

ANALISA PENGATURAN NILAI BOBOT LEAF NODE PADA ALGORITMA WEIGHTED TREE SIMILARITY UNTUK PENCARIAN INFORMASI LOWONGAN PEKERJAAN

Ulla Delfana Rosiani¹

Abstrak

Algoritma *Weighted Tree Similarity* dapat diterapkan pada aplikasi pencarian informasi lowongan pekerjaan karena dapat dirancang bentuk tree bagi pencari informasi lowongan pekerjaan dan tree pemberi informasi lowongan pekerjaan dengan hasil pencarian dapat menyajikan urutan kemiripan mulai dari yang paling mirip/mendekati data pencarian hingga yang paling jauh dari data pencarian, dengan hasil pencarian bernilai akurat. Algoritma ini memiliki keunikan karena memiliki representasi tree yang berbeda dengan yang lain. Tree yang dipergunakan memiliki node berlabel, cabang berlabel serta berbobot. *Leaf node* merupakan bagian yang nantinya akan dibandingkan paling dahulu. Pada penelitian ini dilakukan pengaturan bobot leaf node pada penerapan *algoritma weighted tree similarity* pada pencarian informasi lowongan pekerjaan dengan menggunakan pencocokan string dan angka antara dua tree berbobot, yaitu antara tree pencari kerja dengan tree pemberi informasi pekerjaan. Hasil perhitungan perbandingan antara kedua Tree (Tree pencari informasi lowongan pekerjaan dengan Tree pemberi informasi lowongan pekerjaan) sangat dipengaruhi oleh: Adanya data pada leafnode yang akan dibandingkan, dimana hasil perhitungan berbeda saat formulir isian pencarian diberi data. Serta pengaturan data bobot pada leaf node yang berasal dari data prioritas pencarian, dimana saat data pencarian (*leafnode*) tetap, tapi pengaturan data bobot pada leaf node diubah-ubah, hasil perhitungan akan berbeda dan urutan kemiripan ikut.

Kata-kata kunci: *Weighted Tree Similarity, Leaf Node, Algoritma load flow, losses, pembebanan*

¹*Ulla Delfana Rosiani. Program Studi Manajemen Informatika, Jurusan Teknik Elektro, Politeknik Negeri Malang*

Abstract

Weighted Tree Similarity Algorithm is unique because it has a tree representation that is different from the others. Tree nodes used has labeled, labeled and weighted branch. Leaf nodes are the parts that will be compared first, and this section has weight . This algorithm can be applied to the application search for job information because it can be designed to form job information seeker tree and job information providers tree. Search results are able to present the sequence similarity ranging from the most similar/closer to the most distant from the data search and accurately valued. In this research, setting the weight of leaf nodes in the search job information using strings and numbers matching between two weighted tree: job information seekers tree and job information providers tree. With the results, the comparison between these two trees be affected by: The existence of field data on the leaf nodes will be compared, in which the results of different calculation when a search consent form was not given the data and when the given data. As well as the arrangement of data in the leaf nodes weights derived from the data search priority, which the search data (leaf nodes) given the data, but the data arrangement on the leaf node weights changed, the results will be different calculation and sequence similarity change.

Keywords: *load flow , losses,*

1. PENDAHULUAN

Pada layanan pencarian kerja yang ada saat ini, terutama yang melalui media internet, sistem pencarian informasi pekerjaan hanya berdasarkan nama dan jenis pekerjaan yang dicarinya. Sehingga masih sering terjadi begitu pekerjaan ditemukan, namun syarat yang ditentukan oleh pihak perusahaan tidak terpenuhi. Aplikasi yang akan dibuat dalam penelitian ini akan dibuat berbasis website akan dibuat sistem pencarian berdasarkan kualifikasi detail yang dimiliki seseorang dan pekerjaan yang diinginkan dengan menggunakan *Algoritma Weighted Tree Similarity*. *Algoritma weighted tree similarity* memiliki keunikan karena memiliki representasi tree yang berbeda dengan yang lain. Tree yang dipergunakan memiliki node berlabel, cabang berlabel serta berbobot. Leaf node merupakan bagian yang nantinya akan dibandingkan paling dahulu.

Penelitian yang berhubungan *algoritma weighted tree similarity* telah banyak dilakukan. Antara lain oleh Sarno dan

Rahutomo (2008) yang melakukan penelitian untuk menggunakan *algoritma weighted tree similarity* dalam pencarian semantik dalam mencari suatu halaman website di Internet. Penelitian lain dilakukan oleh Rosiani, U.D (2010) yang melakukan penelitian menggunakan *algoritma weighted tree similarity* untuk pencarian informasi lowongan pekerjaan. Dalam penelitian ini digunakan metode *string matching* pada pencocokan *leaf node*. Hasil pencarian dapat menyajikan urutan kemiripan mulai dari yang paling mirip/mendekati data pencarian hingga yang paling jauh dari data pencarian, dengan hasil pencarian bernilai akurat.

Pada penelitian ini dilakukan pengaturan bobot *leaf node* pada penerapan *algoritma weighted tree similarity* pada pencarian informasi lowongan pekerjaan dengan menggunakan pencocokan string dan angka pada sisi leaf node antara dua tree berbobot, yaitu antara tree pencari kerja dengan tree pemberi informasi pekerjaan. Diharapkan dari penelitian ini bisa dilihat bagaimana pengaruh pengaturan bobot leaf node pada hasil pencarian yang dilakukan.

2. KAJIAN PUSTAKA

2.1 *Weighted Tree Similarity*

Algoritma weighted tree similarity adalah sebuah algoritma untuk mencari nilai keserupaan (*similarity*) dengan menggunakan bentuk tree yang berbobot. Algoritma ini memiliki keunikan karena memiliki representasi tree yang berbeda dengan yang lain. Tree yang dipergunakan memiliki node berlabel, cabang berlabel serta berbobot. Cabang yang berlabel memberikan pemahaman lebih kepada label nodenya. Begitu pula bobot cabang memungkinkan memberikan tingkat kecenderungan kepada cabang tertentu lebih dari yang lainnya.

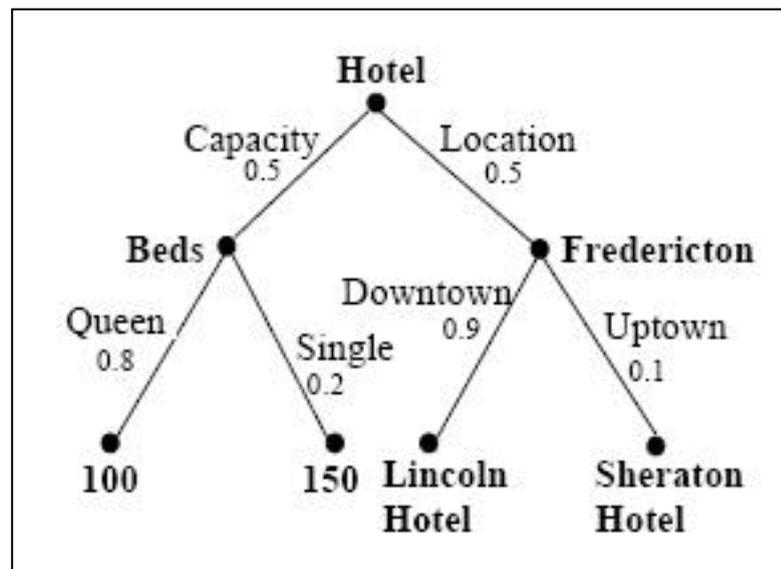
2.2.1 Representasi Tree

Hal yang paling dasar dalam penerapan algoritma *weighted tree similarity* adalah menentukan bentuk (representasi) *tree* dari kasus yang akan diselesaikan. Representasi *tree* dalam *weighted tree* mengikuti aturan sebagai berikut:

- Nodenya berlabel
- Cabang berlabel

- Cabang berbobot
- Label dan bobot ternormalkan. Label terurut sesuai abjad dari kiri ke kanan. Jumlah bobot dalam cabang seringkat subtree yang sama adalah 1.

Sangat memungkinkan representasi *Tree* dan pencocokannya memiliki bentuk yang bermacam-macam. Hal ini terjadi karena setiap kasus memiliki perbedaan terhadap kasus yang lain, sehingga *tree* akan tergantung terhadap kasus yang terjadi. Dokumen yang akan dihitung kemiripannya direpresentasikan dalam sebuah *tree* yang memiliki karakteristik node berlabel, cabang berlabel dan cabang berbobot. Contoh representasi *tree* bisa dilihat pada gambar dibawah ini:



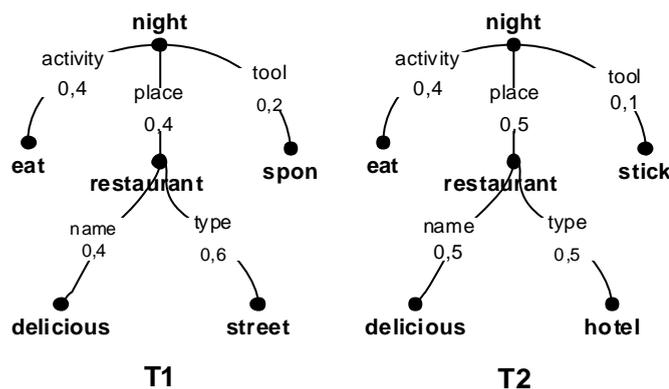
Gambar 1. Contoh Representasi *Tree*

Gambar diatas menunjukkan bentuk *tree* untuk informasi yang ada pada sebuah hotel. Dimana pada bagian node berlabel : Hotel. Pada cabang paling atas dibagi 2: Capacity dengan bobot cabang bernilai 0,5 dan Location dengan bobot cabang bernilai 0,5 juga. Beds dan Fredericton merupakan node juga dengan masing-masing memiliki cabang sendiri-sendiri dan nilai cabang masing-masing. Sedangkan 100, 150, Lincoln Hotel dan Sheraton Hotel dikenal dengan istilah *leaf node*. Pada bagian inilah yang nantinya akan dibandingkan paling dahulu. Ada banyak metode

pencocokan pada sisi leaf node, antara lain: *string matching*, *cosine matching* ataupun *fuzzy matching*.

2.2.2 Penghitung Kemiripan

Algoritma penghitungan kemiripan antara dua *weighted tree* ini terdapat di dalam makalah yang ditulis oleh Yang, Bhavsar, Boley, 2003. Gambar berikut menunjukkan contoh dua buah *tree* T1 dan T2 yang dihitung kemiripannya.



Gambar 2. Contoh Kemiripan Penghitungan Dasar

Nilai kemiripan tiap pasangan *subtree* berada diantara interval $[0,1]$. Nilai 0 bermakna berbeda sama sekali sedangkan 1 bermakna identik. Kedalaman (*depth*) dan lebar (*Breadth*) *tree* tidak dibatasi. Algoritma penghitungan kemiripan *tree* secara rekursif menjelajahi tiap pasang *tree* dari atas ke bawah mulai dari kiri ke kanan. Algoritma mulai menghitung kemiripan dari bawah ke atas ketika mencapai leaf node. Nilai kemiripan tiap pasang *subtree* di level atas dihitung berdasar kepada kemiripan *subtree* di level bawahnya.

Sewaktu penghitungan, kontribusi bobot cabang juga diperhitungkan. Bobot dirata-rata menggunakan rata-rata aritmatika $(w_i+w^p_i)/2$. Nilai rata-rata bobot sebuah cabang dikalikan dengan kemiripan S_i yang diperoleh secara rekursif. Nilai S_i pertama diperoleh berdasar kemiripan leaf node dan dapat diatur nilainya menggunakan fungsi $A(S_i)$. Pada awalnya algoritma *weighted tree similarity* hanya memberi nilai 1 bila leaf nodenya sama dan diberi nilai 0 bila berbeda. Perumusan penghitungan kemiripan *tree* ini terdapat di dalam Persamaan berikut:

$$\sum(A(s_i)(w_i + w_i')/2) \quad (1)$$

dengan $A(s_i)$ adalah nilai kemiripan leaf node, w_i dan w_i' adalah bobot pasangan arc weighted *tree*. Penilaian $A(s_i)$ analog dengan logika AND sedangkan penilaian bobot pasangan analog dengan logika OR.

2.2 Pencocokan *String* (*String Matching*)

Pengertian *string* menurut *Dictionary of Algorithms and Data Structures, National Institute of Standards and Technology (NIST)* adalah susunan dari karakter-karakter (angka, alfabet atau karakter yang lain) dan biasanya direpresentasikan sebagai struktur data *array*. *String* dapat berupa kata, frase, atau kalimat.

Pencocokan *string* merupakan bagian penting dari sebuah proses pencarian *string* (*string searching*) dalam sebuah dokumen. Hasil dari pencarian sebuah *string* dalam dokumen tergantung dari teknik atau cara pencocokan *string* yang digunakan. Pencocokan *string* (*string matching*) menurut *Dictionary of Algorithms and Data Structures, National Institute of Standards and Technology (NIST)*, diartikan sebagai sebuah permasalahan untuk menemukan pola susunan karakter *string* di dalam *string* lain atau bagian dari isi teks.

3. METODE

3.1 Pengumpulan Data

Berdasarkan hasil pengumpulan data pada penelitian sebelumnya, dimana pengambilan data diambil dari hasil survey didapatkan hasil sebagai berikut:

1. Pada media cetak
 - Jenis Media: Koran JawaPost, Koran Kompas, Koran Surya
 - Model penyajian: berupa informasi dari suatu perusahaan yang berisi syarat yang dibutuhkan, pekerjaan yang dibutuhkan, nama perusahaan, batas waktu pengiriman.
2. Pada media elektrik
 - Jenis Media: Website (www.jobsdb.com, www.karir.com, www.bekas.com)

- Model penyajian : sama dengan media cetak, khusus pada media website adanya fasilitas pencarian. Fasilitas pencarian ini terbatas pada pencarian berdasarkan: nama pekerjaan, nama perusahaan dan kota dimana perusahaan tersebut berada.

Sehingga aplikasi pencarian yang dibuat ini adalah aplikasi pencarian dengan banyak *keyword* dan pemberian nilai prioritas pencarian. *Keyword* pencarian yang akan direncanakan mengacu dari data informasi lowongan pekerjaan yang diisikan oleh pemberi informasi lowongan pekerjaan. Adapun kelompok *keyword* (kata kunci) yang akan digunakan dalam pencarian, yaitu:

- Data Perusahaan:
 - Nama Perusahaan
 - Kota tempat bekerja
 - Posisi Pekerjaan
- Data Diri
 - Usia
 - Jenis kelamin
 - Pendidikan terakhir
 - Jurusan
 - Nilai
 - Status pernikahan
 - Pengalaman

3.2 Penentuan Bentuk *Tree*

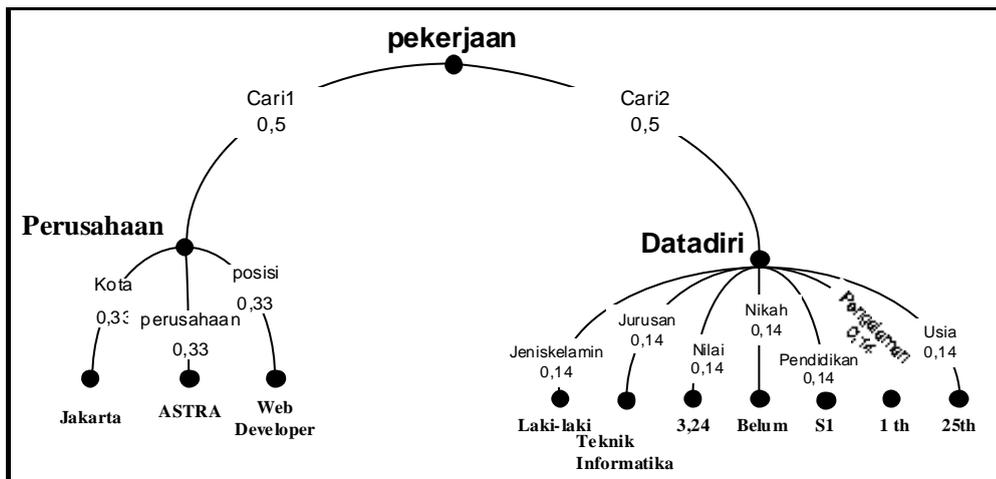
Penggunaan algoritma *Weighted Tree Similarity* dalam pencarian informasi lowongan pekerjaan ini diawali dengan pendefinisian bentuk *tree* baik bagi pemberi informasi lowongan pekerjaan maupun bagi pencari informasi lowongan pekerjaan, yang merupakan dasar dalam proses pencocokan dan pencarian data. Data pencari informasi lowongan informasi pekerjaan akan disimpan dalam bentuk (representasi) *tree* sebagai berikut:

3.2.1 *Tree* Bagi Pemberi Informasi Lowongan Pekerjaan

Bentuk *Tree* bagi pemberi informasi lowongan pekerjaan direncanakan akan memiliki dua level node. Dengan ketentuan sebagai berikut:

- o Node level 1 : **pekerjaan**, dengan 2 cabang:
 - o Nama Cabang Node (arc-Node), ada 2 :
 - **Cari1**
 - **Cari2**
 - o Bobot Cabang Node (arc-weighted) untuk masing-masing cabang:
 - Cari1 : **0,5**
 - Cari2 : **0,5**
- o Node level 2, ada 2 :
 - o **Perusahaan**, ada 3 cabang
 - Nama Cabang Node (arc-Node):
 - **Perusahaan** → nama perusahaan
 - **Kota** → kota tempat perusahaan berada
 - **posisi** → posisi lowongan pekerjaan
 - Bobot Cabang Node (arc-weighted) untuk masing-masing cabang bernilai sama: $1/3 = 0,33$
 - o **Data Diri**, ada 7 cabang
 - Nama Cabang Node (arc-Node) :
 - **Jenis kelamin**
 - **Jurusan** → Jurusan/fakultas pada pendidikan terakhir
 - **Nilai** → Nilai kelulusan
 - **Nikah** → Status pernikahan
 - **Pendidikan** → Pendidikan terakhir
 - **Pengalaman** → Lama pengalaman
 - **Usia**
 - Bobot Cabang Node (*arc-weighted*) untuk masing-masing cabang bernilai sama : $1/7=0,14$. Bagian inilah yang disebut dengan bobot *leaf node*

Informasi lowongan pekerjaan yang disimpan pada database dengan data yang bermacam-macam akan direpresentasikan dalam bentuk *tree* yang sama sebagai berikut:



Gambar 3. Representari *Tree* Pemberi Informasi Pekerjaan

3.2.2. *Tree* Bagi Pencari Informasi Lowongan Pekerjaan

Bentuk *Tree* bagi pencari informasi lowongan pekerjaan sama seperti bentuk *tree* bagi pemberi informasi lowongan pekerjaan, direncanakan akan memiliki dua level node. Dengan ketentuan sebagai berikut:

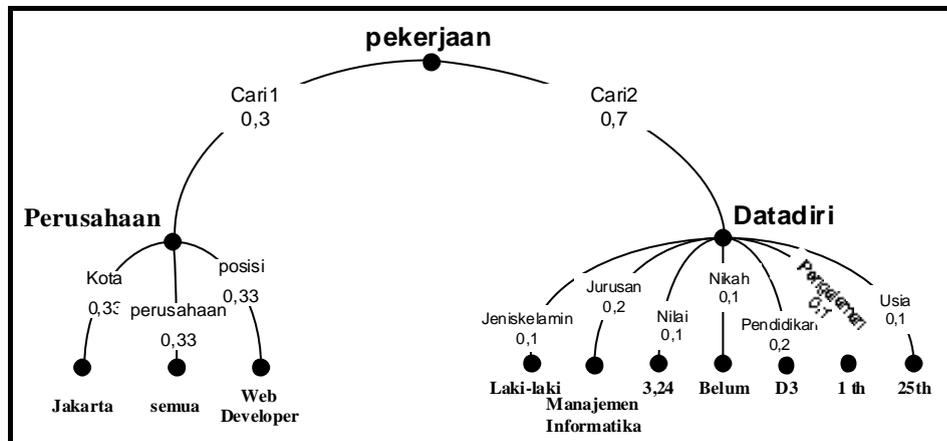
- o Node level 1 : **pekerjaan**, dengan 2 cabang:
 - Nama Cabang Node (arc-Node) 1, ada 2 :
 - **Cari1**
 - **Cari2**
 - Bobot Cabang Node (arc-weighted) untuk masing-masing cabang:
 - Cari1 : **0,3**
 - Cari2 : **0,7**

Penentuan angka bobot berdasarkan hasil pengujian pada bab pengujian, dimana jika menginginkan nilai bobot node data diri lebih dominan, maka nilai bobot harus lebih besar daripada nilai bobot pada node perusahaan. Dasar pemberian bobot ini adalah mengedepankan data diri dari pencari informasi lowongan pekerjaan, sehingga pencarian bisa mengarah pada hasil pencarian yang lebih sesuai dengan data diri bukan berdasarkan perusahaan yang akan dicari. Artinya jika selama ini perusahaan yang dijadikan obyek pencarian, pada pencarian ini justru data diri yang akan dijadikan acuan utama dalam pencarian yang paling sesuai.

- o Node level 2, ada 2 :

- o **Perusahaan**, ada 3 cabang
 - Nama Cabang Node (arc-Node):
 - **Perusahaan** → nama perusahaan
 - **Kota** → kota tempat perusahaan berada
 - **posisi** → posisi lowongan pekerjaanIsian node/*leaf node* pada cabang ini ditentukan berdasarkan data pilihan pada form yang disediakan.
 - Bobot Cabang Node (arc-weighted) untuk masing-masing cabang: Nilai bobot tiap cabang ditentukan berdasarkan ketentuan pilihan yang dilakukan oleh pencari informasi lowongan pekerjaan.
- o **Data Diri**, ada 7 cabang
 - Nama Cabang Node (arc-Node):
 - **Jenis kelamin**
 - **Jurusan** → Jurusan/fakultas pada pendidikan terakhir
 - **Nilai** → Nilai kelulusan
 - **Nikah** → Status pernikahan
 - **Pendidikan** → Pendidikan terakhir
 - **Pengalaman** → Lama pengalaman
 - **Usia**Isian node/*leaf node* pada cabang ini ditentukan berdasarkan data pilihan pada form yang disediakan.
 - Bobot Cabang Node (arc-weighted) untuk masing-masing cabang: Nilai bobot tiap cabang ditentukan berdasarkan ketentuan pilihan yang dilakukan oleh pencari informasi lowongan pekerjaan. Pada bagian inilah pengaturan bobot *Leaf node* yang akan dibandingkan.

Representasi *tree* bagi pencari informasi lowongan pekerjaan akan memiliki bentuk yang sama dengan *tree* pemberi informasi lowongan pekerjaan sebagai berikut:



Gambar 4. Representasi *Tree* Pencari Informasi Pekerjaan

4. HASIL DAN PEMBAHASAN

Pada tahapan pengujian validasi tujuan yang akan dicapai adalah apakah program telah sesuai dengan fungsi yang dituju, misalnya dalam hal ini apakah program dapat membantu mencarikan informasi lowongan pekerjaan sesuai kebutuhan. Adapun pengujian validasi yang akan dilakukan adalah Pengujian fungsi algoritma *Weighted Tree Similarity* pada aplikasi:

- A. Tanpa pengaturan Nilai Prioritas Pencarian
- B. Dengan pengaturan Nilai Prioritas Pencarian

4.1 Pencarian Dengan Data Isian Dan Prioritas Pencarian Tidak Diatur

Bahan Pengujian

1. Data pengujian : 25 Data lowongan pekerjaan
2. Tanggal pengujian : 29 Juli 2013
3. Data pencarian : diisi sebagai berikut:

The image shows a web form titled "Job Seeker agregator". It is divided into two main sections: "Data Diri" (Personal Data) and "Pencarian" (Search).
The "Data Diri" section contains the following fields:

- Jenis Kelamin: Pria (Male)
- Uraian: 1
- Tanggal Lahir: 1/1/1985
- Pendidikan: D4/S1
- Jurusan: AKUNTANSI
- Nilai Terakhir: 3,00
- Status Pernikahan: Belum Nikah
- Pengalaman Kerja: Fresh Graduate

The "Pencarian" section contains the following fields:

- Nama Perusahaan: Semua Perusahaan
- Kota: Semua Kota
- Posisi: Semua Posisi
- Metode: WIFI NONWIFI

At the bottom of the search section, there are two buttons: "Cari" (Search) and "Reset".

Gambar 5. Pengaturan Data Pencarian

Hasil Pengujian:

1. Pengujian dilakukan pengaturan urutan prioritas: **PRIORITAS PENCARIAN TIDAK DIATUR** sehingga nilai bobot *Leaf Node* adalah sama: 0,14 (=1:7)
2. Pencarian dilakukan dengan hasil sebagai berikut:
 - a. Urutan hasil pencarian mulai dari nilai keserupaan paling tinggi hingga yang paling rendah diantara kedua *Tree* adalah sebagai berikut: **Tree ke-4-2-8-23-3-5-6-9-10-11-12-15-17-18-19-21-22-24-25-1-7-13-14-16-20**
 - b. Sementara log pencarian ditampilkan sebagai berikut:

```
[*]-----NEW SESSION-----[*]
rata2 value/node lv11 : 0.8
rata2 value/node lv12 cabang ke-1 : 0.88
rata2 value/node lv12 cabang ke-2 : 0.14
init value: (0.88) => value lv12 cabang ke-1 :
0.88,0.88,0.88,
init value: (0.88) => value lv12 cabang ke-2 :
0.14,0.14,0.14,0.14,0.14,0.14,0.14,
jum: (0.88) => value penyetaraan lv12 cabang ke-1 :
0.88,0.88,0.88,
jum: (0.88) => value penyetaraan lv12 cabang ke-2 :
0.14,0.14,0.14,0.14,0.14,0.14,
Pencarian Pencarian cabang: 1,2,3,4,5,6,7
jum : (0.88) => value disesuaikan dg pointer lv12 cabang ke-1
: 0.88,0.88,0.88,
jum : (0.88) => value disesuaikan dg pointer lv12 cabang ke-2
: 0.14,0.14,0.14,0.14,0.14,0.14,
hasil: *** result pencari terhadap tree ke-1: 0.782
hasil: *** result pencari terhadap tree ke-2: 0.8
hasil: *** result pencari terhadap tree ke-3: 0.816
hasil: *** result pencari terhadap tree ke-4: 0.884
hasil: *** result pencari terhadap tree ke-5: 0.816
hasil: *** result pencari terhadap tree ke-6: 0.816
hasil: *** result pencari terhadap tree ke-7: 0.782
hasil: *** result pencari terhadap tree ke-8: 0.8
hasil: *** result pencari terhadap tree ke-9: 0.816
hasil: *** result pencari terhadap tree ke-10: 0.816
hasil: *** result pencari terhadap tree ke-11: 0.816
hasil: *** result pencari terhadap tree ke-12: 0.816
hasil: *** result pencari terhadap tree ke-13: 0.782
hasil: *** result pencari terhadap tree ke-14: 0.782
hasil: *** result pencari terhadap tree ke-15: 0.816
hasil: *** result pencari terhadap tree ke-16: 0.782
hasil: *** result pencari terhadap tree ke-17: 0.816
hasil: *** result pencari terhadap tree ke-18: 0.816
hasil: *** result pencari terhadap tree ke-19: 0.816
hasil: *** result pencari terhadap tree ke-20: 0.782
hasil: *** result pencari terhadap tree ke-21: 0.816
hasil: *** result pencari terhadap tree ke-22: 0.816
hasil: *** result pencari terhadap tree ke-23: 0.8
hasil: *** result pencari terhadap tree ke-24: 0.816
hasil: *** result pencari terhadap tree ke-25: 0.816
Pengurutan value :
0.884,0.8,0.8,0.8,0.816,0.816,0.816,0.816,0.816,0.816,0.816,
0.816,0.816,0.816,0.816,0.816,0.816,0.816,0.816,0.816,0.782,0.782,
0.782,0.782,0.782,0.782,
```

Gambar 6. Log Pencarian Sub Pengujian 1

4.2 Pencarian Data Isian Dengan Pengaturan Prioritas Pencarian

Bahan Pengujian

1. Data pengujian : 25 Data lowongan pekerjaan
2. Tanggal pengujian : 29 Juli 2013
3. Data pencarian : **Kombinasi 1**, diisi sebagai berikut:

The image shows a web form titled "Job Seeker agregator". It is divided into two main sections: "Data Diri" and "Perusahaan".

Data Diri:

- Jenis Kelamin: Pria (dropdown)
- Tgl Lahir: 1/1/1985 (date picker)
- Pendidikan: S1 (dropdown)
- Jurusan: AKUNTANSI (dropdown)
- Nilai Ujian: 3,000 (text input)
- Status Pernikahan: Belum Nikah (dropdown)
- Pengalaman Kerja: Fresh Graduate (dropdown)

Perusahaan:

- Nama Perusahaan: --Semua Perusahaan-- (dropdown)
- Kota: --Semua Kota-- (dropdown)
- Postal: --Semua Postal-- (dropdown)
- Metode: WJIB (radio button)

Buttons: Lupa, Reset

Gambar 7. Pengaturan Data Pencarian

Hasil pengujian:

1. Proses pengurutan nilai prioritas pencarian berdasarkan isian prioritas pencarian, didapatkan :
 - a. Untuk level 2 cabang 1 tidak ada pengaturan sehingga dianggap urutan seperti normal yaitu cabang ke-1,2,3
 - b. Untuk level 2 cabang 2 ada pengaturan prioritas dengan urutan dari nomer 1 hingga 7 adalah cabang ke-3,6,1,2,4,7,5.
2. Proses penyesuaian nilai prioritas menjadi nilai bobot Berdasarkan pengurutan diatas, pointer akan mengarahkan nilai bobot pada cabang yang telah diurutkan, sehingga nilai bobot pada masing-masing cabang akan didapatkan sebagai berikut:
 1. Untuk level 2 cabang 1 karena tidak ada pengaturan maka nilai bobot d semua cabang adalah sama yaitu 0,325
 2. Untuk level 2 cabang 2 karena adanya pengaturan prioritas maka nilai bobot *Leaf node* mulai dari cabang ke 1 hingga ke 7 adalah: 0,16;0,1;0,2;0,18;0,14;0,08;0,12
3. Proses pencarian dilakukan dengan hasil sebagai berikut:
 - a. Urutan hasil pencarian mulai dari nilai keserupaan paling tinggi hingga yang paling

rendah diantara kedua *Tree* adalah sebagai berikut: **Tree ke 4-8-2-23-9-10-17-18-15-19-24-25-5-3-21-22-6-11-12-13-20-7-14-16-1**

b. Sementara log pencarian ditampilkan sebagai berikut:

```
[*]-----NEW SESSION-----[*]
rata2 value/node lv12 : 0.9
rata2 value/node lv12 cabang ke-1 : 0.99
rata2 value/node lv12 cabang ke-2 : 0.14
init value: (0.99) => value lv12 cabang ke-1 :
0.99,0.99,0.91.
init value: (0.99) => value lv12 cabang ke-2 :
0.2,0.19,0.16,0.14,0.12,0.1,0.09.
sum: (1) => value penyetaraan lv12 cabang ke-1 :
0.99,0.928,0.928.
sum: (0.99) => value penyetaraan lv12 cabang ke-2 :
0.2,0.19,0.16,0.14,0.12,0.1,0.09.
Pointer Prioritas cabang: 4,1,2,3,6,4,1,5,7,2,6,
sum: (0.99) => value disesuaikan dg pointer lv12 cabang ke-1 :
0.928,0.928,0.928.
sum: (0.99) => value disesuaikan dg pointer lv12 cabang ke-2
0.16,0.1,0.2,0.19,0.14,0.09,0.14
*** result pencari terhadap tree ke-1: 0.699
*** result pencari terhadap tree ke-2: 0.807
*** result pencari terhadap tree ke-3: 0.795
*** result pencari terhadap tree ke-4: 0.981
*** result pencari terhadap tree ke-5: 0.795
*** result pencari terhadap tree ke-6: 0.783
*** result pencari terhadap tree ke-7: 0.705
*** result pencari terhadap tree ke-8: 0.808
*** result pencari terhadap tree ke-9: 0.819
*** result pencari terhadap tree ke-10: 0.819
*** result pencari terhadap tree ke-11: 0.783
*** result pencari terhadap tree ke-12: 0.783
*** result pencari terhadap tree ke-13: 0.717
*** result pencari terhadap tree ke-14: 0.703
*** result pencari terhadap tree ke-15: 0.807
*** result pencari terhadap tree ke-16: 0.705
*** result pencari terhadap tree ke-17: 0.819
*** result pencari terhadap tree ke-18: 0.819
*** result pencari terhadap tree ke-19: 0.807
*** result pencari terhadap tree ke-20: 0.717
*** result pencari terhadap tree ke-21: 0.789
*** result pencari terhadap tree ke-22: 0.789
*** result pencari terhadap tree ke-23: 0.897
*** result pencari terhadap tree ke-24: 0.801
*** result pencari terhadap tree ke-25: 0.801
Pengaturan value :
0.921,0.909,0.897,0.897,0.819,0.819,0.819,0.819,0.807,0.807,
0.801,0.801,0.793,0.793,0.789,0.789,0.783,0.783,0.783,0.717,
0.717,0.705,0.705,0.705,0.699.
```

Gambar 8. Log Pencarian Sub Pengujian 2

4.3 Pencarian dengan data isian dan pengaturan prioritas pencarian (kombinasi 2)

Bahan Pengujian

1. Data pengujian : 25 Data lowongan pekerjaan (lampiran 1)
2. Tanggal pengujian : 29 Juli 2013
3. Data pencarian : **Kombinasi 2**, diisi sebagai berikut:

Data Diri	
Jenis Kelamin	Pria
Tgl Lahir	1/1/1985
Pendidikan	D4/S1
Jurusan	AKUNTANSI
Nilai Terakhir	3,00
Status Pernikahan	Belum Nikah
Pengalaman Kerja	Fresh Graduate

Penyesuaian	
Nama Perusahaan	Semua Perusahaan
Kota	--Semua Kota--
Posisi	--Semua Posisi--
Media Sosial	WhatsApp: WIST Telegram: NUNWIST

Cari Reset

Gambar 9. Pengaturan Data Pencarian

Hasil pengujian:

1. Proses pengurutan nilai prioritas pencarian berdasarkan isian prioritas pencarian, didapatkan :
 - a. Untuk level 2 cabang 1 tidak ada pengaturan sehingga dianggap urutan seperti normal yaitu cabang ke-1,2,3
 - b. Untuk level 2 cabang 2 ada pengaturan prioritas dengan urutan dari nomer 1 hingga 7 adalah cabang ke-1,2,3,4,5,6,7
2. Proses penyesuaian nilai prioritas menjadi nilai bobot berdasarkan pengurutan diatas, pointer akan mengarahkan nilai bobot pada cabang yang telah diurutkan, sehingga nilai bobot pada masing-masing cabang akan didapatkan sebagai berikut:
 - a. Untuk level 2 cabang 1 karena tidak ada pengaturan maka nilai bobot d semua cabang adalah sama yaitu 0,325
 - b. Untuk level 2 cabang 2 karena adanya pengaturan prioritas maka nilai bobot *Leaf Node* mulai dari cabang ke 1 hingga ke 7 adalah:
0,2;0,18;0,16;0,14;0,12;0,10;0,08
3. Proses pencarian dilakukan dengan hasil sebagai berikut:

- a. Urutan hasil pencarian mulai dari nilai keserupaan paling tinggi hingga yang paling rendah diantara kedua *Tree* adalah sebagai berikut: *Tree* ke **4-8-2-23-15-19-24-25-9-10-17-18-6-11-12-3-5-21-22-7-14-16-1-13-20**
- b. Sementara log pencarian ditampilkan sebagai berikut:

```
[*]-----NEW SESSION-----[*]
kata2 vluce/node lvl1 : 0.5
kata2 vluce/node lvl2 cabang ke-1 : 0.32
kata2 vluce/node lvl2 cabang ke-2 : 0.14
init value: (0.66) => value lvl2 cabang ke-1 :
0.25,0.22,0.21,
init value: (0.68) => value lvl2 cabang ke-2 :
0.2,0.18,0.16,0.14,0.12,0.1,0.08,
jum: (1) => value penyortiran lvl2 cabang ke-1 :
0.25,0.225,0.225,
jum: (0.90) => value penyortiran lvl2 cabang ke-2 :
0.2,0.18,0.16,0.14,0.12,0.1,0.08,
Pemula Pemula cabang: 1,1,2,2,3,3,4,4,5
jum: (0.975) => value disesuaikan dg pointer lvl2 cabang ke-
1 : 0.325,0.325,0.325,
jum: (0.98) => value disesuaikan dg pointer lvl2 cabang ke-2
0.2,0.18,0.16,0.14,0.12,0.1,0.08
*** result pencari terhadap tree ke-1: 0.729
*** result pencari terhadap tree ke-2: 0.903
*** result pencari terhadap tree ke-3: 0.795
*** result pencari terhadap tree ke-4: 0.981
*** result pencari terhadap tree ke-5: 0.795
*** result pencari terhadap tree ke-6: 0.807
*** result pencari terhadap tree ke-7: 0.741
*** result pencari terhadap tree ke-8: 0.919
*** result pencari terhadap tree ke-9: 0.813
*** result pencari terhadap tree ke-10: 0.813
*** result pencari terhadap tree ke-11: 0.807
*** result pencari terhadap tree ke-12: 0.807
*** result pencari terhadap tree ke-13: 0.692
*** result pencari terhadap tree ke-14: 0.741
*** result pencari terhadap tree ke-15: 0.831
*** result pencari terhadap tree ke-16: 0.741
*** result pencari terhadap tree ke-17: 0.813
*** result pencari terhadap tree ke-18: 0.813
*** result pencari terhadap tree ke-19: 0.831
*** result pencari terhadap tree ke-20: 0.692
*** result pencari terhadap tree ke-21: 0.789
*** result pencari terhadap tree ke-22: 0.789
*** result pencari terhadap tree ke-23: 0.903
*** result pencari terhadap tree ke-24: 0.819
*** result pencari terhadap tree ke-25: 0.819
Pengurutan value :
0.981,0.919,0.903,0.903,0.831,0.831,0.819,0.819,0.813,0.813,
0.813,0.813,0.807,0.807,0.807,0.795,0.795,0.789,0.789,0.741,
0.741,0.741,0.729,0.692,0.692,
```

Gambar 10. Log Pencarian Sub Pengujian 3

Tabel 2. Hasil Pengujian Pengaturan *Leafnode* Dan Bobot Yang Dinamis

PENGATURAN		URUTAN HASIL PERHITUNGAN																									
NO	Data Pencarian	Data Prioritas	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
	(<i>leafnode</i>)	(<i>bobot cabang</i>)	(tree ke-)																								
1	Tidak ada isian	Tidak ada pengaturan	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
2	Ada isian	Tidak ada pengaturan	4	2	8	23	3	5	6	9	10	11	12	15	17	18	19	21	22	24	25	1	7	13	14	16	20
3	Ada isian	Pengaturan I	4	8	2	23	9	10	17	18	15	19	24	25	5	3	21	22	6	11	12	13	20	7	14	16	1
4	Ada isian	Pengaturan II	4	8	2	23	15	19	24	25	9	10	17	18	6	11	12	3	5	21	22	7	14	16	1	13	20

5. PENUTUP

Kesimpulan yang diambil dari penelitian ini adalah:

- 1) Isian data pencarian (data *leaf node*) mempunyai pengaruh menentukan urutan pada hasil perhitungan dengan algoritma *Weighted Tree Similarity* meskipun tidak dilakukan pengaturan prioritas atau nilai bobot.
- 2) Pengaturan nilai prioritas dapat dilakukan dan dapat diubah menjadi nilai bobot sesuai dengan pilihan tiap cabang.
- 3) Pemberian data *leaf node* akan mempengaruhi hasil pencarian, terlihat hasil perhitungannya tidak bernilai 0.
- 4) Hasil perhitungan dipengaruhi oleh nilai keserupaan (*string matching*) antara data *leaf node* di *Tree* pencari informasi lowongan pekerjaan dan *Tree* pemberi informasi pekerjaan dan nilai bobot cabang pada *leafnode* tersebut. Terlihat dari hasil urutan pencarian yang berbeda dengan pengujian sebelum.
- 5) Hasil perhitungan perbandingan antara kedua *Tree* (*Tree* pencari informasi lowongan pekerjaan dengan *Tree* pemberi informasi lowongan pekerjaan) sangat dipengaruhi oleh:
 - a. Data *leafnode* yang akan dibandingkan (ditunjukkan dari sub pengujian 1 dan 2) dimana hasil perhitungan berbeda saat formulir isian pencarian diberi data.
 - b. Data bobot cabang *leafnode* yang berasal dari data prioritas pencarian, seperti yang ditunjukkan pada sub pengujian 3 dan 4. Dimana saat data pencarian (*leafnode*) tetap, tapi pengaturan diubah, hasil perhitungan akan berbeda dan urutan kemiripan ikut berubah.

6. DAFTAR PUSTAKA

- Budiono dan R. Sarno. 2005. *Shape Matching using Thin-Plate Splines Incorporated to Extended Weighted tree similarity Algorithm for Agent Matching in Virtual Market*. Proceeding International Seminar on Information and Communication Technology. August 2005.
- Cahyono, D., Fadlil, J., Sumpeno, S., dan Hariadi, M. 2007. *Temu Kembali Informasi Untuk Pembangkitan Basis Pengetahuan*

Dari Teks Bebas Yang Digunakan Oleh Agen Percakapan Bahasa Alami.

- Fabianto, E., 2004, Tesis. *Estimasi Biaya Rancang Bangun Perangkat Lunak Dengan Menggunakan Algoritma Ratio Extended Weighted Tree Similarity*. Institut Teknologi Sepuluh November.
- Munir, R. 2004. Bahan Kuliah: *Strategi Algoritmik*. Institut Teknologi Bandung.
- Rahutomo, F. 2009. Tesis. *Penerapan Algoritma Weighted Tree Similarity Untuk Pencarian Semantik Wikipedia*. Institut Teknologi Sepuluh November
- Setyawan, S.H. 2004. Tesis. *Penerapan Fuzzy Logic pada Algoritma Extended Weighted Tree Similarity dalam arsitektur Agent Matcher*. Teknologi Sepuluh November
- Syaroni, M., Munir R. 2005. *Pencocokan String Berdasarkan Kemiripan Ucapan (Phonetic String Matching) dalam Bahasa Inggris*. Departemen Teknik Informatika, Fakultas Teknologi Industri, Institut Teknologi Bandung (ITB)
- Rosiani, U.D. 2010. Tesis. *Penerapan Algoritma Weighted Tree Similarity pada Pencarian Informasi Lowongan Pekerjaan*. Universitas Brawijaya Malang.