

MEREKONSTRUKSI METODA PENCARIAN LITERATUR PADA PERPUSTAKAAN DIGITAL MENGGUNAKAN ALGORITMA EXTENDED WEIGHTED TREE SIMILARITY

Firdaus Solihin

fsolihin@gmail.com

Jurusan Teknik Informatika

Fakultas Teknik, Universitas Trunojoyo

ABSTRAK

Dewasa ini, perpustakaan digital (*e-library*) sebagai salah satu hasil implementasi teknologi informasi, semakin banyak bermunculan. Setiap perguruan tinggi berlomba-lomba mengembangkan perpustakaan digital disamping keberadaan perpustakaan secara fisik, hal ini terkait arti pentingnya keberadaan perpustakaan yang sering dianalogikan sebagai jantung dari sebuah perguruan tinggi. Mesin pencarian literatur adalah fasilitas penting yang harus tersedia pada suatu perpustakaan. Saat ini banyak sekali metoda pencarian yang telah dipakai pada berbagai mesin pencarian. Persoalan yang sering timbul dari suatu metoda pencarian adalah masalah keakuratan pencarian dan sejauh mana mesin pencarian dapat memberikan hasil yang tidak hanya tepat tapi juga memiliki tingkat kedekatan kesamaan sehingga dapat lebih memuaskan pengguna. Pada penelitian ini disajikan suatu konsep untuk merekonstruksi metoda mesin pencarian pada perpustakaan digital menggunakan *algoritma extended weighted tree similarity*.

Kata Kunci: *Algoritma Extended Weighted Tree Similarity, Matchmaking, Similarity, E-library.*

1. PENDAHULUAN.

Pemanfaatan teknologi informasi pada berbagai bidang saat ini sedang berkembang sangat pesat, tidak terkecuali di bidang pendidikan. Perpustakaan Digital (*e-Library*) adalah salah satu pengembangan dan pemanfaatan teknologi informasi di bidang pendidikan.

Search Engine atau mesin pencarian literatur adalah merupakan salah satu fungsi penting yang harus ada pada sebuah *e-library*. Banyak sekali metoda yang telah dipakai untuk merepresentasikan mesin pencarian ini.

Persoalan yang sering timbul dari mesin pencarian literatur adalah masalah keakuratan pencarian dan sejauh mana mesin pencarian dapat memberikan hasil yang tidak hanya akurat tapi juga dapat memberikan hasil pencarian yang memiliki urutan tingkat kedekatan kesamaan sehingga hasil ini dapat dijadikan alternative bagi user untuk digunakan.

Pada penelitian ini disajikan suatu konsep untuk melakukan rekonstruksi metoda pencarian literature dalam perpustakaan digital menggunakan algoritma extended weighted tree similarity. Penggunaan arsitektur ini diharapkan dapat meningkatkan keakuratan hasil pencarian serta merepresentasikan hasil pencarian yang memiliki kedekatan yang tinggi sesuai dengan masukan yang akan dicari.

Paper ini terbagi menjadi 6 bagian, dimana diawali pendahuluan di bagian 1, dan bagian 2 menjelaskan mengenai Representasi Tree pada Matchmaking yang kemudian diteruskan dengan pengenalan Algoritma Extended Weigthed Tree Similarity pada bagian 3, sedangkan bagian 4 mengemukakan Rekonstruksi dan pemakaian algoritma extended weighted tree similarity pada

pencarian literatur. bagian 5 menyajikan hasil percobaan pada penelitian ini. serta ditutup dengan Bagian 6 yang menyajikan simpulan.

2. REPRESENTASI TREE PADA MATCHMAKING.

Sistem multi agent berasal dari distributed artificial intelligence yang merupakan salah satu cabang ilmu artificial intelligence (AI). Dalam kerangka perkembangan teknologi informasi yang pesat saat ini membawa dampak yang baik bagi perkembangan teknologi agent dan multi agent sehingga telah dapat dengan mudah diimplementasikan secara nyata [1]. Banyak sekali contoh aplikasi yang menerapkan sistem agent dan multi agent ini seperti ACORN (Agen-base Community Oriented Routing Network), FIPA-OS yang dibuat oleh Foundation for Intelligent Physical Agent.

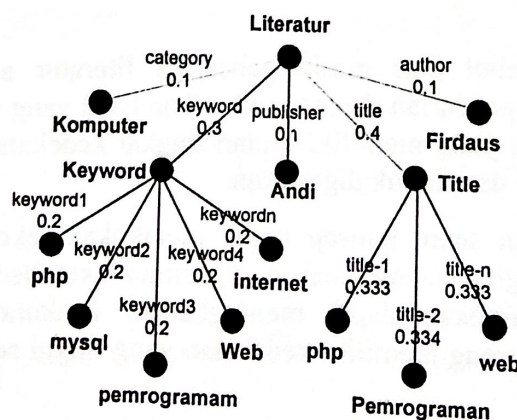
Perkembangan dari sistem multi agent juga tidak terlepas dari kelebihan yang dimilikinya seperti karakteristiknya yang otonomi, proaktif, intelligent (cerdas), corporate (mampu bekerja sama dengan agent lain).

Sistem multi agent dibangun atas 3 kategori yang umum, yaitu, provider agent, service requester agent, dan middle agent. Provider agent adalah penyedia layanan, seperti pencarian informasi dan memecahkan masalah yang spesifik. Requester agent adalah agent yang membutuhkan layanan tersebut. Sedangkan Middle Agent digunakan untuk mencari kecocokan antar agent. Proses pencocokan yang dilakukan oleh Middle agent ini disebut matchmaking.

Proses matchmaking fungsi utamanya untuk menemukan tingkat kesamaan (similarity) yang tertinggi, atau menemukan tingkat jarak (distance) yang terpendek antar representasi spesifikasi provider agent dan requester agent. urutan-urutan proses dalam matchmaking [3], sebagai berikut:

- o Provider agent menawarkan/mengiklankan (advertise) kapabilitasnya kepada middle agent.
- o Middle agent menyimpan informasi kapabilitas tersebut.
- o Requester agent menyampaikan permintaan ke middle agent, untuk mendapatkan provider agent yang sesuai dengan yang spesifikasi yang diinginkan.
- o Middle agent melakukan proses pencocokan antara request agent dengan provider agent, kemudian memberikan hasil berupa subset dari provider agent yang memiliki spesifikasi paling mendekati request tersebut.

Beberapa metode yang digunakan untuk merepresentasikan spesifikasi informasi yang digunakan dalam sistem multi agent antara lain adalah LARKS (Language for Advertisement and Request for Knowledge Sharing) [3], KQML (Knowledge Query and Manipulation Language) [7], EXPECT (Explicit Representations for flexible Acquisition) [8] dan representasi object berupa node labelled, arc labelled, arc weighted tree pada Object Oriented Rule Markup Language (OO RuleML) [6].



Gambar 2.1. Contoh representasi tree.

Metoda OO RuleML memiliki beberapa kelebihan dibanding metoda LARKS dan KQML, kelebihan-kelebihan tersebut antara lain: dapat mengakomodasi representasi object complete yang terstruktur secara fleksible, punya unsur bobot (weight) pada arc dan dapat mewakili tingkat prioritas, preferensi atau tingkat kepentingan. Sehingga proses matchmaking menjadi lebih sempurna [4]. Karena kelebihan-kelebihan inilah maka pada penelitian ini digunakan metoda OO RuleML.

Informasi yang dibawa oleh agent buyer dan agent seller pada metoda OO RuleML direpresentasikan dalam sebuah tree yang memiliki karakteristik node berlabel, arc berlabel dan arc berbobot. Contoh representasi tree bisa dilihat pada gambar 2.1.

Untuk menserialisasikan representasi tree saat terjadi pertukaran informasi antar agent di dalam web digunakan RuleML versi Object Oriented Modelling [6], yang disebut Weighted Object Oriented RuleML yang mengacu pada standarisasi XML (Extended Markup Language). Contohnya bisa dilihat pada gambar berikut:

```

<cterm>
  <_opc><ctor>Literatur</ctor></_opc>
    <_r n="category" w="0,1"><ind>Komputer</ind></_r>
    <_r n="keyword" w="0,3">
      <cterm>
        <_opc><ctor>Keyword</ctor></_opc>
          <_r n="keyword1" w="0,2"><ind>Php</ind></_r>
          <_r n="keyword2" w="0,2"><ind>mysql</ind></_r>
          <_r n="keyword3" w="0,2"><ind>Pemrograman</ind></_r>
          <_r n="keyword4" w="0,2"><ind>Web</ind></_r>
          <_r n="keywordn" w="0,2"><ind>Internet</ind></_r>
        </cterm>
      </_r>
    <_r n="title" w="0,4">
      <cterm>
        <_opc><ctor>Pemrograman Web dengan PHP</ctor></_opc>
          <_r n="Judul1" w="0,333"><ind>Php</ind></_r>
          <_r n="Judul2" w="0,334"><ind>Pemrograman</ind></_r>
          <_r n="Judul3" w="0,333"><ind>Web</ind></_r>
        </cterm>
      </_r>
    <_r n="publisher" w="0,1"><ind>Andi</ind></_r>
    <_r n="author" w="0,1"><ind>Firdaus</ind></_r>
  </cterm>

```

Gambar 2.2. Serialisasi representasi tree.

Pada gambar 2.2 terdapat beberapa simbol, adapun keterangan dari simbol-simbol ini antara lain:

- <cterm> = keseluruhan tree.
- <opc> = root dari tree.
- <ctor> = node label dari root.
- <_r> = role dari setiap arch/edge dan memiliki beberapa attribute yaitu n mewakili label dan w yang mewakili bobot/weight.
- <ind> = label untuk role.

Sub tree dari sebuah role memiliki struktur yang sama/indentik yang diawali dengan <cterm> dan seterusnya seperti pada gambar diatas.

3. ALGORITMA EXTENDED WEIGHTED TREE SIMILARITY.

Algoritma Extended Weighted Tree Similarity adalah salah satu metode pencarian tingkat kesamaan (similarity) antara dua buah tree[4]. Contoh penerapan algoritma ini adalah pada sistem multi agent ACORN (Agent-based Community Oriented Routing Network)[5]. ACORN arsitektur

multi agent terdistribusi yang digunakan untuk pencarian, dan manajemen informasi di dalam jaringan. ACORN menggunakan konsep informasi sebagai agent untuk melakukan routing item informasi individu di seluruh jaringan baik pengguna maupun agent, menyediakan pencarian dan menyediakan informasi ke seluruh jaringan. Agent yang terbentuk dalam sistem ini adalah agent pencari/pengguna informasi dan agent penyedia informasi.

Penentuan tingkat kesamaan (similarity) secara umum dilakukan dengan menentukan distance, atau level similaritynya. Dua buah objek dikatakan mempunyai tingkat kesamaan yang tinggi, ditandai dengan nilai distance yang mendekati nol atau nilai similarity yang mendekati angka 1.

Perhitungan tingkat kesamaan menggunakan metoda distance mudah diterapkan untuk atribut-atribut object yang berbentuk numeric. Contoh penentuan similarity yang menggunakan distance adalah metoda penghitungan Euclidan Distance dan Cosine Angle Distance.

Sedangkan metode yang dipakai untuk menghitung similarity antar object yang memiliki atribut non numeric antara lain: metoda himpunan, structure domain hirarki dan extended weighted tree similarity.

Pada penelitian ini metoda yang digunakan untuk menentukan tingkat similarity antar 2 buah tree adalah algoritma extended weighted tree similarity dimana tingkat kesamaan dilambangkan dengan bilangan real antara 0 dan 1. similarity bernilai 0 apabila keduanya sama sekali tidak memiliki kesamaan dan bernilai 1 apabila benar-benar memiliki kesamaan. Sedangkan bentuk tree yang dapat diproses oleh algoritma ini adalah mengacu pada model Weighted Object-Oriented RuleML.

Algoritma Extended Weighted Tree Similarity memiliki 3 fungsi utama yaitu treesim, treemap dan treeplicity. Sedangkan structure pemanggilan fungsi (function call) terdiri atas parameter yang diletakkan dalam kurung siku "[]" dan argument yang diletakkan dalam tanda kurung "()"

Fungsi treesim diimplementasikan dengan penulisan "treesim[N,A](t,t)" yang akan menghasilkan bilangan real [0,1]. Keterangan detail fungsi ini sebagai berikut:

- o Parameter "N" adalah node-identity fraction, berupa bilangan antara 0 dan 1, yang menentukan besar fraksi yang diberikan atas kesamaan label pada root, relative terhadap hasil perhitungan perbandingan list subtree-nya.
- o Parameter "A" adalah Arc Function untuk melakukan adjustment pada hasil similarity, mengkompensasikan degradasi similarity untuk nested trees.
- o Argument "t" dan "t'" adalah dua buah tree yang akan dicari similaritynya.

Fungsi treemap yang diimplementasikan dengan cara penulisan "treemap[N,A](l,l')" berfungsi untuk membandingkan dua list l dan l' secara recursif, yang masing-masing merupakan himpunan arc pada setiap level yang rootnya identik sesuai identifikasi dari treesim. Fungsi ini menghasilkan bilangan real [0,1].

Fungsi treeplicity(I,t) bersifat recursif untuk mengukur tingkat kesederhanaan (simplicity) suatu tree, yang bernilai antara 0 sampai 1. jika nilai mendekati 1 berarti bahwa tree tersebut semakin sederhana, dan jika nilainya mendekati 0 maka tree tersebut semakin kompleks. Nilai ini akan semakin menurun dengan semakin banyaknya arc dan semakin dalam level (breadth dan depth). Nilai sebagai argument dari fungsi ini adalah depth degradation value yang diawali dengan harga (1-N) oleh fungsi treemap. Setiap ada penambahan level, maka I akan dikalikan dengan sebuah factor global treeplideg yang nilainya ≤ 0.5 . fungsi ini juga menghasilkan bilangan real [0,1].

4. PENGGUNAAN ALGORITMA EXTENDED WEIGHTED TREE SIMILARITY.

Rekonstruksi metoda mesin pencarian pada perpustakaan digital menggunakan algoritma extended weighted tree similarity ini memiliki 3 tahapan, tahap pertama adalah penyediaan data yang dapat direpresentasikan dalam bentuk tree seperti gambar 2.1, dan untuk menjaga konsistensinya bentuk tree ini mengacu pada model Wiegthed Object Oriented RuleML seperti contoh pada gambar 2.2, proses ini biasa disebut provider agent.

Data pada provider agent ini didapat dari proses pemasukan data melalui interface yang khusus disediakan untuk tujuan ini. Selanjutnya data-data ini disimpan dalam media penyimpanan seperti database. Data ini kemudian akan selalu di jaga untuk dapat selalu up to date, sesuai perkembangan jumlah literatur yang didapat.

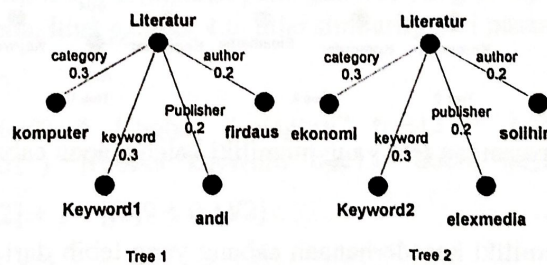
Tahap kedua adalah penangkapan informasi untuk kebutuhan pencocokan dari user. Penangkapan informasi umumnya dilakukan dengan memberikan interface yang dapat diisi oleh user sesuai keinginannya. Agar informasi yang diberikan user ini dapat diproses oleh system ini maka sedapat mungkin user diberikan atau diarahkan untuk memberikan informasi dengan tepat. Proses ini disebut requester agent.

Jika dua tahap diatas telah tersedia maka, tahap ketiga adalah tahapan dimana dilakukan pencocokan antara informasi dari user / requester agent dengan informasi yang telah tersedia pada provider agent. Pada penelitian ini proses pencocokan ini menggunakan Algoritma Extended Weighted Tree Similarity, yaitu mengukur tingkat kesamaan (similarity) antara 2 buah tree. Tingkat kesamaan ini lambangkan dengan bilangan real antara 0 dan 1, dimana similarity akan bernilai 0 jika kedua buah representasi tree tidak memiliki kesamaan sama sekali dan bernilai 1 jika keduanya secara tetap memiliki kesamaan.

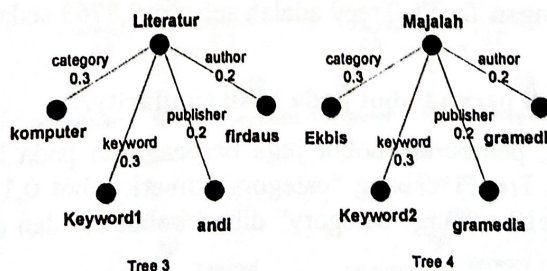
Beberapa studi kasus untuk membuktikan ketepatan pemakaian algoritma extended weighted tree similarity ini untuk menyelesaikan permasalahan yang ada pada proses pencarian literatur, yaitu antara lain:

4.1. Pembuktian Similarity antara dua tree yang memiliki perbedaan pada label root dan/atau label node.

Pada kondisi nyata proses pencocokan tidak selalu akan menemukan kesamaan baik pada label root maupun pada label node. Contoh pertama ini adalah pasangan tree yang memiliki label root sama tetapi label node berbeda serta pasangan tree yang memiliki label root dan label node berbeda. Lihat gambar dibawah ini.



Gambar 4.1. Pasangan tree yang memiliki label root sama tetapi label node berbeda.



Gambar 4.2. Pasangan tree yang memiliki label root dan label node berbeda.

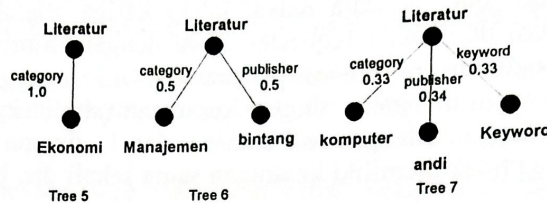
Pada kedua contoh diatas label node kedua pasangan tree itu berbeda sehingga similarity kedua pasangan tree ini berdasarkan label node adalah 0. akan tetapi jika perhitungan nilai similarity juga

melibatkan unsur label root maka secara logika pasangan tree pada gambar 4.1 masih memiliki kesamaan dibandingkan dengan pasangan tree pada gambar 4.2.

Pada kasus ini Algoritma Extended Weighted Tree similarity memberikan nilai *Node-Identity Fraction (N)* untuk pasangan tree pertama sebesar 0,1 dan nilai N untuk pasangan tree kedua sebesar 0.

4.2. Pembuktian pengaruh kelengkapan cabang pada hasil similarity.

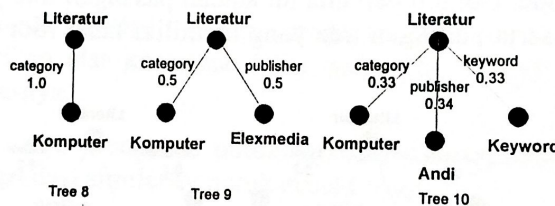
Pada kondisi dimana pasangan tree seperti pada gambar 4.3 yang memiliki kelengkapan cabang berbeda dan mengingat adanya persamaan dan juga perbedaan label maka secara logika nilai similarity pasangan Tree5-Tree6 seharusnya lebih tinggi daripada similarity pasangan Tree6-Tree7.



Gambar 4.3. pasangan tree yang memiliki kelengkapan cabang yang berbeda.

Algoritma Extended Weighted Tree memberikan jawaban yang sesuai dengan penalaran tersebut dimana Pasangan Tree5-Tree6 nilai similaritynya adalah 0,2350 lebih tinggi dari pada nilai similarity pasangan Tree6-Tree7.

Untuk kasus pada gambar 4.4, baik pasangan Tree8-Tree9 maupun pasangan Tree8-Tree10, masing-masing memiliki cabang "category" dengan label sama tetapi berbeda pada pemberian bobot, dimana nilai bobot cabang "category" untuk pasangan Tree8-Tree9 lebih besar dari pada pasangan Tree8-Tree10, sehingga secara logika nilai similarity pasangan Tree8-Tree9 seharusnya lebih tinggi.



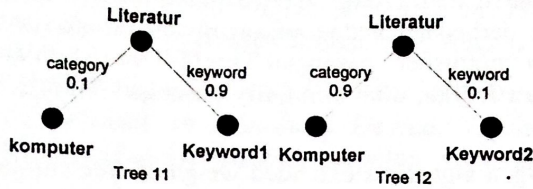
Gambar 4.4. pasangan tree yang memiliki kelengkapan cabang yang berbeda.

Selain itu Tree9 memiliki kesederhanaan cabang yang lebih dari pada Tree10 serta mengingat bahwa cabang-cabang lain selain "category" tidak ada yang sama labelnya, maka secara logika nilai similarity pasangan Tree8-Tree9 seharusnya memiliki nilai similarity yang lebih tinggi.

Kedua logika diatas dapat dijawab dengan baik oleh algoritma extended weighted tree, dimana nilai similarity untuk pasangan Tree8-Tree9 adalah sebesar 0,8763 sedangkan pasangan Tree8-Tree10 adalah 0,8350.

4.3. Pembuktian pengaruh peran bobot pada hasil similarity.

Pada kasus tertentu, pemberian bobot juga berpengaruh pada hasil similarity, seperti contoh pada gambar 4.5, dimana Tree11 cabang "category" diberi bobot 0,1 dan cabang "keyword" diberi bobot 0,9, sebaliknya Tree12 cabang "category" diberi bobot 0,9 dan cabang "keyword" diberi bobot 0,1.



Gambar 4.5. pasangan tree yang memiliki bobot yang berbeda.

Similarity kedua label pada cabang "category" adalah 1 karena keduanya sama, sedangkan cabang "keyword" yang punya label berbeda adalah 0. Total Nilai similarit adalah nilai similarity masing-masing node di kalikan dengan rata-rata bobotnya, contoh perhitungannya seperti dibawah ini:

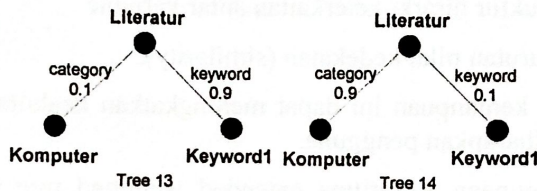
$$\text{Similarity}(\text{tree11}, \text{tree12}) =$$

$$\text{sim}(\text{"Komputer"}, \text{"Komputer"}) * [(\text{bobot "category" tree11} + \text{bobot "category" tree12})/2] + \text{sim}(\text{"Keyword1"}, \text{"Keyword2"}) * [(\text{bobot "keyword" tree11} + \text{bobot "keyword" tree12})/2]$$

$$= 1 * [(0,1 + 0,9)/2] + 0 * [(0,9 + 0,1)/2]$$

$$= 1 * 0,5 + 0 * 0,5$$

$$= 0,5$$



Gambar 4.6. pasangan tree dengan label node sama tetapi bobot berbeda.

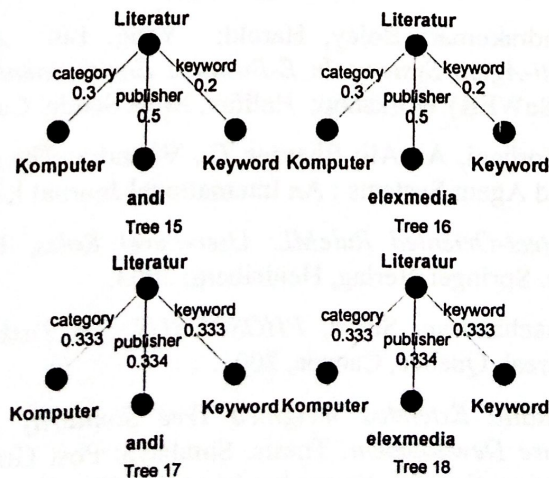
Contoh selanjutnya ini memperlihatkan pasangan tree yang setiap label node pasangannya sama sedangkan bobotnya berbeda, lihat gambar 4.6. nilai similarity dari pasangan tree13-tree14 adalah:

$$\text{Similarity}(\text{tree13}, \text{tree14}) =$$

$$\text{sim}(\text{"Komputer"}, \text{"Komputer"}) * [(\text{bobot "category" tree13} + \text{bobot "category" tree14})/2] + \text{sim}(\text{"keyword1"}, \text{"keyword1"}) * [(\text{bobot "keyword" tree13} + \text{bobot "keyword" tree14})/2]$$

$$= 1 * [(0,1 + 0,9)/2] + 1 * [(0,9 + 0,1)/2]$$

$$= 1 * 0,5 + 1 * 0,5 = 1$$



Gambar 4.7. pasangan tree yang kelengkapan cabang sama tetapi berbeda pada penentuan bobot.

Pasangan Tree15-Tree16 dan Tree17-Tree18 memiliki kesamaan disemua label node kecuali untuk cabang "publisher", perbedaan kedua pasangan ini terletak pada distribusi bobotnya dimana pada nilai bobot pada label "publisher" pasangan Tree15-Tree16 adalah 0,5 sedang pasangan Tree17-Tree18 adalah 0,333. secara logika, nilai similarity pasangan Tree15-Tree16 lebih rendah dari pada pasangan Tree17-Tree18.

Nilai secara perhitungan algoritma extended weighted tree similarity menunjukkan hasil yang sesuai dengan penalaran diatas yaitu pasangan Tree15-Tree16 nilai similaritinya adalah 0,55 sedang pasangan Tree17-Tree18 adalah 0,7

Hasil pengamatan ini menunjukkan bahwa nilai similarity dua tree yang memiliki label node persis sama menyebabkan distribusi bobot cabangnya tidak berpengaruh terhadap nilai similarity. Nilai bobot akan memiliki pengaruh jika ada perbedaan nilai minimal pada salah satu cabangnya.

5. SIMPULAN DAN SARAN.

Simpulan yang dapat diambil dari percobaan diatas, adalah bahwa Algoritma extended weighted tree similarity dapat dijadikan salah satu alternatif untuk diimplementasikan pada mesin pencarian literatur di perpustakaan digital karena memiliki beberapa kemampuan seperti:

- o Dapat mengakomodasi tingkat prioritas secara fleksible.
- o Dapat mengakomodasi ketidaklengkapan informasi yang muncul.
- o Dapat mengadaptasi struktur hirarki keterkaitan antar variable
- o Dapat disajikan sesuai urutan nilai kedekatan (similarity).

Sehingga dengan beberapa kemampuan ini dapat meningkatkan keakuratan informasi yang disajikan serta mendekati apa yang diharapkan pengguna.

Pengembangan penggunaan Algoritma extended weighted tree similarity pada perpustakaan digital ini dapat dikombinasikan dengan preproses pengambilan informasi berupa judul (title) dan kata kunci (Keyword) dari literatur berupa elektronik file yang berjalan secara otomatis menggunakan konsep-konsep information retrieval.

6. DAFTAR PUSTAKA.

1. Sarno, Riyanarto; Yang, Lu; Bhavsar, C., Virendra; Boley, Harold. *The Agent Matcher Architecture Applied to Power Grid Transactions*. IEEE/Web Intelligence Consortium International Conference on Web Intelligence, Halifax, Nova Scotia, Canada, 2003.
2. Setyawan, S.H. and Sarno, R., *Fuzzy Logics Incorporated to Extended Weighted-Tree Similarity Algorithm for Agent Matching in Virtual Market*, International Seminar on Information and Communication Technology, August 2005.
3. Bhavsar, C., Virendrakumar; Boley, Harold; Yang, Lu. *A Weighted-Tree Similarity Algorithm For Multi-Agent Systems In E-Business Environments*. Business Agents and the Semantic Web (BASEWEB) Workshop; Halifax, Nova Scotia, Canada : 2003.
4. Marsh, Stephen; Ghorbani, A., Ali; Bhavsar, C., Virendra. *The ACORN multi-agent system*. Web Intelligence and Agent Systems : An International Journal I, 2003.
5. Boley, Harold. *Object-Oriented RuleML: User-Level Roles, URI-Grounded Clauses and Order-Sorted Terms*. Springer-Verlag, Heidelberg; 2003.
6. Gil. Yolanda, Ramachandran. Surya, *PHOSPORUS : A Task-Based Agent Matchmaker*, AGENT '01, Montreal, Quebec, Canada, 2001.
7. Fabianto, Ervan., *Ratio Extended Weighted Tree Similarity Algorithm Applied to Cost Estimate of Software Development*. Thesis. Surabaya: Post Graduate Program, Faculty of Informatics Engineering, Sepuluh Nopember Institute of Technology; July 2004.

8. Budianto. And Sarno, R. *Shape Matching using Thin-Plate Splines Incorporated to Extended Weighted-tree similarity Algorithm for Agent Matching in Virtual Market*, International Seminar on Information and Communication Technology, August 2005.
9. Solihin, Firdaus. *The Application Of Extended Weighted Tree Similarity Algorithm To Provide Optimum Information in Handheld Device*, Thesis. Surabaya: Post Graduate Program, Faculty of Informatics Engineering, Sepuluh Nopember Institute of Technology; August 2006.
10. Solihin, Firdaus. And Sarno, R. *The Application Of Extended Weighted Tree Similarity Algorithm To News Website Search Tool*, International Seminar on Information and Communication Technology, August 2006.