

IMPLEMENTASI *HYBRID-BASED RECOMMENDATION* DIDALAM SISTEM REKOMENDASI PENCARIAN PEKERJAAN BERBASIS WEB

¹Andre Hasudungan Lubis, ²Solly Aryza Lubis

¹Staf Pengajar Fakultas Teknik Prodi Teknologi Informasi Universitas Medan Area

²Staf Pengajar Fakultas Teknik Prodi Elektro Universitas Pembangunan Panca Budi
andrelubis2201@gmail.com, sollyaryzalubis@gmail.com

ABSTRAK

Aktivitas utama yang dilakukan sehari – hari seorang individu untuk mencari uang adalah dengan cara bekerja. Mencari pekerjaan yang sesuai dengan latar belakang pendidikan yang kita tekuni tidaklah mudah. Banyak individu yang tidak mengetahui kemampuan diri mereka sendiri dan keterbatasan informasi lowongan pekerjaan juga menjadi hambatan bagi pelamar yang ingin mencari pekerjaan. Karena itu, diperlukan satu ‘alat bantu’ yang dapat memberi rekomendasi bidang pekerjaan apa yang sesuai dengan belakang pendidikan yang bersangkutan. Metode *hybrid approach* adalah dengan menggabungkan teknik collaborative-filtering (algoritma *decision tree*) dan content-based (algoritma *nearest neighbor*). Algoritma *decision tree* digunakan untuk pengklasifikasian bidang pekerjaan sedangkan untuk rekomendasi pekerjaan, digunakan algoritma *nearest neighbor*. Pada *nearest neighbor* digunakan rumus *similarity* untuk menghitung kedekatan antara pelamar dan lowongan pekerjaan berdasarkan pencocokan bobot dan atribut yang ada. Output yang dihasilkan dari sistem ini berupa daftar rekomendasi pekerjaan yg sesuai dengan latar belakang pendidikan pelamar.

Kata Kunci: Rekomendasi Pekerjaan, Hybrid Approach, Decision Tree, Nearest Neighbor.

I. PENDAHULUAN

Pekerjaan merupakan aktivitas utama yang dilakukan manusia dengan menyelesaikan suatu tugas atau kerja yang menghasilkan uang bagi seseorang. Pekerjaan biasa disebut juga dengan profesi. Profesi pada zaman sekarang terdapat bermacam – macam bidang. Mencari pekerjaan yang sesuai dengan minat, bakat dan latar belakang pendidikan yang kita tekuni itu tidak mudah. Banyak individu yang tidak mengerti atau mengetahui kemampuan diri mereka sendiri. Selain itu, keterbatasan informasi lowongan pekerjaan juga menjadi hambatan bagi pelamar yang ingin mencari pekerjaan. Faktor - faktor tersebut menjadi hambatan bagi seorang pelamar untuk mencari pekerjaan yang sesuai dengan kemampuan dan latar belakang mereka, terutama untuk seorang *fresh graduate* yang masih belum mempunyai pengalaman pekerjaan.

Untuk mengatasi masalah tersebut, diperlukan satu ‘alat bantu’ yang dapat memberi rekomendasi (kepada pencari kerja) bidang pekerjaan apa yang sesuai dengan minat, bakat dan latar belakang yang bersangkutan. Sistem ini juga akan menyimpan berbagai macam lowongan pekerjaan dari berbagai bidang sehingga hasil dari sistem ini akan memberikan saran lowongan pekerjaan yang sesuai dengan individu tersebut.

Sistem rekomendasi didefinisikan sebagai aplikasi untuk mengusulkan informasi dan menyediakan fasilitas yang diinginkan pengguna dalam membuat suatu keputusan [1]. Sistem rekomendasi [2] dibagi 3 jenis yaitu content-based recommendation, Collaborative recommendation dan hybrid approaches. Penerapan hybrid approaches adalah dengan menggabungkan teknik collaborative-filtering dan content-based diharapkan dapat membantu keterbatasan yang ada pada kedua metode sehingga menghasilkan output yang lebih baik [2]. *Decision tree* adalah salah satu algoritma dalam collaborative-filtering dan algoritma populer yang sudah banyak digunakan [3]. Kelebihan dari *Decision tree* antara lain; menyederhanakan masalah yang kompleks menjadi lebih simpel.

Pada content-based filtering digunakan Algoritma *Nearest Neighbor* yang merupakan algoritma untuk menghitung kedekatan antara kasus baru dengan kasus lama, yaitu berdasarkan pencocokan bobot dari sejumlah atribut yang ada [4].

II. IDENTIFIKASI MASALAH

Dalam upaya membantu pelamar untuk memberi informasi lowongan pekerjaan dan mencari pekerjaan yang sesuai dengan latar belakang pelamar tersebut, maka diperlukannya suatu pendekatan atau sistem yang dapat digunakan untuk menyediakan informasi

lowongan pekerjaan dan dapat juga digunakan untuk memberikan rekomendasi bagi pelamar pekerjaan dalam menentukan pekerjaan yang sesuai dengan latar belakang pendidikan pelamar.

III. METODOLOGI PENELITIAN

A. Basis Data (Database)

Database management system (DBMS) adalah sebuah set dari fasilitas yang terorganisir untuk mengakses dan mempertahankan satu atau banyak basis data. Penggunaan DBMS dilakukan di dalam integrasi dari suatu sistem yang menggunakan data tersebut dalam sebuah perangkat lunak [13]. Menurut Beynon-Davies[13], DBMS dibagi atas 4 jenis yaitu : *Data definition, Data maintenance, Data retrieval* dan *Data Control*.

B. Sistem Rekomendasi

Sistem Rekomendasi adalah peralatan perangkat lunak dan teknik yang menyediakan saran untuk *items* yang bisa digunakan oleh *user* [14]. Secara general sistem rekomendasi didefinisikan sebagai sistem pendukung yang membantu *user* untuk mencari informasi, produk dan servis dengan menggabungkan dan menganalisa saran dari *user* lain, yang berarti meninjau dari beberapa pihak dan *user* atribut [15].

Sistem rekomendasi menjadi sebuah penelitian bidang yang penting sejak munculnya makalah pertama tentang *collaborative-filtering* pada pertengahan 1990an [16]. Tujuan dari sistem rekomendasi adalah menghasilkan rekomendasi yang berguna kepada *user* untuk *items* atau produk yang paling menguntungkan bagi *user* [17].

Sistem Rekomendasi menurut Melville dan Sindhvani[17], terbagi atas 3 jenis, yaitu: Content-based Filtering, Collaborative Filtering dan Hybrid-based Filtering.

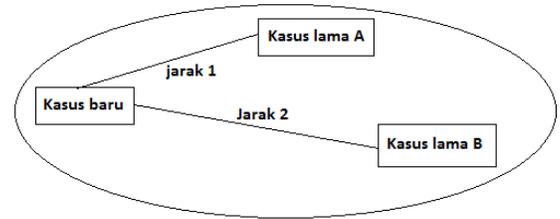
C. Decision tree

Klasifikasi data bisa dideskripsikan sebagai pengawas dari sebuah learning algoritma di dalam proses mesin pembelajaran. *Decision tree* adalah sebuah representasi dari prosedur keputusan untuk menentukan kelas dan dikonstruksikan menjadi algoritma non-incremental tree-induction atau algoritma incremental tree-induction [18].

Metode *Decision tree* pada aplikasi ini digunakan untuk penyaringan data yang telah diinput oleh *user* sebelumnya. Metode *Decision tree* dilakukan pada penyaringan data yang telah diinput sebelumnya sehingga menghasilkan parameter – parameter yang sesuai untuk dilakukannya rangking.

D. Algoritma Nearest Neighbor

Metode *nearest neighbor* langsung mengeksploitasi jarak antara data sampel untuk melakukan klasifikasi. *Nearest neighbor* secara independen mengevaluasi jarak antara data yang satu dengan data yang lain [19]. *Rules nearest neighbor* mengidentifikasi katagori dari data poin yang baru (kasus baru) berdasarkan *nearest neighbor* dari data (kasus lama) yang telah diketahui nilainya [20].



Gambar 1 kedekatan kasus

Pada gambar 1 dapat dilihat bahwa terdapat dua kasus lama, yaitu kasus A dan kasus B. Pada saat kasus baru muncul, solusi yang digunakan untuk memecahkan kasus baru tersebut adalah dengan kasus lama A dikarenakan jarak 1 yaitu kasus baru dengan kasus A lebih dekat dibandingkan jarak 2 yaitu jarak kasus baru dengan kasus B. Maka diambil kesimpulan bahwa kasus baru dipecahkan oleh kasus A.

Untuk memperoleh rangking pada *nearest neighbor*, sistem melakukan perhitungan kedekatan (*similarity*) antara 2 kasus. Berikut adalah rumus dari yang digunakan di dalam metode *nearest neighbor*.

$$similarity(T, S) = \frac{1}{\sum^n (T_i \cdot S_i) * W_i}$$

Dimana :

T : Kasus baru

S : Kasus yang ada dalam memori/basisdata (kasus lama)

n : Jumlah atribut dalam setiap kasus

i : atribut variabel antara 1 s/d n

f : fungsi *similarity* atribut i antara kasus T dan kasus S

w : bobot yang diberikan pada atribut ke-i

Penentuan kedekatan biasanya berada pada nilai antara 0 s/d 1. Nilai 0 berarti bahwa kedua kasus mutlak tidak memiliki kesamaan, sebaliknya untuk nilai 1, berarti kasus tersebut mutlak memiliki kesamaan.

E. Data yang digunakan

Data yang digunakan pada sistem ini diambil dari 2 sumber yaitu jobsdb untuk data lowongan pekerjaan dan Universitas Sumatera Utara untuk data jurusan universitas. Pada tabel 1 berikut akan dijabarkan

tentang sumber, jumlah data dan tahun dari data yang akan digunakan :

Tabel 1 Sumber data

No	Sumber	Jumlah data	Bln/Tahun	Data yang diambil
1	Jobsdb (www.jobsdb.com)	1600	Januari 2013 – Maret 2013	data lowongan pekerjaan
2	Univesitas Sumatera Utara (www.usu.ac.id)	58	Maret 2013	Data fakultas dan jurusan yang ada di USU

Data yang digunakan dalam penelitian ini bersumber dari data website pencarian pekerjaan yaitu jobsDB. Data yang diambil melalui media internet yang diakses dari website jobsdb.com berupa informasi lowongan pekerjaan. Dari sumber yang sama juga digunakan data lowongan pekerjaan tahun 2010 – 2013. Tabel 2 berikut adalah sampel data lowongan pekerjaan yang diambil dari jobsdb.

Tabel 2 Sampel data lowongan pekerjaan

No	Nama Pekerjaan	Jenis Pendidikan	Kualifikasi Pendidikan	IP
1	Web application developer (maven lab pte ltd)	Informasi Teknologi	S1	2,75
2	ASP.Net Developers	Ilmu komputer	D3	3,00
3	Temporary Payroll Executive – PT MindChamps Indonesia	akutansi	D3	2,75
4	Staff Accounting/Staff Accounting SILVER BOX, UD	ekonomi	D3	2,75
5	Finance Staff Eurokars Group	ekonomi	S1	2,75
.
1000	Architects (ARC – SML) SINAR MAS LAND	Arsitektur	S1	2,75

Data jurusan yang digunakan dalam penelitian ini bersumber dari Universitas Sumatera Utara (USU). Data diambil melalui media internet yang diakses dari website <http://www.usu.ac.id/>. Data yang diambil berupa kualifikasi pendidikan, fakultas (bidang pendidikan) dan jurusan pada universitas tersebut. Contoh data yang diambil akan dijabarkan pada tabel 3 berikut :

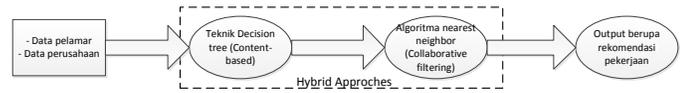
Tabel 3 Sampel data latar belakang pendidikan

No	Kualifikasi Pendidikan	Fakultas	Jurusan
1	D3	MIPA	Kimia
2	D3	Ilmu Budaya	Sastra Inggris
3	D3	Ekonomi	Akutansi
4	S1	MIPA	Kimia
5	S1	FASILKOMTI	Teknologi Informasi
.	.	.	.
.	.	.	.
.	.	.	.

F. Penerapan Hybrid-based recommendation pada data

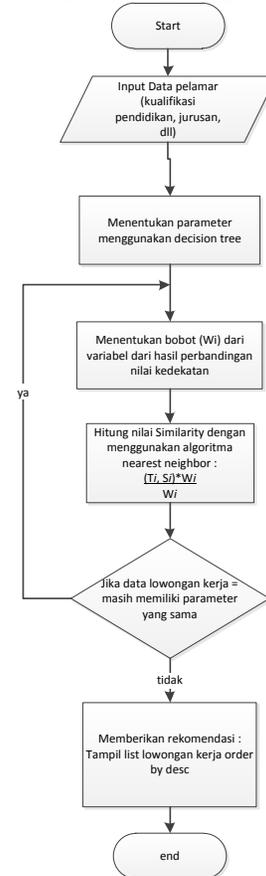
Untuk membangun sistem rekomendasi pada penelitian ini, digunakan metode *hybrid-approches*. Metode *hybrid-approches* ini adalah gabungan dari 2 metode, metode *Content-based* dan metode *Collaborative Filtering*.

Metode pada *Content-based* menggunakan teknik *Decision Tree* dan metode untuk *Collaborative Filtering* menggunakan algoritma *Nearest Neighbor*. Pada gambar 2 berikut akan digambarkan tentang proses penggunaan metode – metode tersebut :



Gambar 2 Kombinasi sekuensial

Adapun Flowchart *hybrid-based recommendation* pada sistem ini yang akan menghasilkan sebuah rekomendasi akan dijabarkan pada gambar 3:



Gambar 3 Flow Chart sistem rekomendasi

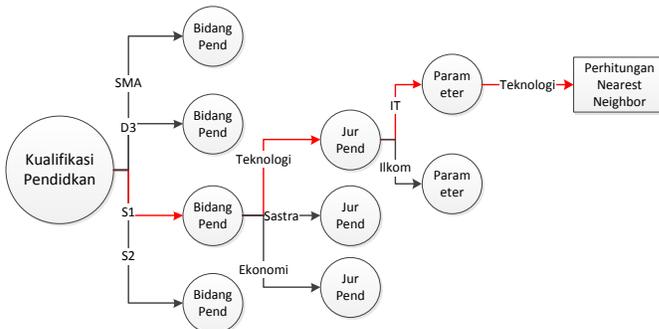
Pada penyeleksian data menggunakan teknik *decision tree*, data yang telah diinput oleh user akan

dibandingkan dengan kasus yang telah disimpan terlebih dahulu di database. Setelah data dibandingkan dan mendapatkan kecocokan antar data, akan menghasilkan parameter yang akan digunakan dalam perhitungan di *Nearest Neighbor*. Pada tabel 4, dapat dilihat bahwa misalkan pelamar menginput data :

Tabel 4 Sampel data pelamar

1	Nama	Nurul Khadijah
2	Alamat	Jln Sei Beras No. 3
3	Bidang Pendidikan	Teknologi
4	Jenis Pendidikan	Informasi Teknologi
5	Kualifikasi Pendidikan	S1
6	IP	3
7	Kemampuan Komputer	Web

Maka dari data tersebut dapat dilihat bahwa kualifikasi pendidikan pelamar S1, bidang pendidikan teknologi dan jenis pendidikan Informasi Teknologi (IT). Dari data tersebut kemudian dilakukan pengkualifikasian sehingga menghasilkan parameter. Yang kemudian dapat dilanjutkan ke perhitungan *nearest neighbor*. Proses penentuan parameter pada *decision tree* dapat dilihat pada gambar 4 berikut :



Gambar 4 penentuan parameter

Nearest neighbor akan memberikan rekomendasi pekerjaan dengan menggunakan informasi ranking berdasarkan data yang telah masukkan oleh *user*. Data *user* yang dimasukkan akan dicari dan dibandingkan dengan data yang tersimpan terlebih dahulu di dalam database.

Masing – masing data akan diberi bobot kemudian data tersebut akan dibandingkan dengan nilai – nilai dari bobot dan kedekatan di dalam variabel.

Variabel – variabel tersebut akan dihitung kesamaannya (*similarity*) menggunakan algoritma *Nearest Neighbor* kemudian disaring atau di-ranking berdasarkan hasil jarak tersebar dari perhitungan bobot setiap variabel.

Pada tabel 5 berikut adalah tabel kedekatan atribut dan bobot variabel pada sistem rekomendasi:

Tabel 5 Bobot Variabel

No	Variabel	Bobot
1	Jenis pendidikan	1
2	Kualifikasi pendidikan	0,75
3	IP (Indeks Prestasi)	0,5
4	Kemampuan komputer	1

a. Menentukan kedekatan nilai pada atribut – atribut dari variabel

Tabel 6 merupakan contoh penentuan nilai pada atribut, yaitu :

Tabel 6 Sampel nilai atribut jenis pendidikan

Nilai 1	Nilai 2	Bobot
IT	Ilkom	0,5
Ilkom	IT	0,5
Ilkom	Ilkom	1
IT	IT	1

1. Menghitung kedekatan kasus baru dengan kasus lama

Kasus Lama (daftar pekerjaan) :

Tabel 7 Tabel Kasus Lama (Daftar Pekerjaan)

No	Jenis pendidikan	Kualifikasi pendidikan	IP	Kemampuan komputer
1	IT	S1	2,75	Web
2	IT	D3	3,0	Prog

Kasus Baru (pelamar perkerjaan) :

Tabel 8 Tabel kasus baru (data pelamar pekerjaan)

No	Jenis pendidikan	Kualifikasi pendidikan	IP	Kemampuan komputer
1	IT	S1	3,0	Web

Kemudian dilakukan langkah – langkah sebagai berikut untuk menentukan *similarity* kasus baru dengan kasus sebelumnya:

1. Menghitung kedekatan kasus baru dengan kasus lama no.1 :

a. Kedekatan nilai atribut jenis jenis pendidikan : IT – IT = 1

- Kasus lama jenis pendidikan = IT
- Kasus baru jenis pendidikan = IT

kemudian melakukan mambandingkan kedekatan atribut :

Tabel 9 Kedekatan Nilai Atribut Jenis Pendidikan

Nilai 1	Nilai 2	Bobot
IT	Ilkom	0,5
Ilkom	IT	0,5
Ilkom	Ilkom	1
IT	IT	1

Maka bobot dari : IT – IT = 1

b. Bobot atribut jenis pendidikan : 1
 bobot 1 dapat kita lihat dari tabel 5. Bobot variabel untuk jenis pendidikan adalah 1

c. Kedekatan nilai atribut Kualifikasi pendidikan : S1 – S1 = 1
 - Kasus lama Kualifikasi pendidikan = S1
 - Kasus baru Kualifikasi pendidikan = S1

Tabel 10 Kedekatan Nilai Atribut Kualifikasi Pendidikan

Nilai 1	Nilai 2	Bobot
S2	S2	1
S1	S1	1
D3	D3	1
SMK	SMK	1
...
SMK	D3	0,25

Maka bobot dari : S1 – S1 = 1

d. Bobot atribut Kualifikasi pendidikan : 0.8

Tabel 11 Bobot Variabel

No	Variabel	Bobot
1	Jenis pendidikan	1
2	Kualifikasi pendidikan	0,75
3	IP (Indeks Prestasi)	0,5
4	Kemampuan komputer	1

e. Kedekatan nilai atribut jenis IP : 3,0 – 2,75 = 0.8

- Kasus lama IP = 3,0 (B)
- Kasus baru IP = 2,75 (C)

Tabel 12 Kedekatan Nilai Atribut IP

Nilai 1	Nilai 2	Bobot
A	A	1
B	B	1
C	C	1
A	B	0,75
A	C	0,5
B	A	0,75
B	C	0,75
C	A	0,5
C	B	0,5

Ket : C = 2,75 – 2,99; B = 3,00 – 3,50; A = 3,51 – 4,00

Maka bobot dari : 3,0 – 2,75 = 0,75

f. Bobot atribut IP : 0.6

Tabel 13 Bobot Variabel

No	Variabel	Bobot
1	Jenis pendidikan	1
2	Kualifikasi pendidikan	0,75
3	IP (Indeks Prestasi)	0,5
4	Kemampuan komputer	1

g. Kedekatan nilai atribut Kemp komputer : web–web = 1

- Kasus lama Kemampuan komputer = web
- Kasus baru Kemampuan komputer = web

Tabel 14 Kedekatan Nilai Atribut Kemampuan Komputer

Nilai	Nilai 2	Bobot
web	web	1
Desain	Desain	1
Jaringan	Jaringan	1
Programing	Programing	1
Programing	Programing	1
web	Desain	0,5
...

h. Bobot atribut Kemampuan komputer : 1

Tabel 15 Bobot Variabel

No	Variabel	Bobot
1	Jenis pendidikan	1
2	Kualifikasi pendidikan	0,75
3	IP (Indeks Prestasi)	0,5
4	Kemampuan komputer	1

Setelah didapat nilai kedekatan dari variabel a, b, c dst, seperti yang telah diajarkan di no 1. maka perhitungan *Similarity* untuk *user* baru dengan *user* no. 1 :

$$\text{Jarak} = \frac{(a*b)+(c*d)+(e*f)+(g*h)}{b+d+f+h}$$

$$\text{Jarak} = \frac{(1*1) + (1*0,75) + (0,5*0,5) + (1*1)}{1 + 0,75 + 0,5 + 1}$$

$$\text{Jarak} = \frac{3,00}{3,25}$$

$$\text{Jarak} = 0,923$$

2. Menghitung kedekatan kasus baru dengan kasus lama no.2:

Setelah menghitung kasus 1, Kemudian dilakukan kembali perhitungan *Similarity* untuk *user* ke-2. perhitungan kedekatan *user* baru dengan *user* no.2 yaitu :

a. Kedekatan nilai atribut jenis jenis pend : IT – IT = 1

b. Bobot atribut jenis pend : 1

c. Kedekatan nilai atribut Kualifikasi pend : S1 – D3 = 0,75

d. Bobot atribut Kualifikasi pend: 0.75

e. Kedekatan nilai atrinut jenis GPA : 3,0 – 3,0 = 1

f. Bobot atribut GPA: 0.75

g. Kedekatan nilai atribut Kemampuan komputer: web – prog = 0,75

h. Bobot atribut Kemampuan komputer : 1

Maka :

$$\text{Jarak} = \frac{(a*b)+(c*d)+(e*f)+(g*h)}{b+d+f+h}$$

$$\text{Jarak} = \frac{(1*1) + (1*0,5) + (1*0,5) + (0,75*1)}{1 + 0,75 + 0,5 + 1}$$

Jarak = $\frac{2,75}{3,25}$
 Jarak = 0,846

Maka hasil jarak dari perhitungan tersebut adalah:

Tabel 16 Hasil perhitungan *similarity*

No	Nama	hasil
1	Kasus 1	0,9471
2	Kasus 2	0,845

Dari langkah 1 – 2 dan tabel 16 dapat diketahui bahwa hasil perhitungan pada kasus no 1 adalah 0,9471 dan hasil perhitungan pada kasus 2 adalah 0,845. Dari kedua hasil tersebut, hasil perhitungan dari kasus 1 yaitu 0,9471 lebih mendekati ke nilai 1 dari pada hasil dari kasus 2 yaitu sebesar 0,845.

Maka dari itu, user pada kasus baru memiliki kemiripan terdekat dengan kasus 1, sehingga hasil rekomendasi terbaik ada pada kasus 1.

IV. HASIL DAN PEMBAHASAN

A. Tampilan Rekomendasi Pekerjaan

Dari penelitian ini akan dihasilkan output berupa rekomendasi pekerjaan yang sesuai dengan latar belakang dan bidang pendidikan seorang pelamar pekerjaan.

Pelamar pekerjaan akan diberikan pilihan lowongan pekerjaan yang disusun berdasarkan ranking pekerjaan yang paling sesuai dengan pelamar. Di daftar tersebut akan ditampilkan nama pekerjaan dan keterangan pekerjaan, seperti misalnya rekomendasi yang paling sesuai yaitu berupa pekerjaan di Ray White Indonesia yang menerima pelamar baru lulus dari universitas dengan hasil perhitungan sebesar 0,860.

Hasil output selanjutnya dari aplikasi ini akan dijabarkan pada gambar 5 berikut ini:



Gambar 5 Hasil output rekomendasi pekerjaan

B. Pengujian data pada sistem

Setelah dilakukan pengujian sistem dengan menggunakan 100 data, maka pengujian selanjutnya

akan dilakukan pada 7 bidang pekerjaan. masing – masing bidang diambil sebanyak 250 data. Sehingga total keseluruhan data sebanyak 1750 data lowongan pekerjaan.

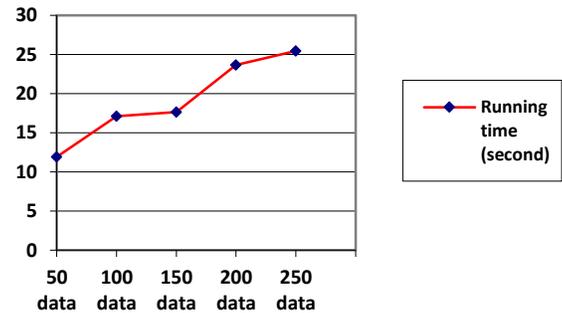
Data lowongan pekerjaan tersebut diambil dari website jobsdb dalam kurun waktu 2011 – 2013. Pengujian dilakukan pada perangkat keras dan eksekusi yang sama pada saat eksekusi sistem.

Pengujian yang dilakukan selanjutnya adalah pengujian terhadap data untuk mengetahui *running time* dari sistem yang telah dibuat. Tabel 17 berikut adalah hasil pengujian *running time* terhadap 250 data dalam bidang teknologi.

Tabel 17 *Running time* sistem

No	Data	<i>Running time</i>
1	50 data	11.90/s
2	100 data	17.13/s
3	150 data	17.67/s
4	200 data	23.68/s
5	250 data	25.46/s

Setelah melihat data pada tabel 17, maka data tersebut dapat dibuat grafik yang dapat dilihat pada gambar 6.



Gambar 6 Grafik *Running time* sistem

Dengan menganalisa data dan grafik yang telah dijabarkan sebelumnya, maka dapat ditarik kesimpulan bahwa proses dalam sistem rata – rata mengkonsumsi waktu sekitar 25 detik untuk menampilkan 250 data.

V. KESIMPULAN

Berdasarkan pembahasa implementasi dan pengujian yang telah dilakukan maka dapat diperoleh beberapa kesimpulan:

1. Pemanfaatan sistem rekomendasi dengan metode *decision tree* dengan algoritma *nearest neighbor* dapat di implementasikan pada pemberian rekomendasi pekerjaan kepada pelamar yang menginginkan pekerjaan sesuai dengan latar belakang pendidikan pelamar tersebut.
2. Dengan adanya sistem ini, maka pelamar dapat dengan mudah mengakses dan mendapatkan

informasi tentang lowongan pekerjaan pada perusahaan yang membutuhkan tenaga kerja atau karyawan baru.

3. Pelamar juga mendapatkan rekomendasi yang sesuai dengan kualifikasi pelamar tersebut dan *requirement* yang diinginkan oleh perusahaan tanpa harus mencari atau memilih pekerjaan secara satu persatu (manual), Sehingga pelamar dapat memilih dan bekerja di bidang pekerjaan yang sesuai dengan latar belakang pendidikan pelamar tersebut
4. Berdasarkan pengujian, rata – rata waktu yang diperlukan untuk menjalankan sistem rekomendasi tersebut adalah sekitar 25 detik.
5. Kualifikasi lowongan pekerjaan sangat berpengaruh terhadap hasil rekomendasi. Semakin banyak variasi kualifikasi lowongan pekerjaan maka akurasi dari rekomendasi pekerjaan semakin baik. Sehingga variasi kualifikasi tersebut dapat menghasilkan rekomendasi pekerjaan yang lebih akurat.

DAFTAR PUSTAKA

- [1]Ricci, F., 2002. Travel recommender system. *IEEE Intelligent Systems*
- [2]Adomavicius, G and Tuzhilin, A., 2005. *Toward the Next Generation of Recommender Systems: A survey of the state-of-the-art and possible extensions*. NYU
- [3] Supranto, J. 1998. *Teknik Pengambilan Keputusan*. Penerbit Rineka Cipt:Jakarta.
- [4]Kusrini dan Emha. 2009. *Algoritma Data Mining*. Edisi ke-1. Penerbit Andi Offset:Yogyakarta.
- [5] Sanjoyo, P. A. 2008. *Pembangunan Perangkat Lunak Sistem Rekomendasi Bursa Elektronik Telepon Genggam*. Institut Teknik Bandung
- Manikam R., M., dan Ali, U. 2010. *Kamus Elektronik Sistem Isyarat Bahasa Indonesia*. Universitas Mercu Buana: Jakarta.
- [6] Purwanto, A. 2009. *Metode Analisis Rekomendasi Pada Sistem Rekomendasi*. Universitas Pendidikan Indonesia Sholihin, A., Solihin, F., dan Rachman, F. H. 2013. Penerapan Modifikasi Metode *Enhanced Confix Stripping Stemmer* Pada Teks Berbahasa Madura. *Jurnal Sarjana Teknik Informatika* Vol. 1 No. 1. Hal 1-4.
- [7] Hidayat, M. K. P. 2007. *Model Sistem Informasi Toko Film Digital Dengan Recommender System*. Tesis, Institut Teknologi bandung.
- [8] Akhiro, R. 2008. *Studi Multi Criteria Decision Making (MCDM) untuk Recomender System Bursa Tenaga Kerja*. Tesis, Institut Teknologi Bandung.
- [9] Diahpangastuti, N. 2012. *Sistem Rekomendasi Bidang Minat Mahasiswa menggunakan Metode Association Rule dan Algoritma Apriori*. Skripsi, Institut Teknologi Sepuluh nopember
- [10]Setiawan, B. 2010. *Perancangan Sistem Pendukung Keputusan (SPK) untuk menentukan Kelaiklautan Kapal*.
- [11]Pudjiantoro,T. Renaldi, F. dan Teogunandi, A. 2011. *Penerapan Data Mining untuk menganalisa kemungkinan pengunduran diri calon mahasiswa baru*. Skripsi, Bali
- [12]Hayati, N. 2011. *Metode HYBRID (Content dan Collaborative based) Nearest Neighbour untuk sistem rekomendasi pariwisata*. Skripsi, Universitas Sumatera Utara, Medan.
- [13]Beynon-Davies. P. 2004. *Database System:Third Edition*. Palgrave, Basingstoke: United Kingdom.
- [14]Mahmood, T., Ricci, F. 2009. Improving recommender systems with adaptive conversational strategies. In: C. Cattuto, G. Ruffo, F. Menczer (eds.) *Hypertext*, pp. 73–82. ACM
- [15]Frias-Martinez, E., Magoulas, G., Chen, S. Y., and Macredie, R. 2006. Automated user modeling for personalized digital libraries, *International Journal of Information Management*, Vol. 26. pp.234–248.
- [16]Herlocker, J.L., and Konstan, J.A. 2001. content-Independent Task-Focused Recommendation, *IEEE Internet Computing*, Vol. 5, 2001, pp. 40-47
- [17]Mellville, P and Sindhvani, V. 2010. *Recomender Systems: Encyclopedia of machine learning ch:00338*. IBM research center
- [18]Utgoff, P. E. 1989. ID5: Incremental induction of decision trees, *Machine Learning*, Vol. 4, No. 2, pp. 161-186.
- [19]Xu, Y. Zhu, Q. Chen, Y and Pan J-S. 2013. An Improvement to the Nearest Neighbor Classifier and Face Recognition Experiments: *International Journal of Innovative Computing, Information and Control* Vol 9 no 2 pp 543-554.
- [20]Vaidehi, V and Vasuhi, S. 2008. *Person Authentication using Face Recognition, Proceedings of the world congress on engg and computer science* pp 1-6.