

Pencarian Lowongan Pekerjaan Berbasis Agen Berdasarkan Profil Pencari Kerja dengan Pendekatan Semantic Web Service

Teguh Susyanto*¹, Khabib Mustofa²

¹Staf pengajar Jurusan Teknik Informatika; STMIK Sinar Nusantara, Surakarta

²Staf pengajar Jurusan Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta

e-mail: *teguh@sinus.ac.id, khabib@ugm.ac.id

Abstrak

Layanan pencarian informasi pekerjaan yang ada saat ini masih memiliki banyak kelemahan dan sering gagal menyajikan informasi lowongan pekerjaan yang relevan dan sesuai kebutuhan pencari kerja. Hal ini disebabkan metode pencarian yang diterapkan pada mesin pencari masih menggunakan pencocokan berbasis sintaks dan belum adanya integrasi dengan portal penyedia informasi pekerjaan, sehingga menyulitkan para pencari kerja untuk mendapatkan informasi yang relevan. Untuk mengatasi kelemahan tersebut, diusulkan sebuah prototipe aplikasi pencarian lowongan pekerjaan dengan melibatkan web service sebagai penyedia informasi pekerjaan.

Tujuan penelitian ini adalah terbentuknya rancang bangun aplikasi pencarian informasi lowongan kerja berdasarkan personalisasi pencari kerja dengan mengkombinasikan pendekatan multi agent dan semantic web service.

Prototipe aplikasi dirancang menggunakan teknologi multi agen yang memiliki kemampuan untuk memilih, memanggil web service penyedia informasi pekerjaan dan melakukan proses matching informasi lowongan pekerjaan yang sesuai dengan profil pencari kerja secara otomatis. Proses pemilihan web service menggunakan kombinasi algoritma service matching dan Simple Additive Weighting. Algoritma semantic matching digunakan untuk menghitung nilai similaritas antara lowongan pekerjaan dengan profil pencari kerja.

Berdasarkan pengujian yang dilaksanakan dengan responden, dinyatakan prototipe aplikasi ini telah dapat memberikan rekomendasi lowongan pekerjaan yang sesuai dengan profil pencari kerja.

Kata kunci— lowongan pekerjaan, web service, multi agen, semantic matching

Abstract

Currently, job searching service still has many weaknesses and often fails to provide relevant job information that matches the needs of job seekers. This is due to the searching method applied in the search engines still uses the syntax-based matching and the lack of integration among the job service providers. Therefore it's difficult for the job seekers to get the desired information. To overcome these weaknesses, a prototype of a job vacancy searching by involving a web service as a job information provider is proposed.

This thesis is aimed to create job search based on the personalization of job seeker by combining multi agent and semantic web service approaches.

The designing of the prototype used a multiagent technology whose capability was to call job service provider and run matching process of the job vacancy appropriate with the job seeker's profile automatically. Algorithm of the service selection used service matching and Simple Additive Weighting. The similarity between the job offer and the job seeker's profile was calculated by using semantic algorithm.

Based on the testing carried out to the respondents, it's stated that this prototype has been able to give recommendation of job appropriate with the job seeker's.

Keywords—*job offer, semantic web service, multi agent, semantic matching*

1. PENDAHULUAN

Layanan pencarian lowongan pekerjaan telah banyak diterapkan dalam portal-portal bursa kerja online yang menawarkan informasi pekerjaan kepada pencari kerja [1]. Layanan pencarian lowongan pekerjaan yang ada saat ini masih memiliki banyak kelemahan dan sering gagal menyajikan informasi yang relevan dan sesuai dengan kebutuhan pencari kerja. Hal ini disebabkan karena pencarian pekerjaan hanya menggunakan susunan teks (*keyword*) yang terbatas sehingga hasil yang didapatkan sebatas informasi-informasi yang mengandung teks-teks kunci pencarian. Hal ini disebabkan metode yang diterapkan pada mesin pencari masih menggunakan pencocokan berbasis sintaks, dan belum menggunakan metode *semantic matching* [2]. Kelemahan lain, yaitu belum adanya sharing informasi portal-portal bursa kerja yang lain, sehingga lowongan pekerjaan yang didapatkan kurang sesuai dengan kebutuhan pencari kerja dan tidak mampu menyediakan alternatif pilihan pekerjaan yang lain.

Teknologi *semantic web* telah diterapkan [2,3,4] dalam sistem informasi rekrutmen untuk menyeleksi lowongan pekerjaan yang relevan bagi pencari kerja. Proses penyeleksian memanfaatkan algoritma *semantic matching* dengan mengukur derajat kemiripan antara pekerjaan yang ditawarkan dengan profil pencari kerja. Namun sistem yang ditawarkan diorientasikan hanya memberi manfaat kepada perusahaan-perusahaan perekrut tenaga kerja, sedangkan kebutuhan pencari kerja belum sepenuhnya terakomodir. Alasan inilah yang memotivasi penulis untuk mengusulkan prototipe sistem pencarian secara otomatis lowongan pekerjaan berdasarkan profil pencari kerja. Sistem yang diusulkan mirip dengan penelitian [5], namun bedanya terletak pada sumber informasi pekerjaan yang digunakan, yaitu memanfaatkan *semantic web service* untuk menghubungkan dengan penyedia lowongan pekerjaan, sehingga proses *collecting* informasi pekerjaan dapat dilakukan secara dinamis.

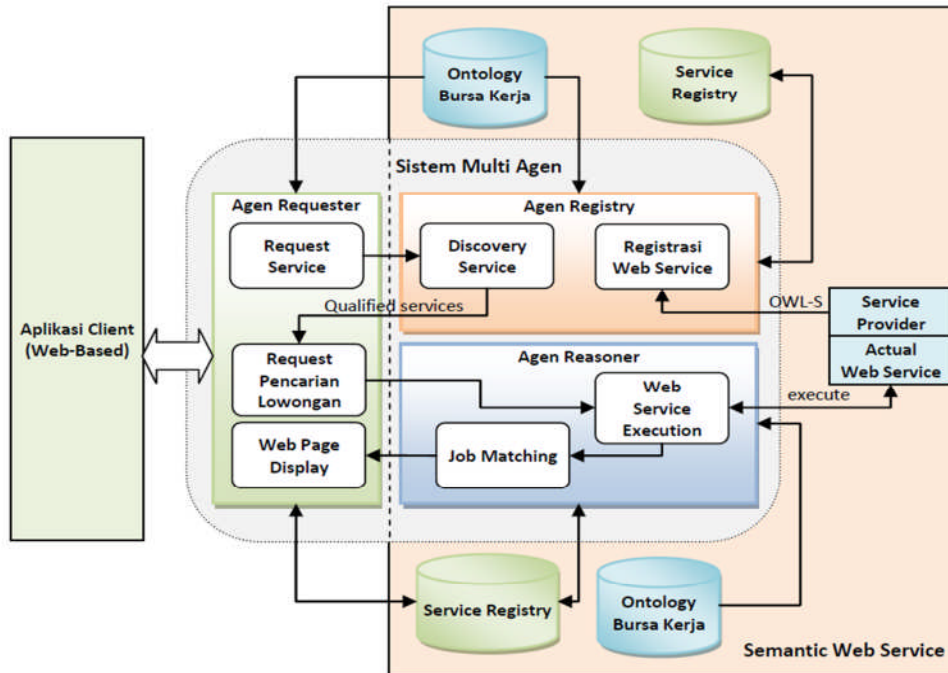
Semantic web services merupakan pengembangan dari *semantic web* dan *web service*, untuk menyusun mekanisme yang mampu dimengerti oleh, yaitu dengan mendeskripsikan *web service* dengan bahasa semantik sehingga memungkinkan proses *discovery*, *composition*, *invocation*, dan *monitoring web service* dapat dilakukan secara dinamis [6].

Kombinasi teknologi sistem agen dan *semantic web service* dalam penelitian ini ditujukan untuk memberikan kemudahan bagi pencari kerja dalam rangka menemukan lowongan pekerjaan yang sesuai pada web service penyedia informasi pekerjaan.

2. METODE PENELITIAN

2.1 Arsitektur Sistem

Arsitektur aplikasi yang dirancang berdasarkan pada lingkungan multi agen dan lingkungan *semantic web service* seperti yang ditunjukkan pada Gambar 1. Lingkungan multi agen tersusun dari agen-agen pengendali sistem untuk pencarian web service sumber informasi pekerjaan dan pengendali proses *job matching*. Secara umum agen-agen ini sebagai perantara antara pengguna dengan web service penyedia informasi pekerjaan. Lingkungan *semantic web service* tersusun atas komponen-komponen yang mengintegrasikan sistem dengan web service bursa kerja dari lingkungan eksternal.



Gambar 1 Rancangan arsitektur sistem

Secara rinci komponen-komponen utama pembentuk sistem terdiri dari:

a. *Antarmuka pengguna (Aplikasi Client Web-based)*

Aplikasi *Client Web-based* ini difungsikan sebagai antarmuka yang menjembatani antara pengguna dengan komponen-komponen aplikasi yang lain. Antarmuka ini digunakan untuk mengendalikan masukan dari pencari kerja dan menyajikan informasi hasil pemrosesan.

b. *Lingkungan multi agen*

Lingkungan multi agen terdiri tiga jenis agen, meliputi Ajen Requester, Ajen Registry dan Ajen Reasoner. Masing-masing agen memiliki peran yang berbeda dan mampu berkomunikasi dengan agen lain. Standar komunikasi agen menggunakan FIPA-ACL (*Foundation for Intelligent Physical Agents - Agent Communication Language*) dalam lingkungan *platform JADE*. Jenis agen Requester mengendalikan modul-modul proses yang berkaitan dengan antarmuka pengguna meliputi: (1) *Request Service*, yang difungsikan untuk menyusun query pencarian web service, (2) *Request Pencarian Lowongan*, yang difungsikan untuk menyusun query pencarian lowongan pekerjaan dalam format semantik, (3) *Web page display*, difungsikan untuk memberikan respon dari permintaan pengguna terkait informasi hasil pencarian web service dan pencarian lowongan untuk ditampilkan dalam halaman web.

Ajen registry mengendalikan proses yang berkaitan pengelolaan web service yang tersimpan dalam service registry, adapun modul-modul yang dikendalikan meliputi : (1) *Discovery Service*, difungsikan untuk melakukan pencarian web service berdasarkan query diberikan. Dalam proses ini dilakukan proses pencocokan web service yang tersimpan dalam service registry dengan query service. Adapun algoritma yang digunakan pada proses *discovery service* adalah *service matching* empat level. (2) *Registrasi Web Service*, difungsikan untuk mengelola registrasi web service yang ditambahkan dalam service registry.

Ajen Reasoner bertugas untuk mengendalikan modul : (1) *Web Service Execution*, yang difungsikan untuk melakukan proses pemanggilan web service untuk mendapatkan informasi pekerjaan. (2) *Job Matching*, yang berfungsi untuk melakukan proses pencocokan informasi pekerjaan dengan profil pengguna.

c. *Basis Pengetahuan (Knowledge Base)*

Lapis ini merupakan basis pengetahuan (*belief*) dari sistem yang terdiri dari *Ontologi Bursa Kerja*, *Service Registry*. *Ontologi Bursa Kerja* merupakan skema yang tersusun dari konsep-konsep (*vocabulary*) dalam domain bursa kerja yang direpresentasikan dalam bahasa OWL. Dalam ontologi bursa kerja ini memuat hubungan antar konsep secara hirarki dan relasi terdiri jabatan pekerjaan, sektor usaha, lokasi, jenjang dan jurusan pendidikan, dan pengalaman pekerjaan. *Service registry*, menyimpan kumpulan alamat *OWL-S* yang diregistrasikan, dan profil pencari kerja.

2.2 Cara Kerja Sistem

Berdasarkan modul-modul dalam rancangan arsitektur seperti Gambar 1, cara kerja sistem dijelaskan dalam dua tahap proses inti yaitu *Service Discovery* dan *Job Matching*. *Service Discovery* ini ditujukan untuk mendapatkan web service sebagai sumber informasi yang relevan bagi pengguna yang akan dimanfaatkan untuk memperoleh kumpulan informasi pekerjaan. Proses ini dibangkitkan dengan mengirimkan sebuah *query*, yang memiliki spesifikasi berupa parameter fungsional *web service* baik parameter *input* maupun *output*. Setiap parameter yang dirumuskan harus mengacu pada konsep-konsep (*vocabulary*) yang didefinisikan dalam *Ontologi Bursa Kerja*. Tiap parameter fungsional *input/output* dalam *query* akan dibandingkan dengan tiap-tiap parameter *input/output* dalam *web service* yang telah diregistrasikan dalam *Service Registry* untuk ditentukan derajat similaritasnya. Proses penentuan nilai similaritas ini dihitung dengan menggunakan algoritma *service matching* empat level, adapun rincian nilai similaritas terdiri dari *exact*, *plugin*, *subsume* dan *fail*. Untuk menghasilkan *web service* yang paling sesuai ditentukan dengan metode *Simple Additive Weighting (SAW)* berdasarkan atribut-atribut: nilai similaritas parameter *input/output*, rata-rata waktu eksekusi dan *hit service*. Hasil perhitungan seluruh *web service* dengan penjumlahan terbobot kemudian diranking berdasarkan perolehan total *score* masing-masing *web service*.

Proses tahap kedua adalah *Job Matching*, proses ini diawali dengan memilih *web service* hasil perankingan pada *Service Discovery* dan menentukan preferensi pencarian meliputi interval atau batas akhir pencarian informasi. Proses eksekusi web service dilakukan secara periodik sampai waktu batas akhir yang telah ditentukan, informasi lowongan pekerjaan hasil eksekusi kemudian akan ditransformasikan ke bentuk semantik dengan bahasa *XSLT* dan mengacu pada *vocabulary* yang telah didefinisikan dalam domain ontologi bursa kerja sebagai basis pengetahuannya. Proses selanjutnya adalah membandingkan antar *vocabulary* yang digunakan untuk mendeskripsikan profil pencari kerja dan spesifikasi lowongan pekerjaan dengan menggunakan metode *Semantic similarity* yang ditentukan berdasarkan nilai *semantic distance*. Adapun proses pencocokan mengacu pada atribut bidang dan posisi pekerjaan, pendidikan, pengalaman, umur, jenis kelamin dan lokasi pekerjaan. Informasi pekerjaan yang memenuhi dihitung berdasarkan nilai similaritas tiap spesifikasi antara profil pencari kerja dengan informasi lowongan yang ditawarkan dan dikalikan dengan bobot masing-masing atribut. Selanjutnya perolehan nilai similaritas tiap-tiap lowongan pekerjaan yang ditawarkan diranking secara *descending*.

2.3 Domain Ontologi Bursa Kerja

Knowledge base utama dari sistem berupa skema *OWL Ontology* yang berisi kumpulan kosa kata (*vocabulary*) yang mendeskripsikan hubungan makna/semantik dari obyek-obyek dalam lingkup/domain bursa kerja. Adapun jenis *vocabulary* ini terdiri dari konsep (*concept*) atau kelas (*class*), individu (*instance*) dan relasi (*property*).

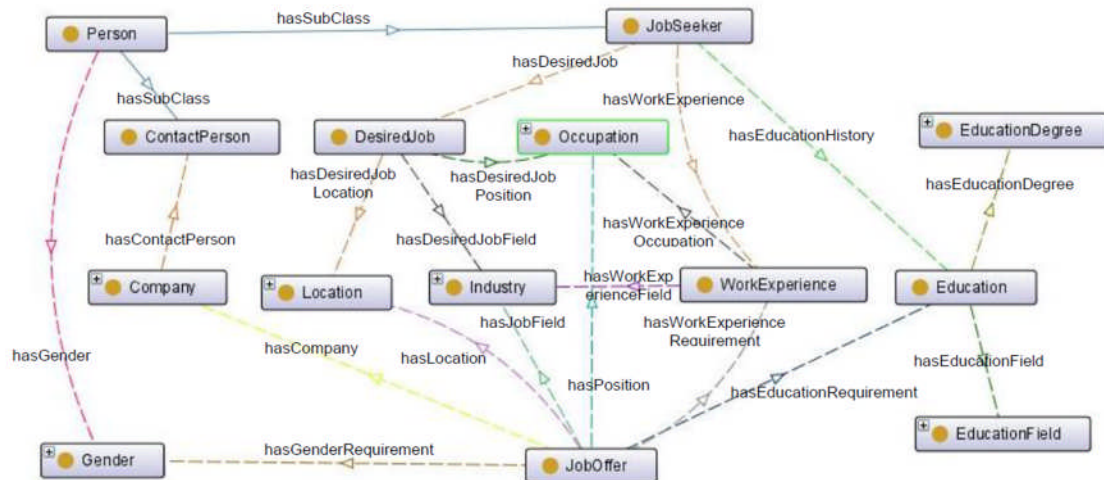
Relasi atau *property* terdiri dari dua macam yaitu *ObjectProperty* dan *DatatypeProperty* dimana *ObjectProperty* menghubungkan antar Kelas (*Class*), dan *DatatypeProperty* merelasikan antara Kelas dengan obyek data (*Literal*). Sementara individu (*instance*) merupakan obyek realisasi dari konsep/kelas.

Untuk menentukan nilai similaritas dengan algoritma *semantic matching* antara informasi pekerjaan yang ditawarkan (*JobOffer*) dengan informasi profil pencari kerja

(*JobSeeker*) disusun beberapa klasifikasi konsep-konsep kelas sebagai *vocabulary* terkontrol dalam struktur hirarkis/taksonomi dan merupakan bagian ontology bursa kerja di antaranya:

1. klasifikasi jabatan/posisi pekerjaan (*Occupation*) dengan mengacu pada *International Standard Classification of Occupation (ISCO-08)* [7].
2. klasifikasi sektor usaha (*Industry*) disusun mengacu pada *International Standard Industrial Classification of All Economic Activities (ISIC)* [8].
3. klasifikasi lokasi (*Location*) disusun dengan mengacu pada *Klasifikasi Perkotaan dan pedesaan di Indonesia* [9].
4. klasifikasi jenjang pendidikan (*EducationDegree*) dan klasifikasi jurusan pendidikan (*EducationField*) yang mengacu pada *International Standard Classification of Education (ISCED)* [10].

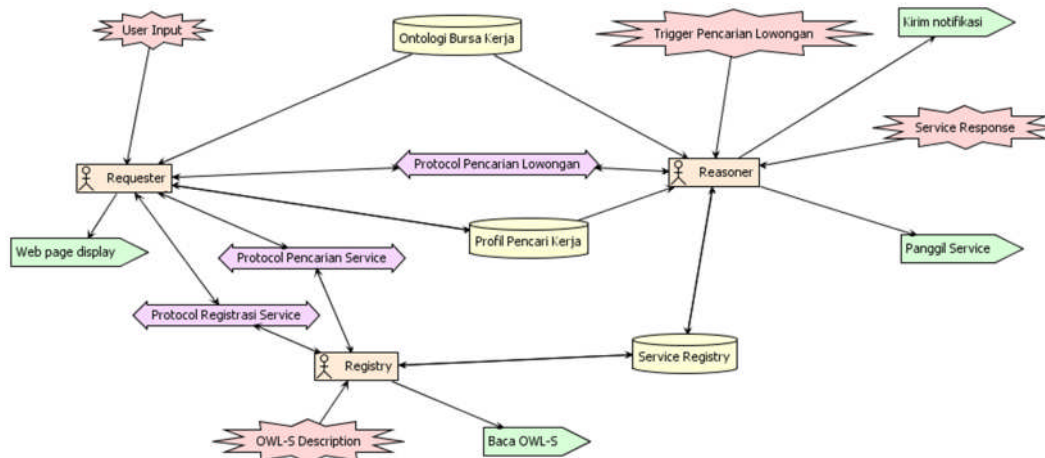
Hasil perancangan ontologi bursa kerja secara grafis ditunjukkan pada Gambar 2.



Gambar 2 Domain ontologi bursa kerja

2.4 Rancangan Agen

Pada Gambar 3 diperlihatkan interaksi antar agen untuk mengimplementasikan tiap-tiap skenario dari sistem. *Agen Requester* berinteraksi dengan *Agen Registry* melalui protokol pencarian service dan protokol registrasi service setelah dibangkitkan dengan *percept user input*. Selain itu *Agen Requester* juga berinteraksi dengan *Agen Reasoner* melalui protokol pencarian lowongan dengan mengirimkan pesan permintaan pencarian informasi lowongan. Dalam Sistem overview diagram juga dideskripsikan basis data sebagai belief dari agen, sebagai *knowledge* yang dibutuhkan oleh sistem, meliputi Basis data Profil Pencari Kerja, Basis data *Service Registry*, Ontologi Bursa Kerja.



Gambar 3 System overview diagram

2.5 Semantic Matching

Menurut [11,12,13], derajat kemiripan konsep antara *query* dengan *resource* dalam *domain ontology* diperoleh dari besaran jarak (*distance*) berdasarkan hubungan hirarki antar konsep secara tematik. Jika diberikan dua konsep c_1 dan c_2 , maka dari dua konsep ini dapat ditentukan besaran jarak hirarkis antara c_1 dan c_2 , yang dinotasikan dengan $d_c(c_1, c_2)$, dengan demikian derajat kemiripan antara c_1 dan c_2 dapat diperoleh dengan persamaan 1.

$$\text{sim}_c(c_1, c_2) = 1 - d_c(c_1, c_2) \quad (1)$$

Jarak antara dua konsep dihitung berdasarkan jarak posisi antar node konsep dalam struktur hirarki konsep dalam domain ontologi, setiap node di dalam hirarki konsep memiliki nilai yang disebut *milestone*. Nilai *milestone* ini dapat diperoleh dengan persamaan 2.

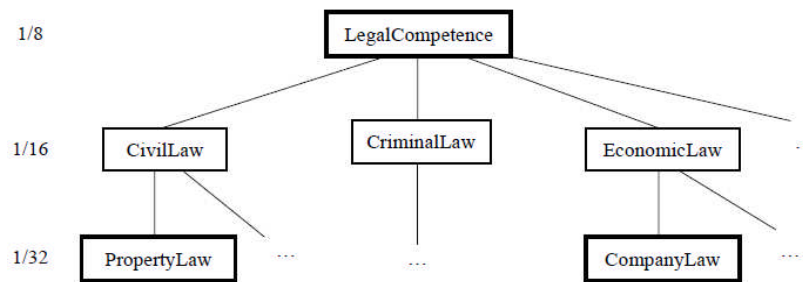
$$\text{milestone}(n) = \frac{1/2}{k^{l(n)}} \quad (2)$$

di mana k adalah faktor tetapan yang nilai lebih besar dari 1 yang menunjukkan tingkat di mana nilai menurun sepanjang hirarki (saat ini, kita menetapkan k sama dengan 2), dan $l(n)$ adalah kedalaman node n dalam hirarki (secara konservatif dipilih jalan terpanjang dari node ke root untuk mengukurnya). Untuk akar (*root*), $l(\text{root}) = 0$.

$$d_c(c_1, c_2) = d_c(c_1, ccp) + d_c(c_2, ccp) \quad (3)$$

$$d_c(c, ccp) = \text{milestone}(ccp) - \text{milestone}(c) \quad (4)$$

Pada persamaan 3 dan 4, konsep c_1 dan c_2 dalam konsep hirarki memiliki *closest common parent (ccp)* atau kelas *parent* terdekat, sehingga jarak antara dua konsep c_1 dan c_2 dapat ditentukan berdasarkan nilai *milestone*.



Gambar 4 Contoh *Skill Ontology* dan nilai *milestone*[12]

Pada Gambar 4 diilustrasikan perhitungan *semantic similarity* antar dua konsep, dimisalkan dua konsep masing-masing `PropertyLaw` dan `EconomicLaw` akan dicari nilai similaritasnya, tahap pertama ditentukan *closest common parent* (ccp), kelas yang paling memenuhi adalah `LegalCompetence`, karena memiliki jarak terdekat antara `PropertyLaw` dan `EconomicLaw`. Tahap berikutnya menentukan nilai *distance* antara dua konsep *class* tersebut.

$$\begin{aligned}
 d_c(\text{PropertyLaw}, \text{EconomicLaw}) &= d_c(\text{PropertyLaw}, \text{LegalCompetence}) + \\
 & d_c(\text{EconomicLaw}, \text{LegalCompetence}) \\
 &= (1/8 - 1/32) + (1/8 - 1/16) \\
 &= 0.21875
 \end{aligned}$$

sehingga nilai similaritas dua konsep di atas dapat ditentukan dengan

$$\text{Sim}_c(\text{PropertyLaw}, \text{EconomicLaw}) = 1 - 0.21875 = 0.78125$$

dengan demikian antara konsep Class `PropertyLaw` dengan `EconomicLaw` memiliki nilai kemiripan sebesar 0.78125.

3. HASIL DAN PEMBAHASAN

3.1 Proses Matching

Proses *matching* antara lowongan pekerjaan dengan *pencari kerja* ditentukan oleh atribut jenis kelamin, usia, riwayat pendidikan, riwayat pengalaman dan minat pekerjaan. Selisih jarak antara konsep yang diperbandingkan akan menentukan besaran nilai similaritas.

Data nyata untuk pengujian sistem digunakan 10 SWS yang diperoleh dari hasil pengkonversian dari 10 web service dengan menggunakan tool WSDL2OWLS. Selain itu untuk mensimulasikan output berupa informasi pekerjaan dari setiap web service telah diinputkan 79 informasi pekerjaan yang dikutip dari portal jobsdb.com dan 55 informasi pekerjaan dari portal infokerja.depnakertrans.go.id. Setiap informasi pekerjaan yang digunakan dalam pengujian ini memiliki rincian atribut meliputi judul pekerjaan, perusahaan pemilik pekerjaan, bidang, posisi dan lokasi pekerjaan, syarat jenis kelamin, syarat usia, syarat pendidikan dan syarat pengalaman bekerja.

3.2 Pengujian Service Discovery

Pengujian pada tahap ini untuk memastikan dapat berjalannya proses pencarian SWS dalam menemukan kembali penyedia informasi lowongan pekerjaan yang terbaik yang akan digunakan pada proses pencarian informasi pekerjaan oleh pencari kerja. Proses pengujian dikerjakan dengan memberikan 10 sampel SWS yang telah diregistrasikan ke dalam basis data service registry. SWS yang diregistrasi ke basis data yang tersusun atas kode, URL owls, parameter input/output, rata-rata waktu eksekusi dan hit pemanggilan web service.

Diberikan lima test case sebagai kueri pencarian SWS ke dalam sistem, masing-masing tersusun atas kombinasi parameter input web service yang berbeda-beda, sedangkan parameter

output kueri diasumsikan memiliki tipe parameter yang sama yaitu Lowongan Pekerjaan. Sedangkan untuk mendapatkan SWS yang terbaik diberikan nilai bobot (kepentingan) tiga kriteria yaitu bobot similaritas parameter input/output = 0.6, rata-rata waktu eksekusi = 0.2 dan hit = 0.2.

Berdasarkan pengujian service discovery pertama sampai kelima dengan memberikan kueri pencarian yang berbeda-beda sistem aplikasi telah dapat merespon dengan baik. Dimana query dengan parameter input yang lebih detail/lemgkap akan menghasilkan jenis web service yang ditampilkan lebih banyak.

3.3 Pengujian Job Matching

Pengujian job matching ditujukan untuk memastikan aplikasi dapat menyajikan informasi lowongan pekerjaan terbaik sesuai dengan data personal pencari kerja. Informasi pekerjaan terbaik ditentukan oleh nilai semantic distance terkecil dari perbandingan setiap atribut pencari kerja dengan atribut lowongan pekerjaan.

Berdasarkan hasil pengujian empat tahap job matching dengan memberikan nilai bobot yang berbeda-beda akan mempengaruhi prioritas posisi informasi lowongan pekerjaan yang ditampilkan. Konfigurasi nilai bobot setiap atribut profil pencari kerja akan mempengaruhi besaran nilai similaritas setiap lowongan pekerjaan, semakin besar nilai similaritas yang didapatkan maka semakin dekat dan sesuai dengan profil pencari kerja.

Ketersediaan nilai atribut yang dibanding baik profil pencari kerja (*job seeker*) maupun lowongan pekerjaan (*job offer*) akan mempengaruhi nilai similaritas yang dihasilkan. Setiap nilai atribut lowongan pekerjaan merupakan prasyarat yang harus dipenuhi pencari kerja, dengan demikian jika setiap nilai atribut lowongan pekerjaan telah didefinisikan maka nilai atribut pencari kerja yang berhubungan harus tersedia, jika tidak maka nilai kedekatannya adalah 0, seperti ditunjukkan pada Tabel 1. Namun sebaliknya, apabila nilai atribut pencari kerja tersedia sedangkan nilai atribut lowongan pekerjaan tidak tersedia maka nilai similaritasnya adalah 1, karena tidak melanggar prasyarat lowongan pekerjaan.

Tabel 1 Pengaruh kelengkapan nilai atribut *job offer* dan *job seeker* terhadap nilai similaritas

Nilai Atribut <i>Job Offer</i>	Similaritas	
	Terdefinisi	0
Tidak terdefinisi	0	1
	Tidak Terdefinisi	Terdefinisi
	Nilai Atribut Profil <i>Job Seeker</i>	

Berdasarkan asumsi di atas maka dapat dinyatakan bahwa semakin lengkap profil pencari kerja memiliki kecenderungan mendapat hasil nilai similaritas terhadap lowongan pekerjaan lebih tinggi dibanding dengan profil pencari kerja yang tidak lengkap. Semakin banyak data profil yang didefinisikan akan memperbesar setiap prasyarat lowongan pekerjaan yang terpenuhi, sehingga akan menghasilkan informasi lowongan pekerjaan yang lebih beragam.

3.2 Pengujian User Acceptance

Sistem aplikasi yang diusulkan telah diujikan terhadap 30 responden dengan menjawab 15 pertanyaan kuisioner. Kuisioner ini dimaksudkan untuk mengukur tingkat penerimaan pengguna terhadap sistem aplikasi yang diusulkan, terutama untuk tanggapan pengguna terhadap keluaran aplikasi berupa lowongan pekerjaan yang direkomendasikan. Variabel independen dalam uji statistik ini meliputi: data personal/DP (jenis kelamin dan umur), tingkat

pendidikan/TP (jenjang pendidikan, jurusan pendidikan dan nilai kelulusan), pengalaman kerja/PK (posisi pengalaman kerja, bidang pengalaman kerja dan lama pengalaman), minat pekerjaan/MP (minat jabatan, minat bidang kerja, dan lokasi kerja), sedangkan variabel dependen berupa Kemampuan Aplikasi/KM.

Berdasarkan jawaban responden yang telah diterima, ditunjukkan hasil uji determinasi pada Gambar 5 dan uji regresi pada Gambar 6.

Model Summary^b

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	Durbin-Watson
1	.774 ^a	.600	.535	1.22621	1.938

a. Predictors: (Constant), DP, PK, MP, TP

b. Dependent Variable: KM

Gambar 5 Hasil uji determinasi

Berdasarkan Gambar 5 bahwa variabel data personal (DP), tingkat pendidikan (TP), pengalaman kerja (PK) dan minat pekerjaan (MP) dapat menentukan/menjelaskan variabel kemampuan aplikasi (KM) sebesar 77,40%, sedangkan sisanya 22,60% diterangkan oleh variabel lain diluar model. Dengan demikian data profil responden (pencari kerja) yang diinputkan ke sistem telah dapat memberikan hasil keluaran berupa lowongan pekerjaan yang sesuai profil responden.

Coefficients^a

Model	Unstandardized Coefficients		Standardized Coefficients	t	Sig.
	B	Std. Error	Beta		
¹ (Constant)	.034	2.157		.016	.988
MP	.228	.142	.214	1.600	.122
PK	.270	.146	.252	1.851	.076
TP	.584	.173	.491	3.375	.002
DP	.346	.254	.188	1.359	.186

a. Dependent Variable: KM

Gambar 6 Hasil uji regresi

Hasil uji regresi seperti ditunjukkan pada Gambar 6 didapatkan persamaan regresi untuk variabel kemampuan aplikasi (KM) = 0,034 + 0,228MP + 0,270PK + 0,584TP + 0,346DP. Jika dilihat lebih mendetail pengaruh variabel independen terhadap kemampuan aplikasi, dari keempat variabel independen tersebut yang memiliki nilai signifikan dengan nilai kurang dari 10% (0.10) adalah variabel pengalaman kerja (PK) memiliki nilai 0,076 dan tingkat pendidikan (TP) memiliki nilai 0,002. Sedangkan yang memiliki nilai signifikan kurang dari 5% (0.05) adalah variabel tingkat pendidikan dengan nilai 0,002.

Dengan demikian rekomendasi lowongan pekerjaan yang dihasilkan oleh aplikasi dinyatakan sesuai karena banyak dipengaruhi oleh riwayat pendidikan dan pengalaman bekerja responden.

3.3 Pengujian Kinerja Aplikasi

Pengukuran kinerja aplikasi dilakukan dengan pencatatan waktu yang diperlukan untuk mengerjakan proses pencarian dalam aplikasi. Dalam hal ini penulis melakukan simulasi beban tugas pencarian ke agen dengan memberikan jumlah proses yang berbeda-beda secara bertahap. Proses pengujian difokuskan untuk mengamati selisih waktu yang dibutuhkan antar setiap tahap pengujian.

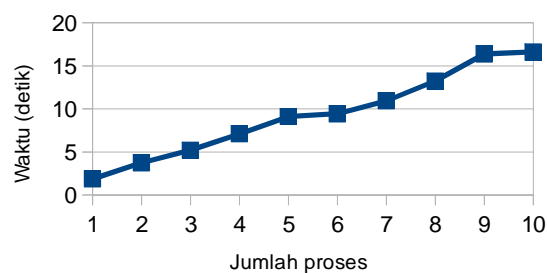
Pada Tabel 2, tahap pengujian kinerja pencarian dilakukan sebanyak 10 tahap, dimulai dari tahap 1 (satu) proses, 2 (dua) proses sampai 10 (sepuluh) proses. Setiap tahap pengujian dilakukan sebanyak lima pengulangan pengujian kemudian dihitung rata-rata waktu yang diperoleh. Pada tahap pengujian pertama (1 proses) diperoleh rata-rata waktu 1.876 detik, tahap pengujian kedua (2 proses) diperoleh waktu rata-rata 3.740 detik, tahap pengujian ketiga (3 proses) diperoleh waktu rata-rata 5.219 detik, tahap pengujian keempat (4 proses) diperoleh waktu rata-rata 7.102 detik, tahap pengujian kelima (5 proses) diperoleh waktu rata-rata 9.106 detik, tahap pengujian keenam (6 proses) diperoleh waktu rata-rata 9.437 detik, tahap pengujian ketujuh (7 proses) diperoleh waktu rata-rata 10.942 detik, tahap pengujian kedelapan (8 proses) diperoleh waktu rata-rata 13.245 detik, tahap pengujian kesembilan (9 proses) diperoleh waktu rata-rata 16.385 detik, dan tahap pengujian kesepuluh (10 proses) diperoleh waktu rata-rata 16.608 detik.

Berdasarkan kesepuluh tahap pengujian, setiap penambahan satu proses pencarian membutuhkan waktu pemrosesan yang lebih lama dibandingkan dengan jumlah proses yang lebih sedikit. Pada Gambar 7 diperlihatkan rasio antara lama waktu yang diperlukan dengan jumlah proses pencarian pekerjaan berjalan secara linear, secara konsisten setiap penambahan beban proses juga membutuhkan waktu pemrosesan lebih banyak rata-rata 1.63 detik.

Tabel 2 Waktu yang dibutuhkan dengan jumlah proses pencarian

Test	Jumlah Proses Pencarian (satu waktu)(detik)									
	1	2	3	4	5	6	7	8	9	10
1	1.839	4.886	5.320	6.947	9.265	9.694	10.810	13.451	17.194	16.997
2	1.753	3.506	5.627	7.265	9.514	9.321	10.839	13.392	15.930	17.132
3	2.106	3.586	4.945	7.133	9.029	9.210	10.533	13.252	16.465	16.487
4	2.011	3.379	5.134	7.172	9.019	9.595	11.005	13.577	16.264	16.266
5	1.670	3.341	5.070	6.994	8.704	9.366	11.525	12.551	16.074	16.159
Rata-rata	1.876	3.740	5.219	7.102	9.106	9.437	10.942	13.245	16.385	16.608

Rasio waktu dengan jumlah proses pencarian



Gambar 7 Grafik rasio waktu dengan jumlah proses pencarian

4. KESIMPULAN

Berdasarkan dari hasil analisa, perancangan, implementasi dan pengujian didapatkan kesimpulan:

1. Service discovery dalam sistem telah dapat menampilkan web service terbaik berdasarkan nilai kecocokan parameter input/output terbesar, nilai hit terbesar dan waktu eksekusi terkecil.
2. Sistem aplikasi yang dibangun telah dapat memberikan informasi lowongan pekerjaan yang sesuai dengan data personal, riwayat pendidikan, pengalaman bekerja dan minat pekerjaan pencari kerja.
3. Semakin detail profil pencari kerja yang didefinisikan dari pada lowongan pekerjaan maka akan memperbesar kedekatan hubungan keduanya, namun sebaliknya semakin kurang detail profil pencari kerja yang didefinisikan dibandingkan lowongan pekerjaan akan semakin memperkecil kedekatan yang didapatkan.
4. Setiap penambahan jumlah proses pencarian lowongan pekerjaan yang dibebankan ke aplikasi akan menambahkan jumlah waktu yang diperlukan. Adapun penambahan jumlah proses pencarian bersifat linear terhadap waktu yang dibutuhkan.
5. Software agen telah dapat difungsikan sebagai perantara yang memudahkan pencari kerja dalam menemukan lowongan pekerjaan terbaik.

5. SARAN

Hasil dari pengujian yang dilakukan, masih banyak kekurangan sehingga perlu dikembangkan untuk meningkatkan kinerja dari aplikasi yang dibangun, adapun saran yang diberikan:

1. Perlu dikembangkan bagian aplikasi yang dapat mentransformasikan format XML ke bentuk semantic RDF/OWL secara otomatis dari WSDL menjadi OWL-S terutama yang melibatkan tipe data kompleks.
2. Aplikasi dapat dikembangkan dengan menambahkan kemampuan *Service Composition* sehingga memungkinkan aplikasi dapat menerima permintaan-permintaan user yang kompleks dan variatif.
3. Aplikasi perlu dikembangkan untuk memfasilitasi penyedia kerja dapat menyeleksi secara otomatis profil pencari kerja terhadap kesesuaian informasi lowongan pekerjaan yang diberikan.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada segenap pimpinan Direktorat Pendidikan Tinggi dan pimpinan STMIK Sinar Nusantara Surakarta yang telah memberi dukungan financial terhadap penelitian ini.

DAFTAR PUSTAKA

- [1] P. Montuschi, V. Gatteschi, F. Lamberti, A. Sanna, and C. Demartini, Job Recruitment and Job Seeking Processes: How Technology Can Help, *IT Prof.*, vol. 16, no. 5, pp. 41–49, 2014.
- [2] L. Yang, Z. Hu, and J. Long, Service of searching and ranking in a semantic-based expert information system, in *Proceedings - 2010 IEEE Asia-Pacific Services Computing Conference, APSCC 2010*, 2010, pp. 609–614.
- [3] M. Mochol, H. Wache, and L. Nixon, Improving the accuracy of job search with semantic techniques, *Bus. Inf. Syst.*, 2007.

-
- [4] D. Celik, A. Karakas, G. Bal, C. Gultunca, A. Elci, B. Buluz, and M. C. Alevli, Towards an Information Extraction System Based on Ontology to Match Resumes and Jobs, in *Proceedings - International Computer Software and Applications Conference*, 2013, pp. 333–338.
- [5] E. Malherbe, M. Diaby, M. Cataldi, E. Viennet, and M.-A. Aufaure, Field selection for job categorization and recommendation to social network users, in *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)*, 2014, no. Asonam, pp. 588–595.
- [6] L. Zhou, An Approach of Semantic Web Service Discovery, *2010 Int. Conf. Commun. Mob. Comput.*, pp. 537–540, Apr. 2010.
- [7] International Labour Organization, *International Standard Classification of Occupations (ISCO-08)*, vol. I. Geneva, 2012.
- [8] Anonim, *International Standard Industrial Classification of All Economic Activities (ISIC), Rev.4*. New York, USA: United Nations Publication, 2008.
- [9] Anonim, *Klasifikasi Perkotaan dan Perdesaan di Indonesia*. Badan Pusat Statistik, 2010.
- [10] Anonim, *International Standard Classification of Education (ISCED) 2011*. Unesco, 2012.
- [11] Zhong, J., Zhu, H., Li, J. and Yu, Y., 2002, Conceptual graph matching for semantic search, *10th International Conference on Conceptual Structures, ICCS 2002 Borovets, Bulgaria, July 15–19, 2002 Proceedings*, Springer Berlin Heidelberg, pp. 92–106.
- [12] Bizer, C., Heese, R. and Mochol, M., 2005, The impact of semantic web technologies on job recruitment processes, *Wirtschaftsinformatik*, pp. 1367–1381.
- [13] Oldakowski, R. and Bizer, C., 2005, SemMF: A framework for calculating semantic similarity of objects represented as RDF graphs, *4th International Semantic Web Conference*, pp. 2–4.