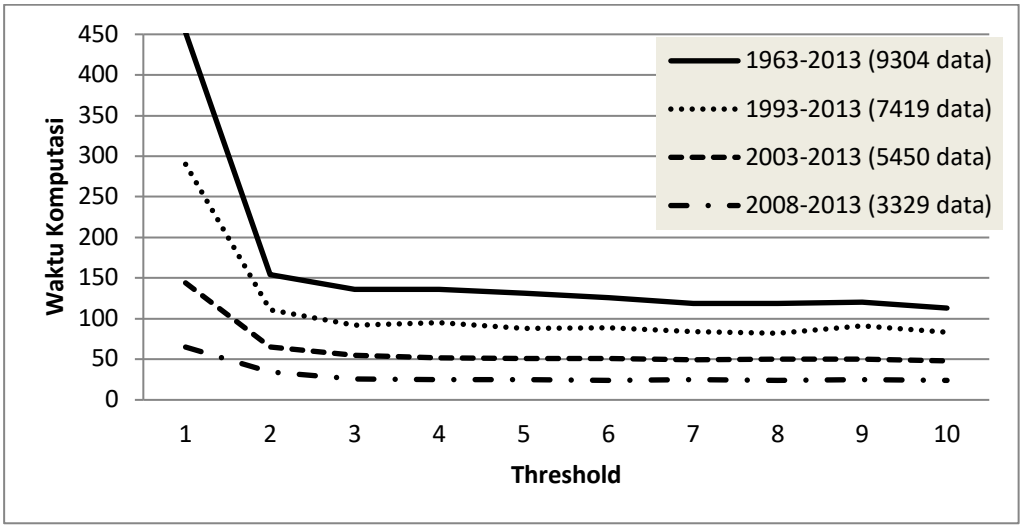
Pada uji coba kajian ini, dilakukan beberapa kali percobaan proses *indexing* menggunakan 4 buah koleksi data dengan jumlah data yang berbeda. Percobaan ini bertujuan untuk melihat keterhubungan antara jumlah data dan nilai *threshold* dengan waktu komputasi saat *indexing*, keterhubungan antara jumlah data dan nilai *threshold* dengan jumlah istilah subjek yang dihasilkan, serta keterhubungan antara jumlah data dan nilai *threshold* dengan jumlah data yang lolos *threshold*. Selain itu, percobaan ini bertujuan untuk menentukan nilai *threshold* yang dapat menghasilkan waktu komputasi yang efisien dan jumlah istilah yang optimal.

Dalam percobaan ini, disiapkan 4 buah koleksi data, yaitu data tahun 1963-2013 dengan data sebanyak 9.304 *record* data, data tahun 1993-2013 sebanyak 7.419 *record* data, data tahun 2003-2013 sebanyak 5.450 *record* data dan data tahun 2008-2013 sebanyak 3.329 *record* data. Setelah dilakukan percobaan dengan menggunakan 10 *threshold*, maka didapatkan hasil seperti pada Gambar 4, Gambar 5, dan Gambar 6.

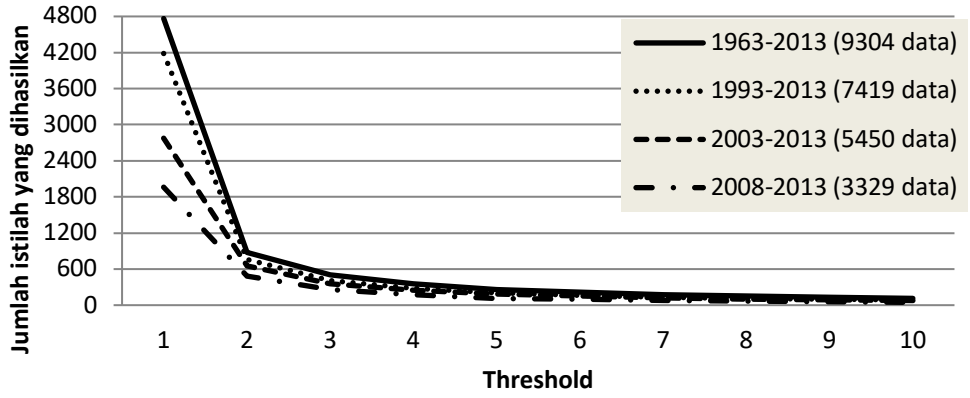
Dengan melihat Gambar 4, dapat disimpulkan bahwa semakin besar data yang akan di-*index*, semakin besar pula waktu komputasi yang dihasilkan. Waktu komputasi sangat tinggi untuk *threshold*=1 dengan jumlah data berapapun sedangkan untuk *threshold*=2 dan seterusnya waktu komputasi menurun. Hal ini terjadi karena tidak ada data yang tersisihkan dari nilai *threshold* yang diberikan atau dapat dikatakan bahwa semua data di-*index* secara keseluruhan sehingga waktu komputasi menjadi sangat tinggi. Akan tetapi, nilai *threshold* tidak mempengaruhi waktu komputasi secara signifikan. Terlihat dari perbedaan yang tidak begitu terlihat antara waktu komputasi yang dihasilkan *threshold*=3 sampai *threshold*=10.

Dengan melihat Gambar 5 dan grafik Gambar 6, dapat disimpulkan bahwa semakin besar data yang akan di-*index*, semakin banyak pula istilah subjek yang akan dihasilkan dan semakin banyak data yang lolos *threshold*. Jumlah istilah semakin menurun seiring bertambahnya nilai

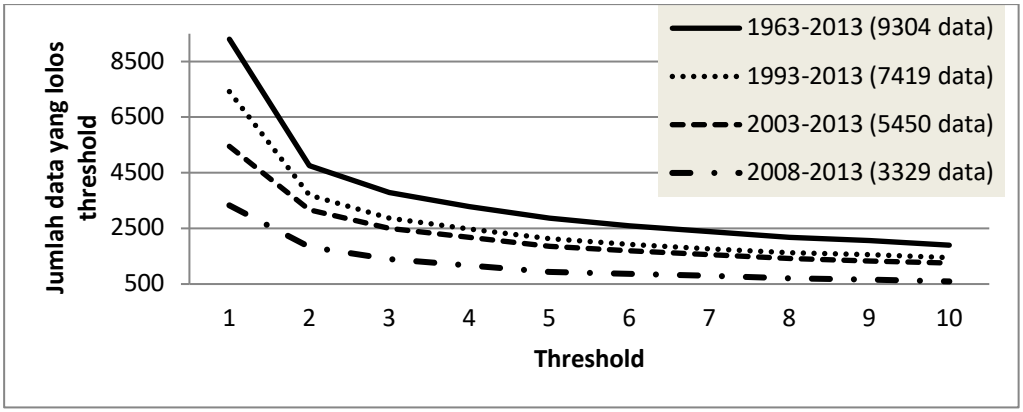
threshold. Hal ini terjadi karena semakin tinggi nilai *threshold*, semakin banyak data yang tersisihkan dan tidak mencukupi untuk lolos dari *threshold* yang ditentukan sehingga jumlah istilah subjek yang dihasilkan semakin sedikit. Oleh karena itu, dapat dikatakan bahwa nilai *threshold* mempunyai pengaruh yang signifikan terhadap jumlah istilah subjek yang dihasilkan.



Gambar 4 Grafik ketehubungan jumlah data dengan waktu komputasi



Gambar 5 Grafik ketehubungan jumlah data dengan jumlah istilah yang dihasilkan



Gambar 6 Grafik ketehubungan jumlah data dengan jumlah data yang lolos *threshold*

Berdasarkan Gambar 4, Gambar 5 dan Gambar 6, dapat dianalisis bahwa nilai *threshold* yang dapat menghasilkan waktu komputasi yang efisien adalah nilai *threshold* antara 3 sampai 10 karena pada nilai *threshold* antara 3 sampai 10 waktu komputasi cenderung sama. Untuk menentukan nilai *threshold* yang menghasilkan jumlah istilah subjek yang optimal, perlu dianalisis terlebih dahulu berapa jumlah istilah yang dianggap optimal dengan cara menghitung persentase antara jumlah istilah subjek dengan jumlah data yang lolos *threshold*. Persentase tersebut menunjukkan keragaman dari istilah yang dihasilkan terhadap jumlah data yang lolos *threshold*. Semakin kecil persentase ragam semakin baik karena istilah yang dihasilkan akan dapat mewakili data yang banyak sehingga dapat dikatakan optimal. Misalkan untuk data tahun 1963-2013 dengan nilai *threshold*=3, jumlah data yang lolos *threshold* berjumlah 3.782 dan jumlah istilah yang dihasilkan sebanyak 503 subjek, maka persentasenya sebesar 13,29%. Untuk data yang sama, nilai *threshold*=4 persentasenya 10,72% dan untuk nilai *threshold*=5 persentasenya sebesar 9,19%.

Tujuan dilakukannya perhitungan persentase ragam ini adalah untuk mendapatkan persentase yang terkecil dengan mempertimbangkan jumlah data yang lolos *threshold*. Oleh karena itu, pada penelitian ini ditetapkan persentase minimum untuk ragam sebesar 10% sehingga nilai *threshold* yang dipilih adalah 4 sebagai nilai *threshold* yang menghasilkan jumlah istilah subjek yang optimal. Selanjutnya, dilakukan proses *thresholding* terhadap data yang sebelumnya sudah dikenai *stemming*. Setelah itu, akan didapat daftar istilah baru yang memungkinkan bersih dari data yang kurang sesuai. Selanjutnya, dilakukan proses pembuatan *inverted index*.

Perhitungan Peluang Antaristilah

Proses perhitungan peluang dimulai dengan membuat kombinasi antaristilah dari daftar istilah yang sudah dibuat. Kombinasi tersebut merupakan pasangan-pasangan istilah yang mungkin berpasangan pada satu *field* subjek tesis. Proses perhitungannya adalah sebagai berikut.

1. Subjek yang nantinya akan dilihat keterhubungannya dinamakan subjek *parent* dan subjek-subjek berhubungan dinamakan subjek *child*.
2. Merujuk ke persamaan (2), maka yang bertindak sebagai *a* adalah *child* dan *parent* bertindak sebagai *b*.
3. Langkah pertama adalah menghitung jumlah tesis yang mengandung istilah subjek *parent* pada daftar istilah, lalu menghitung juga jumlah tesis yang mengandung istilah subjek *parent* dan *child* yang berada pada satu tesis. Setelah nilai keduanya sudah diketahui, maka dihitung perbandingan antara jumlah tesis yang mengandung istilah subjek *parent* dan *child* yang berada pada satu tesis dengan jumlah tesis yang mengandung istilah subjek *parent* pada daftar istilah. Perbandingan antara jumlah keduanya itu disebut peluang subjek *child* jika subjek *parent* yang dipilih. Jika nilai peluang antaristilah tersebut sudah dihitung, maka nilai peluang tersebut disimpan beserta istilah yang bersangkutan ke dalam *database*.

Tabel 3 adalah contoh nilai peluang beserta istilah yang bersangkutan. Subjek2 merupakan subjek *parent* sedangkan Subjek1 adalah subjek *child*. Diketahui bahwa istilah subjek *agribusi* pada daftar istilah berjumlah 18 sedangkan istilah subjek *agricultur econom* berjumlah 51. Lalu istilah subjek *agribusi* dan *agricultur econom* yang terletak pada 1 tesis berjumlah 5. Maka hasil perhitungan peluangnya seperti pada Tabel 3.

Tabel 3 Contoh nilai peluang

Subjek1	Subjek2	Peluang
agribusi	agricultur econom	0.098039
agricultur econom	agribusi	0.277778

Perancangan Visualisasi dan Antarmuka

Proses perancangan visualisasi dimulai dengan pembuatan *script* PHP untuk menghubungkan koleksi data dan daftar istilah yang tersimpan di *database* MySQL dengan *script* untuk visualisasi. Pada penelitian ini, *script* untuk visualisasi menggunakan *script* JQuery JIT. Setelah koleksi data sudah terhubung dengan *script* visualisasi, dilakukan pengaturan terhadap atribut-atribut visualisasi, seperti ukuran dan bentuk *node*, warna *node* dan *edge*, jarak antara *node parent* dan *node child*, serta warna *background* visualisasi. Setelah selesai, maka dilanjutkan dengan pembuatan antarmuka pengguna.

Sistem yang telah dibuat umumnya mengikuti sistem dengan *QB paradigm* (*Query searching and Browsing*), yaitu pengguna akan diminta memasukkan *query* berupa istilah subjek, lalu sistem akan menampilkan hasil pencarian tersebut, selanjutnya pengguna akan menelusuri istilah-istilah yang terhubung dengan istilah yang dimasukkan sebagai *query* pada visualisasi yang ditampilkan (Zhang 2008).

Antarmuka pengguna terbagi menjadi 2 bagian, yaitu bagian untuk visualisasi dan bagian untuk *background processing*. Antarmuka bagian visualisasi adalah bagian yang mencakup antarmuka utama untuk menampilkan visualisasi dari keterhubungan istilah *query* yang dimasukkan pengguna beserta fitur-fitur interaksi visualisasi, sedangkan antarmuka bagian *background processing* adalah bagian yang mencakup antarmuka untuk pengguna yang menangani koleksi data yang akan ditampilkan dalam visualisasi.

Fitur-fitur interaksi yang terdapat pada bagian visualisasi antara lain: *search box* untuk pencarian istilah pada daftar istilah, halaman yang memuat daftar istilah untuk memudahkan pengguna memilih istilah yang akan dimasukkan sebagai *query*, penambahan animasi dan *link* pada *child*, serta fitur *spanning*, *dragging* dan *zooming* untuk hasil visualisasi. Antarmuka bagian *background processing* terdiri atas halaman untuk menampilkan informasi tentang koleksi data sedang digunakan, halaman untuk melakukan proses *indexing* termasuk proses impor data dari XLS ke MySQL, dan halaman untuk menampilkan daftar istilah yang digunakan saat ini. Gambar 7 adalah tampilan antarmuka bagian visualisasi.

Dapat dilihat dari Gambar 7, visualisasi yang ditampilkan adalah hasil visualisasi dari *query forestry*. Istilah *forestry* bertindak sebagai *node parent* dengan jumlah *node child* sebanyak 19 buah. Hasil visualisasi tersebut mengandung makna bahwa istilah subjek *forestry* memiliki keterhubungan dengan 19 istilah lainnya. Oleh karena itu, dapat dikatakan bahwa 19 *child* tersebut memiliki kesamaan yaitu, berada dalam satu *field* subjek dengan subjek *forestry* di dalam metadata tesis IPB.

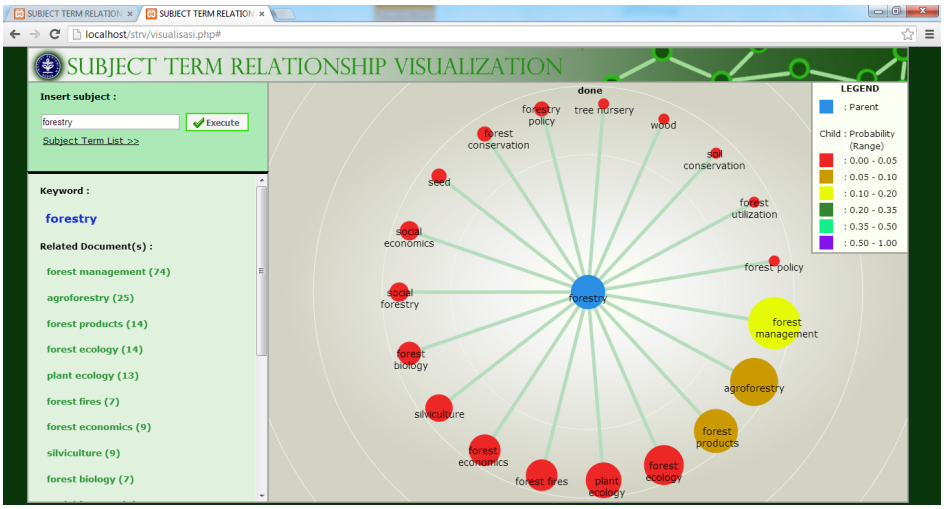
Terdapat perbedaan pada setiap *node child* baik segi warna maupun ukuran. Pada kajian ini, perbedaan warna menunjukkan *range* nilai peluang, sedangkan ukuran *node* menunjukkan rasio dari peluang untuk setiap *child* yang berhubungan dengan *parent*. Selain itu, arah putar berlawanan jarum jam pada *node-node child* dari hasil visualisasi membentuk urutan peluang dari peluang terkecil hingga yang terbesar. Jadi, dapat disimpulkan bahwa istilah subjek *forest management* memiliki hubungan paling erat dengan subjek *forestry* dibandingkan dengan istilah subjek lainnya.

TENTANG SISTEM

Sistem Visualisasi dan Penelusuran Keterhubungan Subjek dibangun menggunakan *platform* web dengan PHP sebagai bahasa pemrogramannya dan MySQL sebagai aplikasi manajemen basis data. Sistem ini bersifat *open source* sehingga dapat dikembangkan lebih lanjut oleh berbagai pihak yang ingin melakukan optimasi terhadap sistem ini. Kode sumber serta tata laksana instalasi dan penggunaan sistem ini dapat diunduh pada http://repository.ipb.ac.id/strv.

SIMPULAN

Pada kajian ini telah dipaparkan pembangunan sistem penelusuran dokumen karya ilmiah berdasarkan keterhubungan subjek dan menampilkan hasil visualisasinya. Sistem ini memiliki keunggulan visualisasi sederhana yang mudah dimengerti oleh pemustaka. Sistem ini juga semaksimal mungkin telah melalui tahap optimasi agar dapat memberikan layanan penelusuran yang efektif dan efisien. Dengan pemaparan pada kajian ini diharapkan perpustakaan dapat meningkatkan layanan kepada pemustaka dalam mengembangkan pengetahuan sehingga mendukung terciptanya iklim inovasi dan invensi di Indonesia sesuai dengan Tujuan Pembanguan Berkelanjutan.



Gambar 7 Antarmuka bagian visualisasi

REFERENSI

Chase P, D'Amore R, Gershon N, Holland R, Hyland R, *et al*. Semantic Visualization. *ACL-COLING Workshop on Content Visualization and Intermedia Representations.* 1998 Agustus 15; Montreal, Canada. New Brunswick(US): ACL. hlm 52-62.

Kboubi F, Habacha A, BenAhmed M. 2012. Semantic visualization and navigation in textual corpus. *International Journal of Information Sciences and Techniques*. 2(1):53-62.

Manber U, Smith M, Gopal B. 1997. Webglimpse: Combining browsing and searching. *USENIX 1997 Annual Technical Conference*; 1997 Jan 6-10; Anaheim, California. Berkeley(US): USENIX Association. hlm 195-206.

Manning CD, Raghavan P, Schutze H. 2008. *Introduction to Information Retrieval.* New York(US): Cambridge University Press.

Olston C, Chi E. 2003. ScentTrails: integrating browsing and searching on the web. *ACM Transactions on Computer-Human Interaction*. 10(3):177-197.

[PNRI] Perpustakaan Nasional Republik Indonesia. 2005. *INDOMARC Format MARC Indonesia*. Jakarta(ID): Perpustakaan Nasional Republik Indonesia.

[UNESCO] United Nations Educational, Scientific dan Cultural Organization. 2004. *CDS/ISIS for Windows Reference Manual (Version1.5).* Nelson(NZ): UNESCO.

Zhang J. 2008. *Visualization for Information Retrieval*. Berlin(DE): Springer-Verlag.