SISTEM PENCARIAN FORUM BERBASIS ONTOLOGI DAN LABEL

Adi Wibowo[1)], **Gregorius Satiabudhi**[2)] , **Yulius Pranata**[3)]
[1,2,3)]Program Studi Teknik Informatika Universitas Kristen Petra Surabaya
Jl. Siwalankerto 121-131 Surabaya Telp (031)-2983455
e-mail : [1)] adiw@petra.ac.id, [2)] greg@petra.ac.id

Abstrak
Salah satu kegiatan yang sering dilakukan pengguna internet adalah berdiskusi melalui sebuah forum. Setiap forum terbagi ke dalam beberapa kategori, dan setiap kategori memiliki beberapa percakapan (thread). Setiap percakapan dapat diberi label (tag) baik oleh pengguna yang membuka percakapan tersebut, atau peserta forum. Masalah yang muncul adalah bila sebuah forum telah menjadi besar mencari percakapan yang tepat yang sesuai kebutuhan pengguna menjadi lebih sulit. Penelitian ini bertujuan mengusulkan metode rekomendasi percakapan di sebuah forum internet yang menggunakan label dan didukung oleh ontologi yang sesuai dengan kategori percakapan.Penelitian ini mengkhususkan pada forum dengan kategori teknologi komputer. Forum tersebut adalah StackOverflow. Pada setiap percakapan pada forum tersebut terdapat label-label yang menjelaskan isi percakapan, misalnya C#, recursive, programming, dll. Label-label tersebut dikembangkan (diperbanyak) menggunakan ontologi komputer. Hasilnya adalah setiap percakapan akan diwakili oleh label-label asli dari StackOverflow, dan ditambah dengan label-label baru yang berasal dari ontologi. Label-label tersebut digunakan dalam proses pencarian berbasis Vector Space Model (VSM).Hasil penelitian menunjukkan bahwa penggunaan ontologi sebagai metode keyword extension meningkatkan nilai recall dari metode VSM tersebut.
***Kata Kunci** : Forum, Ontologi, Label, Search*

1. PENDAHULUAN

Forum menyediakan tempat bagi pengguna untuk saling berkomunikasi. Setiap pengguna dapat mengajukan pertanyaan, atau pendapat mengenai sebuah topik dalam sebuah percakapan (*thread*) antar pengguna. Sebuah topik yang hampir sama dapat dibahas di beberapa percakapan, misalnya topik tentang *agile development* menghasilkan 927 percakapan di forum StackOverflow.com.

Pada forum yang besar tidak semua percakapan dapat dilihat secara langsung. Pengguna akan menggunakan fasilitas pencarian untuk menemukan topik-topik percakapan yang sesuai dengan kebutuhannya. Bila seorang pengguna telah menemukan dan kemudian membaca sebuah percakapan, biasanya pengguna ingin mencari percakapan lain yang masih setopik dengan apa yang ia butuhkan. Untuk mendapatkan percakapan lain yang setopik pengguna itu akan kembali ke halaman pencarian untuk melihat percakapan lainnya yang dihasilkan oleh mesin pencari di forum tersebut. Tentunya pengguna akan mendapatkan kemudahan bila di bagian bawah halaman percakapan tersebut terdapat daftar percakapan yang mirip dengan halaman percakapan yang sedang ia lihat. Untuk itu dibutuhkan sebuah sistem rekomendasi yang dapat mengetahui topik-topik apa yang dapat mewakiliki percakapan tersebut, dan dapat mengetahui halaman-halaman percakapan lain apa saja yang mirip dengan topik yang sedang dibaca pengguna.

Untuk mendapatkan topik dari sebuah percakapan dapat digunakan pendekatan seperti *part-of-speech tagging* untuk mengenali bagian-bagian dari sebuah kalimat dan menentukan bagian mana yang lebih penting dalam mewakili kalimat tersebut. Pendekatan lain adalah menggunakan *suffix tree clustering* untuk mengelompokkan frase-frase dari sebuah dokumen dan menentukan sebagian cluster kalimat yang dianggap dapat mewakili topik dokumen tersebut. Pada penelitian ini tidak digunakan pendekatan otomatis seperti di atas karena pendekatan otomatis masih memiliki kelemahan yaitu frase yang dihasilkan tidak selalu menghasilkan frase paling mewakili isi dokumen. Topik sebuah percakapan didapatkan dari label-label yang diberikan pengguna sendiri atau pembaca dari percakapan. Hal ini menyebabkan label topik yang diberikan selalu relevan dengan topik sesungguhnya dari percakapan tersebut.

Untuk memberikan rekomendasi percakapan-percakapan lain yang sesuai dengan percakapan yang sedang dibaca, perlu ditentukan sistem pencarian yang dapat menemukan kemiripan antar dokumen. Penelitian ini mengusulkan pemakaian algoritma yang cukup banyak digunakan, yaitu vector space model (vsm). Vector Space Model memiliki kelemahan, yaitu tidak dapat menemukan relasi antar dua dokumen yang sebenarnya memiliki topik yang sama, tetapi menggunakan term-term yang berbeda. Untuk mengatasi perbedaan term yang sebenarnya memiliki makna / topik hampir sama tersebut pada penelitian ini digunakan ontologi. Ontologi dapat memberikan relasi antara term-term yang saling berkaitan dalam sebuah domain pengetahuan tertentu.

2. TINJAUAN PUSTAKA

Pablo Castells, et.al. dalam penelitiannya yang berjudul "An Adaptation of the Vector-Space Model for Ontology-Based Information Retrieval" (Pablo Castells, 2007) mengusulkan penggunaan *semantic indexing* untuk melengkapi pencarian berbasis Vector Space Model. Pada pendekatan ini setiap dokumen akan dilengkapi dengan file Resource Description Framework (RDF). Di dalam setiap file RDF terdapat *annotation* yang berasal dari kelas-kelas ontologi yang digunakan oleh sekumpulan dokumen tersebut. Dengan adanya *annotation* tersebut maka setiap dokumen juga dapat dicari menggunakan RDF Data Query Language (RDQL). RDQL adalah sebuah bahasa yang digunakan oleh *query engine* berbasis semantik untuk menemukan dokumen yang sesuai dengan topik. Masalah yang muncul adalah bahwa saat ini belum ditemukan adanya forum yang menggunakan RDF untuk melengkapi deskripsi sebuah percakapan. RDF dapat dibuat secara manual oleh staf administrasi untuk setiap percakapan, tetapi hal ini tentunya menambah kompleks penanganan sebuah forum.

Forum-forum yang ada di internet saat ini sudah banyak menggunakan label (tag) untuk memperjelas topik-topik dari sebuah percakapan. Sebagai contoh forum StackOverflow.com menggunakan label-label seperti *agile*, *project-management*, atau *apache* yang dapat ditambahkan untuk membantu pengguna forum memahami topik dari percakapan yang sedang dibaca. Penelitian ini mengusulkan penggunaan ontologi untuk mengembangkan label (label extension) untuk mendukung pencarian berbasis *vector space model*.

Vector Space Model adalah sebuah model untuk mengukur kemiripan antara dua dokumen. Setiap dokumen dipetakan ke sekumpulan *weighted terms* sehingga membentuk vektor (Gerard Salton, 1987). Bobot setiap term bisa didapatkan dari beberapa komponen, yaitu term frequency, collection frequency, dan komponen normalisasi. Bobot term ditunjukkan oleh persamaan (1). Bobot suatu term semakin besar jika term tersebut sering muncul dalam satu dokumen dan semakin kecil jika term tersebut muncul dalam banyak dokumen. Pada *vector space model* perlu diketahui bobot setiap *term* yang terdapat di sebuah dokumen, dan term yang terdapat di sebuah query.

$$\text{Term Weight} = \quad w_i = tf_i * log\left(\frac{D}{df_i}\right) \qquad (1)$$

tfi = frekuensi *term* atau banyak *term* i yang ada pada sebuah dokumen
dfi = frekuensi dokumen atau banyak dokumen yang mengandung *term* i
D = banyaknya dokumen yang terdapat pada *database*

Vektor yang terbentuk dari weighted terms untuk mewakili dokumen D dan query Q ditunjukkan pada persamaan (2).

$$\vec{d} = (w_{d1}, w_{d2}, w_{d3} \ldots, w_{dk})$$
$$\vec{q} = (w_{q1}, w_{q2}, w_{q3} \ldots, w_{qk}) \qquad (2)$$

Untuk mengukur kemiripan antar dokumen dengan *query* digunakan persamaan (3). w_{qk} adalah bobot term k pada query, sedangkan w_{dk} adalah bobot term k pada dokumen d. t adalah jumlah term yang unik di seluruh dokumen yang tersimpan di database.

$$\text{similarity}(Q,D) = \frac{\sum_{k=1}^{t} w_{qk} \cdot w_{dk}}{\sqrt{\sum_{k=1}^{t} (w_{qk})^2 \cdot \sum_{k=1}^{t} (w_{dk})^2}} \qquad (3)$$

Ontologi adalah representasi dari sebuah bidang pengetahuan (Liu, 2009). Representasi dapat berbentuk kelas dari konsep pengetahuan, atribut (*property*), atau hubungan (*relationship*) antar konsep dalam sebuah bidang pengetahuan. Representasi dapat digunakan untuk menjelaskan makna, atau menunjukkan batasan-batasan dalam penggunaan konsep pengetahuan tersebut. Dalam penelitian ini digunakan ontologi yang disediakan oleh Freebase. Freebase menyediakan ontologi untuk 76 konsep pengetahuan yang disebut dengan domain. Beberapa contoh dari domain yang disediakan oleh Freebase adalah *physics*, *computers*, *religion*, *symbols*, *language*,

aviation, *theatre*, dan *astronomy*. Setiap domain pengetahuan terdiri dari topik-topik. Setiap topik dijelaskan menggunakan beberapa tipe (Freebase, 2013). Tipe adalah *conceptual container* yang menjelaskan topik dari sebuah perspektif. Untuk menjelaskan topik dari perspektif tertentu, setiap tipe memiliki beberapa property. Beberapa contoh tipe untuk domain *computer* adalah *Operating System*, *Programming Language*, *Programming Language Paradigm*, *File Format*. Contoh property dari tipe *Programming Language* adalah *Parent Language, Language Paradigm, Introduced, Influenced by, Influenced, Dialects.* Setiap property akan memiliki suatu nilai, misalnya tipe matlab memiliki property *influenced by* berisi APL, property *language paradigm* berisi *Imperative programming, array programming, object oriented programming*, property *software genre* berisi *mathematics*, dan *numerical data*. Setiap isi dari property dilengkapi dengan fulltext yang memberikan deskripsi atau penjelasan lengkap tentang isi property tersebut.

Pada penelitian ini label yang didapat dari percakapan di forum akan dibandingkan dengan tipe dari ontologi Freebase. Bila label tersebut ada di dalam ontologi sebagai tipe, setiap nilai property dari tipe tersebut akan digunakan sebagai label extension untuk membantu memberikan gambaran topik percakapan yang lebih lengkap dibandingkan hanya menggunakan label asli.

3. METODE PENELITIAN

Untuk mengetahui apakah penggunaan ontologi dapat digunakan sebagai *label extension* yang membantu pencarian forum berbasis VSM, penelitian ini menggunakan Freebase sebagai sumber ontologi, dan StackOverflow sebagai sumber forum yang diuji. Tidak semua domain pengetahuan di Freebase digunakan sebagai sumber ontologi. Hanya domain *computers* saja yang digunakan dalam penelitian ini agar sesuai dengan topik-topik percakapan di forum StackOverflow yang adalah forum tentang teknologi komputer.

Penelitian ini mengunduh sekitar 585 percakapan dari StackOverflow secara acak. Setelah 585 percakapan diunduh kemudian diproses untuk siap diindex oleh VSM. Proses persiapan meliputi:

1. Menghilangkan stopwords dari bahasa Inggris
2. Membentuk root words melalui Porter Stemmer
3. Memeriksa apakah term-term dalam percakapan tersebut adalah *irregular verbs* dalam bahasa Inggris agar tidak perlu melalui proses *stemming*, dan
4. Menentukan apakah sekumpulan term membentuk frase yang juga harus dihindarkan dari proses stemming.

Dua pengujian dilakukan setelah setiap term dari seluruh percakapan yang diunduh dan diproses. Pengujian tipe pertama adalah melakukan proses pencarian menggunakan label dan judul percakapan sebagai sumber term untuk membentuk vektor setiap dokumen. Pengujian tipe kedua menambahkan term-term baru yang didapatkan dari property dari tipe domain *computers*. Setiap label baru yang didapatkan dari pemanfaatan ontologi juga akan mendapatkan proses sama seperti setiap term dari dokumen di atas. Hasil pengujian akan dibandingkan untuk mengetahui pengujian mana yang menghasilkan nilai recall dan precission yang lebih baik.

Precision dan *Recall* adalah salah satu metode untuk melakukan evaluasi pada kinerja dari sistem *information retrieval*. *Precision* merupakan jumlah dokumen relevan yang ditemukan dengan total jumlah dokumen yang ditemukan oleh search engine. *Precision* mengindikasi kualitas himpunan hasil pencarian, tetapi tidak memandang jumlah dokumen yang relevan dalam kumpulan dokumen. Maka dari itu diperlukan *Recall* yaitu rasio jumlah dokumen relevan yang ditemukan dengan total jumlah dokumen dalam kumpulan dokumen yang dianggap relevan. (Manning, 2008).

$$\text{precision} = \frac{\text{number of relevant items retrieved}}{\text{total number of items retrieved}} \qquad (4)$$

$$\text{recall} = \frac{\text{number of relevant items retrieved}}{\text{number of relevant items in collection}}$$

4. HASIL DAN PEMBAHASAN

Untuk pengujian digunakan percakapan berjudul "*What event is raised when a user interacts with the DateTimePicker control?*". Percakapan tersebut di StackOverflow memiliki label C#, *.net, winforms*, dan *datetimepicker*. Label *winforms*, dan *datetimepicker* belum ada dalam ontologi di Freebase sehingga tidak dapat diproses lebih lanjut. Dari label C# didapatkan beberapa nilai property seperti ditunjukkan pada Gambar 1.

key	value
dialects	CÏ‰,Spec sharp,Cobra,Polyphonic C sharp
influenced	CÏ‰,Windows PowerShell,PHP,Java,F#,D,Objeck,Nemerl...
influenced_by	Object Pascal,C,Modula-3,C++,Haskell,Java,Microsof...
introduced	2001
language_designers	Microsoft,Anders Hejlsberg
language_paradigms	Multi-paradigm programming language,Generic progra...

Gambar 1. Nilai Property dari Label "C#"

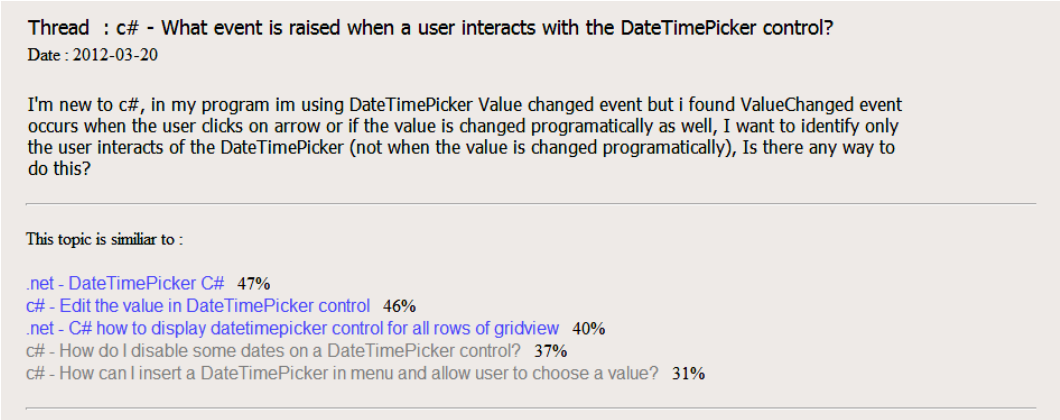
Dari label .net didapatkan nilai property seperti ditunjukkan pada Gambar 2.

key	value
file_formats_supported	Disco

Gambar 2. Nilai Property dari Label ".net"

Secara keseluruhan label dari percakapan dikembangkan menjadi *Cobra, Polyphonic C sharp, C#, Windows PowerShell, PHP, Java, F#, D, Objeck, Nemerle, Vala, Oxygene, Cobra, Vala, Polyphonic C sharp, Spec sharp, Fan, Object Pascal, C, Modula-3, C++, Haskell, Java, Microsoft Silverlight, XAML, Eiffel, 2001, Microsoft, Anders Hejlsberg, Multi-paradigm programming language, Generic programming, Component-oriented programming, Object-oriented programming, Structured programming, Disco, winforms, datetimepick, .net.*

Dari pengujian tipe 1 yang hanya menggunakan judul dan label asli didapatkan recall sebesar 0,452, sedangkan pengujian tipe 2 menghasilkan recall sebesar 0,562. Untuk precision pengujian tipe 1 menghasilkan nilai 0,332 dan pengujian tipe 2 menghasilkan nilai 0,297. Hasil rekomendasi percakapan menggunakan hasil pengujian tipe 2 ditunjukkan pada Gambar 3.



Gambar 3. Hasil Rekomendasi Percakapan yang Mirip dengan
Percakapan Berjudul "*What event is raised when a user interacts with the DateTimePicker control?*"

Untuk memperjelas pengaruh ontologi pada VSM dilakukan pengujian dengan beberapa frase label. Hasil pengujian ditunjukkan pada Tabel 1.

Tabel 1. Hasil Pengujian VSM dan Label Expansion

Frase mewakili label	Dok. Relevan	Pengujian Tipe 1				Pengujian Tipe 2			
		Retr.	Relevant	Recall	Prec.	Retr.	Relevant	Recall	Prec.
web services	21	35	18	0.857	0.514	44	20	0.952	0.455
matlab	8	24	5	0.625	0.208	26	6	0.75	0.231
php	20	22	19	0.95	0.864	25	19	0.95	0.76
boost	6	6	5	0.833	0.833	11	5	0.833	0.455

Rata-rata recall untuk pengujian tipe 1 adalah 0,816, sedangkan untuk pengujian tipe 2 sebesar 0,871. Untuk pengukuran precission pengujian tipe 1 menghasilkan rata-rata 0.605, sedangkan untuk pengujian tipe 2 sebesar 0,475. Dari nilai rata-rata di atas terlihat bahwa terdapat peningkatan nilai recall, sedangkan nilai precision mengalami penurunan. Penurunan precision disebabkan jumlah percakapan yang dianggap mirip dengan percakapan yang diuji mengalami peningkatan akibat lebih banyaknya term yang digunakan oleh VSM.

Pada contoh pengujian dengan frase "php" ditemukan bahwa ada satu percakapan yang seharusnya relevan tetapi tidak ditemukan oleh pengujian. Hal ini disebabkan label dari percakapan tersebut adalah "php5" sedangkan ontologi yang berasal dari freebase hanya memiliki frase "php". Hal ini menunjukkan bahwa bila ontologi tidak menunjukkan hubungan sebuah frase dengan frase lainnya, maka vsm tidak akan mampu menemukan relasi antar percakapan yang memiliki frase yang seharusnya saling berhubungan tersebut.

5. KESIMPULAN

Penggunaan ontologi dapat meningkatkan nilai recall dari pencarian berbasis vector space model. Kerugian yang didapatkan adalah menurunnya nilai precision dibanding tanpa menggunakan ontologi sebagai keyword extension. Kelemahan dari pendekatan ini juga adalah bahwa ontologi yang digunakan mempengaruhi kualitas peningkatan recall dari VSM.

DAFTAR PUSTAKA

Castells, Pablo., Miriam Fernandez, and David Vallet, 2007, *An Adaptation of the Vector-Space Model for Ontology-Based Information Retrieval*, Vol. 19, IEEE Transactions on Knowledge and Data Engineering, pp 261-272.

Freebase, 2013, *Freebase API*, https://developers.google.com/freebase/guide/basic_concept diakses pada tanggal 30 April 2013.

Liu, Ling, and M. Tamer Özsu (Eds.), 2009, *Encyclopedia of Database Systems*, Springer-Verlag.

Manning, Christopher D., Prabhakar Raghavan and Hinrich Schütze, 2008, *Introduction to Information Retrieval*, Cambridge University Press.

Salton, Gerard and Chris Buckley, 1987, *Term Weighting Approaches in Automatic Text Retrieval*, Technical Report TR87-881, Department of Computer Science, Cornell University,. Information Processing and Management Vol.32 (4), p. 431-443.