

ANALYTICAL HIERARCHY PROCCES (AHP) UNTUK MEMBANGUN MESIN PENCARI DATA LULUSAN PERGURUAN TINGGI BERDASARKAN KEBUTUHAN PENGGUNA LULUSAN

Agung Budi Prasetyo

Program Studi Teknik Informatika
STMIK AKAKOM Yogyakarta
Email: agung_bp@akakom.ac.id

Danny Kriestanto

Program Studi Teknik Informatika
STMIK AKAKOM Yogyakarta
Email: danny@akakom.ac.id

ABSTRAK

Telah menjadi tugas perguruan tinggi untuk membuat lulusan terserap dunia kerja. Keterserapan lulusan di dunia kerja akan tinggi apabila perguruan tinggi dapat melakukan *link & match* antara kebutuhan perusahaan dengan kompetensi yang dimiliki lulusan. *Link & match* yang baik dapat terjadi jika didukung oleh ketersediaan data yang akurat dan pengolahan data yang baik. Makalah ini melaporkan hasil penelitian pembuatan mesin pencari data lulusan yang dapat dimanfaatkan oleh pengguna lulusan untuk mencari lulusan suatu perguruan tinggi. Dengan metode *Analytical Hierarchy Procces* (AHP) kriteria calon pegawai yang ditetapkan pengguna lulusan akan diurutkan berdasarkan skala prioritas kemudian dicocokkan dengan kompetensi lulusan. Apabila ditemukan kompetensi lulusan yang sesuai atau yang hampir sesuai maka mesin pencari akan menampilkan lulusan yang dimaksud beserta biodatanya untuk selanjutnya dapat dihubungi pihak pengguna lulusan. Dengan 14 kriteria dan 57 sub kriteria yang tersedia pengguna lulusan dapat menemukan sendiri lulusan yang dicari sesuai dengan kriteria yang dikehendaknya.

Kata kunci: mesin pencari, data lulusan, AHP.

ABSTRACT

University have a responsibility to make graduates absorbed by world of work. Absorption of graduates would be higher if university can link and match the competencies required by the workforce seeker with the competencies of graduates. Link & match can be happened if it is supported by accurate data and good data processing. This paper reports the results of research about the graduates data search engine that can be used by workforce seeker to seek a university graduates. With Analytical Hierarchy Process (AHP), candidates criteria set by workforce seeker sorted by priority and then matched with the graduates competence. If found suitable graduates competence or close the search engine will display the biographical data of graduates to the next can be contacted by workforce seeker. With 14 criteria and 57 sub-criteria are provided, workforce seekers themselves may find graduates that are required, in accordance with criteria that pleases.

Keywords: search engine, graduates data, analytical hierarchy process.

1. PENDAHULUAN

Kebutuhan terhadap data dan informasi yang mutakhir, terbaharui, dan sinambung mengenai hubungan antara dunia pendidikan tinggi dan dunia kerja menjadi hal yang penting dan mendesak [1]. Kendala yang dihadapi adalah belum banyak tersedianya *tool* yang dapat membantu pengguna lulusan menemukan lulusan yang tepat sesuai kriteria dibutuhkan. Diperlukan metode yang dapat melakukan *link & match* antara kompetensi lulusan dan kriteria yang ditentukan pengguna lulusan.

Metode *Analytical Hierarchy Process* (AHP) dapat digunakan untuk menyelesaikan masalah *Multi Kriteria Decision Making*, yaitu mengevaluasi alternatif-alternatif terhadap sekumpulan atribut atau kriteria, dengan setiap atribut saling tidak bergantung satu dengan lainnya [2].

AHP merupakan suatu teori pengukuran yang digunakan untuk menderivasikan skala rasio baik dari perbandingan-perbandingan berpasangan diskrit maupun kontinu [3]. Dalam menggunakan AHP diperlukan suatu hirarki untuk mendefenisikan masalah dan perbandingan berpasangan untuk menentukan hubungan dalam struktur tersebut. Struktur hirarki digambarkan dalam suatu diagram pohon yang berisi goal (tujuan masalah yang akan dicari solusinya), kriteria, subkriteria dan alternatif .

Sinaga [4] membuktikan AHP mampu membantu lulusan perguruan tinggi dalam memilih perusahaan mana yang akan dimasuki. Dengan 4 buah kriteria (Gaji, Jenjang karir, Fasilitas dan Penempatan) AHP mampu memberikan hasil berupa daftar perusahaan BUMN yang diurutkan berdasarkan kriteria yang dipilih lulusan.

Dengan sasaran pemakai pihak pengguna lulusan makalah ini memaparkan bagaimana pembuatan mesin pencari menggunakan AHP untuk menemukan lulusan yang memiliki kompetensi yang sesuai dengan harapan pengguna lulusan.

2. METODOLOGI PENELITIAN

Penelitian dilakukan dengan metodologi sebagai berikut: (1) identifikasi masalah (2) telaah algoritma AHP (3) desain dan pembuatan aplikasi (4) pengujian aplikasi.

2.1 Identifikasi Masalah

Studi kasus dilakukan pada STMIK AKAKOM Yogyakarta, di mana meskipun telah memiliki data lulusan, khususnya berupa data kompetensi, pengelola pusat karir belum dapat mengolahnya menjadi rekomendasi lulusan apabila calon pengguna lulusan memerlukan lulusan dengan spesifikasi tertentu.

Spesifikasi yang dimaksud meliputi 14 aspek kriteria yang sering dicari oleh pengguna lulusan yaitu: (1) Jenjang, (2) Menguasai Bahasa Pemrograman, (3) Menguasai Sistem Operasi, (4) Menguasai Program Aplikasi, (5) Memiliki Kompetensi tertentu di bidang IT, (6) menyukai bidang IT, (7) Rencana untuk berkarir di bidang tertentu (IT), (8) Bersedia direkrut kerja oleh perusahaan IT, (9) memiliki pengalaman kerja, (10) Memiliki pengalaman dalam tim, (11) memiliki pengalaman dalam proyek, (12) Memiliki sertifikasi tertentu, (13) Memiliki pengalaman bisnis dan (14) Menguasai bahasa asing tertentu.

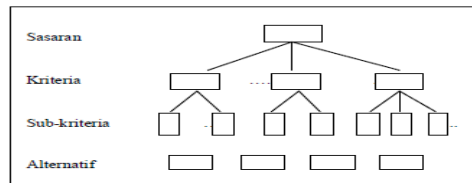
Adapun untuk beberapa kriteria terdapat sub kriteria yaitu kriteria jenjang memiliki 2 sub kriteria yaitu Strata 1 (J1) dan Diploma 3 (J2). Kriteria menguasai bahasa pemrograman memiliki 9 sub kriteria yaitu bahasa pemrograman Java (PMR1), bahasa pemrograman C/C++/C# (PMR2), bahasa pemrograman Pascal/Delphi (PMR3), bahasa pemrograman Basic/ VB (PMR4), bahasa pemrograman PHP (PMR5), bahasa pemrograman Asp.net (PMR6), bahasa pemrograman Python/ Pearl (PMR7), bahasa pemrograman Prolog (PMR8), dan bahasa Asembler (PMR9). Kriteria menguasai sistem operasi memiliki 6 sub kriteria yaitu sistem operasi Windows (SO1), sistem operasi Linux (SO2), sistem operasi Sun (SO3), sistem operasi Unix (SO4), sistem operasi Android (SO5), dan sistem operasi Windows Phone (SO6). Kriteria menguasai program aplikasi memiliki 8 sub kriteria yaitu Aplikasi Perkantoran (Wordprocessing/ Spreadsheet/ Presentation/ dll) (PA1), Aplikasi Grafis dan Desain (PA2), Aplikasi Multimedia (audio/ video) (PA3), Aplikasi Internet (Email/ Chat/ Browsing/ Map/ dll) (PA4), Aplikasi Mobile (Android/ Windows Phone/ dll) (PA5), Aplikasi Keamanan Jaringan (PA6), Aplikasi Database (MySQL/ Oracle/ Paradow/ dll) (PA7), Aplikasi Keuangan dan Akuntansi (PA8).

Kriteria menguasai kompetensi IT tertentu memiliki 14 sub kriteria yaitu Komunikasi Data dan Jaringan Komputer (KIT1), Keamanan Jaringan (KIT2), Sistem Kendali dan Robotika (KIT3), Programming (KIT4), Sistem Cerdas (KIT5), Komputasi Cloud (KIT6), Komputasi Mobile (KIT7), Web, Framework, GIS (KIT8), Data Warehouse dan Data Mining (KIT9), Basis Data dan Sistem Informasi (KIT10), Multimedia dan Pengolahan Citra (KIT11), Games (KIT12), Hardware, Instalasi dan Maintenance (KIT13), dan Komputer Akuntansi (KIT14). Kriteria rencana untuk berkarir di bidang IT tertentu memiliki 11 sub kriteria yaitu sebagai Programmer (KR1), sebagai Sistem Analis (KR2), sebagai Quality Assurance (KR3), sebagai Operator (KR4), sebagai Tester (KR5), sebagai Implementor (KR6), sebagai Operating Maintenance (KR7), sebagai Web Developer (KR8), sebagai Desain Grafis (KR9), sebagai IT Support (KR10), dan karir lainnya (KR11). Kriteria menguasai bahasa asing memiliki 7 sub kriteria yaitu bahasa Inggris (BA1), bahasa Prancis (BA2), bahasa Portugues (BA3), bahasa Arab (BA4), bahasa Jepang (BA5), bahasa Mandarin (BA6), dan bahasa Korea (BA7).

Adapun data yang berwujud kompetensi lulusan yang dimiliki oleh pusat karir adalah berupa nilai bobot masing-masing lulusan terhadap kriteria-kriteria yang telah disebutkan di atas. Untuk dapat memberikan rekomendasi nama lulusan kepada calon pengguna lulusan diperlukan sebuah aplikasi yang dapat mengolah data-data tersebut menjadi sebuah rekomendasi nama-nama lulusan yang sesuai dengan spesifikasi yang diminta oleh pengguna lulusan.

2.2 Telaah Algoritma AHP

Menurut Saaty pada prinsipnya AHP bekerja dengan menyederhanakan suatu persoalan kompleks yang tidak terstruktur menjadi bagian-bagian lebih terstruktur dan menata dalam suatu hirarki [5]. Secara grafis, persoalan keputusan dengan AHP dapat dikonstruksikan sebagai diagram bertingkat, dimulai dengan sasaran, kemudian kriteria level pertama, sub kriteria dan akhirnya alternatif (Gambar 1).



Gambar 1. Struktur hirarki dalam AHP [5]

Sinaga [4] dan Indriyati [5] menguraikan langkah-langkah kerja AHP sebagai berikut:

1. Menentukan tujuan, kriteria dan alternatif keputusan
2. Menyusun hirarki untuk berbagai kriteria dan alternatif keputusan
3. Membuat matriks perbandingan berpasangan A berdasarkan masing-masing kriteria dan sub kriteria. Matriks perbandingan berpasangan (*pairwise comparisons matrix*) dibuat berdasarkan masukan dari pengguna (*comparative judgement*) dengan menilai tingkat kepentingan suatu kriteria dibandingkan kriteria lainnya. Pada *comparative judgement* angka perbandingan berupa skala 1 sampai 9 di mana skala 1 yang menunjukkan tingkat yang paling rendah (*equal importance*) sampai dengan skala 9 yang menunjukkan tingkatan yang paling tinggi (*extreme importance*). Nilai dan definisi dari skala perbandingan Saaty pada tabel 2.

Tabel 2. Skala Penilaian Perbandingan Berpasangan [5]

Nilai	Definisi
1	Kedua elemen sama penting
3	Elemen yang satu sedikit lebih penting dari lainnya
5	Elemen yang satu jelas lebih penting dari lainnya
7	Elemen yang satu sangat jelas lebih penting dari lainnya
9	Mutlak lebih penting dari lainnya
2,4,6,8	Apabila ragu-ragu antara dua nilai yang berdekatan

Skala 1 memiliki arti kriteria yang satu sama penting dibanding kriteria lainnya, skala 3 memiliki arti kriteria yang satu *sedikit lebih penting* dibanding kriteria lainnya, skala 5 memiliki arti kriteria yang satu *jelas lebih penting* dibanding kriteria lainnya, Skala 7 memiliki arti kriteria yang satu *sangat jelas lebih penting* dibanding kriteria lainnya, Skala 9 memiliki arti kriteria yang satu *mutlak lebih penting* dibanding kriteria lainnya. Untuk polaritas kepentingan yang berkebalikan untuk menjelaskan ‘kurang penting dibanding’ digunakan 1/ skala (satu per skala).

4. Menentukan vektor bobot dengan cara :
 - a. Normalisasi setiap kolom j dalam matriks A sedemikian sehingga :

$$\sum_i a_{ij} = 1 \tag{1}$$

a_{ij} adalah elemen matriks A

- b. Untuk setiap baris i dalam matriks tersebut hitung nilai rata-ratanya (vektor eigen) :

$$w_i = \frac{1}{n} \sum_j a_{ij} \tag{2}$$

w_i adalah bobot ke i dari vektor bobot

5. Pengujian konsistensi matriks perbandingan dengan cara :

Jika A adalah matriks perbandingan berpasangan dan W adalah vektor bobot, maka konsistensi dari A dapat diuji dengan cara :

- a. Hitung :

$$t = \frac{1}{n} \sum_1^n \left(\frac{\text{elemen ke } i \text{ pada } (A)(W^T)}{\text{elemen ke } i \text{ pada } W^T} \right) \tag{3}$$

- b. Hitung indeks konsistensi :

$$CI = \frac{t-n}{n-1} \tag{4}$$

- c. Periksa rasio konsistensi :

$$CR = \frac{CI}{RI} \tag{5}$$

Jika $CR \leq 0,1$ maka A konsisten

Jika $CR > 0,1$ maka A tidak konsisten

Nilai RI = nilai indeks random berdasarkan tabel 1

Tabel 1. Random Indeks [6]

<i>N</i>	<i>RI</i>	<i>N</i>	<i>RI</i>
1	0	9	1.45
2	0	10	1.49
3	0.58	11	1.51
4	0.9	12	1.48
5	1.12	13	1.56
6	1.24	14	1.57
7	1.32	15	1.59
8	1.41		

6. Menentukan bobot global apabila terdapat sub kriteria.
7. Menentukan perankingan alternatif.
 Jika ada n kriteria dan m alternatif, maka langkah-langkah untuk menentukan perankingan alternatif adalah :
 - a. Untuk setiap kriteria i , tetapkan matriks perbandingan berpasangan A_i untuk m alternatif.
 - b. Tentukan vektor bobot untuk setiap A_i yang mempresentasikan bobot relatif dari setiap alternatif ke j pada kriteria ke i (s_{ij}).
 - c. Hitung total skor :

$$S_j = \sum_i (S_{ij})(W_i) \tag{6}$$

2.3 Desain Dan Pembuatan Aplikasi

Proses pembuatan aplikasi mencakup 2 aspek yaitu : (1) pembuatan web dan (2) penerapan AHP. Dalam pembuatan web, aplikasi mesin pencari memiliki 2 entitas berbeda yaitu pengguna lulusan dan lulusan (Gambar 2). Pengguna lulusan memberikan masukan data berupa prioritas kriteria yang diinginkan sedangkan lulusan memberikan masukan berupa skor bobot dirinya terhadap setiap kriteria/ sub kriteria yang ada.

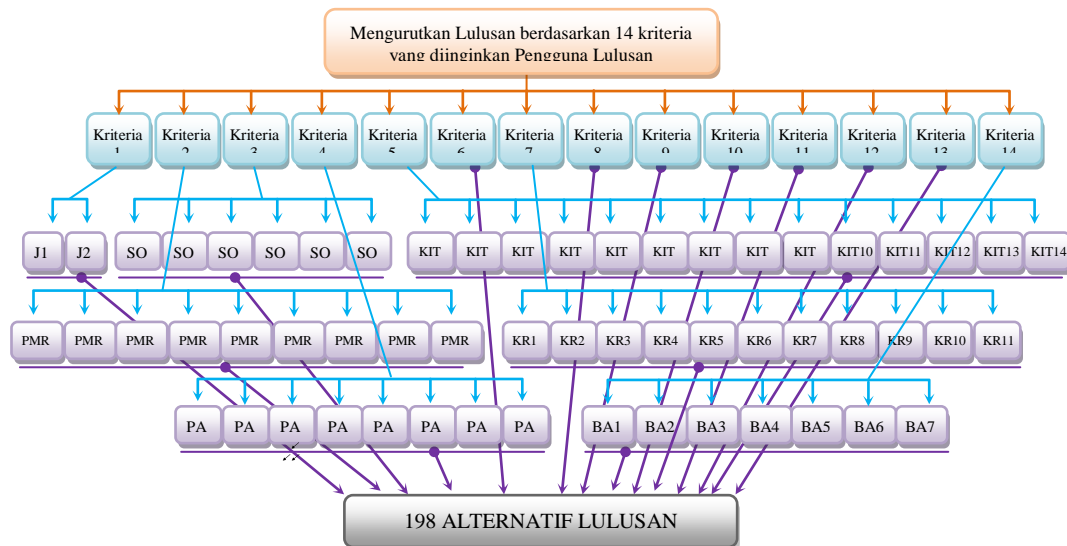


Gambar 2. Diagram Konteks Aplikasi

Penerapan AHP pada pembuatan mesin pencari dipaparkan sebagai berikut :

2.3.1 Membuat Struktur Hirarki

Membuat struktur hirarki diawali dengan menentukan tujuan umum yang hendak dicapai yaitu menemukan sebanyak M_{hasil} orang lulusan STMIK AKAKOM yang memiliki kompetensi sesuai dengan kriteria yang ditetapkan pengguna lulusan. Kriteria yang disediakan oleh aplikasi sebanyak 14 kriteria dan 57 sub kriteria. Alternatif lulusan yang tersedia sebanyak $M=198$ lulusan.



Gambar 3. Struktur Hirarki Pencarian Lulusan

Pembuatan struktur hirarki (*decomposition*) disusun dalam 4 level (Gambar 3). Level pertama merupakan sasaran yang ingin dicapai. Level kedua dan ketiga menunjukkan 14 aspek kriteria beserta sub kriterianya sebagaimana telah dipaparkan pada identifikasi masalah.

Alternatif pada level keempat adalah $M = 198$ orang lulusan STMIK AKAKOM yang akan diambil sebanyak M_{Hasil} orang berdasarkan kebutuhan kuota yang dikehendaki pengguna lulusan. (sumber: data lulusan STMIK AKAKOM periode II 2014/2015).

2.3.2 Membentuk Matriks Perbandingan Berpasangan

Matriks perbandingan berpasangan (*pairwise comparisons matrix*) dibuat berdasarkan masukan dari pengguna lulusan (*comparative judgement*) dengan menilai tingkat kepentingan suatu kriteria dibandingkan kriteria lainnya. Pada *comparative judgement* angka perbandingan berupa skala 1 sampai 9 di mana skala 1 yang menunjukkan tingkat yang paling rendah (*equal importance*) sampai dengan skala 9 yang menunjukkan tingkatan yang paling tinggi (*extreme importance*).

Form comparative judgement dibuat dengan kode sebagaimana tersaji dalam script berikut :

```

<?php
$N = 14;
$R = 1;
echo"<form action='settingKritRespon.php' method='get' >";
echo"<table>";
//----- bagian kepala tabel -----
echo "<tr>";
for ($kolom=1; $kolom<=$N-1; $kolom++)
    echo "<td> Kriteria $kolom<br>terhadap </td>";
echo "</tr>";
//----- bagian isi tabel -----
echo "<tr>";
for ($baris=1; $baris<=$N-1; $baris++)
{ echo "<td>";
    for ($kolom=$baris+1; $kolom<=$N; $kolom++)
    {
        echo"<input type='hidden' name='_nb[$R]' value='$nnomorBaris[$R]'\>";
        echo " <input type='hidden' name = '_sby[$R]' value='$ssubyek[$R]'\>";
        echo " <select name = '_pred[$R]'\>";
        for ($s=9; $s>=2; $s--)
            echo " <option value = $s> $s ";
        echo " <option value = '$ppredikat[$R]' selected> 1";
        for ($s=2; $s<=9; $s++)
            echo "<option value = 1/$s> 1/$s";
        echo " </select> ";
        echo " <input type='hidden' name = '_oby[$R]' value='$soobyek[$R]'\>";
        echo "<br>Kriteria $kolom<br><br>";
        $R++;
    }
}
echo "</td>";

```

```
}  
echo "</tr>";  
echo "</table>";  
echo "<input type='submit' value='CEK KONSISTENSI'>";  
echo "</form>";  
?>
```

Setiap nilai yang diberikan pengguna lulusan disimpan pada variabel larik yang terdiri dari variabel `_nb[$R]`, `_sby[$R]`, `_pred[$R]`, dan `_oby[$R]`. Variabel `_nb[$R]` digunakan untuk mencatat nomor baris dari matriks perbandingan berpasangan yang akan dibuat, `_sby[$R]` digunakan untuk mencatat kriteria yang akan dibandingkan, variabel `_oby[$R]` digunakan untuk mencatat kriteria lain yang menjadi obyek perbandingan, sedangkan variabel `_pred[$R]` digunakan mencatat skala antara 1 – 9 (dan 1 – 1/9 untuk polaritas yang berbalikan) antara kriteria dan kriteria lain yang menjadi obyek perbandingnya. Dari *comparative judgement* selanjutnya dibentuk matriks perbandingan berpasangan melalui kode sebagaimana tersaji dalam script program berikut :

```
<?php  
$nomorBaris = $_GET['_nb'];  
$subyek = $_GET['_sby'];  
$predikat = $_GET['_pred'];  
$obyek = $_GET['_oby'];  
  
//matriks perbandingan berpasangan  
$record=1;  
for ($bar=1; $bar<=$N; $bar++)  
{ for ($kol=$bar; $kol<=$N; $kol++)  
  { if ($bar==$kol)  
    { $mat[$bar][$kol] = 1;  
    }  
    else  
    { $mat[$bar][$kol] = $predikat[$record];  
      $mat[$kol][$bar] = 1/$predikat[$record];  
      $record++;  
    }  
  }  
}  
echo "<table border=1>";  
for ($kol=1; $kol<=$N; $kol++)  
{ $sumKol[$kol] = 0;  
}  
for ($bar=1; $bar<=$N; $bar++)  
{ echo "<tr>";  
  echo "<td> baris $bar </td>";  
  for ($kol=1; $kol<=$N; $kol++)  
  { echo "<td> $mat[$bar][$kol] </td>";  
    $sumKol[$kol] = $sumKol[$kol] + $mat[$bar][$kol];  
  }  
  echo "</tr>";  
}  
for ($kol=1; $kol<=$N; $kol++)  
{ echo "<td> $sumKol[$kol] </td>";  
}  
echo "</tr>";  
echo "</table>";  
?>
```

Jumlah dari masing-masing kolom matriks perbandingan berpasangan dihitung dengan cara menjumlahkan semua nilai matriks perbandingan berpasangan pada kolom tersebut mulai dari baris pertama hingga baris ke `$N`, di mana `$N` adalah banyaknya kriteria. Hasil penjumlahan setiap kolom selanjutnya akan digunakan untuk membuat matriks normal.

2.3.3 Menormalkan Data

Menormalkan data dilakukan dengan membuat matriks normal berdasarkan persamaan (1) dengan cara menjumlahkan nilai – nilai dari setiap kolom pada matriks perbandingan berpasangan, dilanjutkan dengan membuat matriks baru (matriks normal) di mana setiap elemennya merupakan hasil bagi setiap elemen matriks perbandingan berpasangan dibagi dengan jumlah kolom matriks perbandingan berpasangan. Untuk membuat matriks normal sebagaimana dimaksud di atas dibuatlah kode sebagaimana tersaji dalam script berikut :

```
echo "<table border=1>";
echo "<tr>";
for ($kol=1; $kol<=$N; $kol++)
{ echo "<td> kol $kol </td>";
}
echo "<td>EIGEN</td>";
echo "</tr>";
for ($bar=1; $bar<=$N; $bar++)
{ echo "<tr>";
echo "<td> baris $bar </td>";
for ($kol=1; $kol<=$N; $kol++)
{
$matNormal[$bar][$kol]=$mat[$bar][$kol]/$sumKol[$kol];
echo "<td> $matNormal[$bar][$kol] </td>";
}
echo "</tr>";
}
echo "</table>";
```

2.3.4 Menghitung Vektor Eigen

Menghitung vektor eigen dilakukan berdasarkan persamaan (2) dengan cara menjumlahkan nilai – nilai dari setiap baris pada matriks normal dan membaginya dengan jumlah elemen untuk mendapatkan nilai rata – rata. Dalam membuat vektor eigen dibuatlah kode sebagaimana tersaji dalam script berikut :

```
for ($bar=1; $bar<=$N; $bar++)
{ echo "<tr>";
echo "<td> baris $bar </td>";
$eigen[$bar] = 0;
for ($kol=1; $kol<=$N; $kol++)
{
$matNormal[$bar][$kol]=$mat[$bar][$kol]/$sumKol[$kol];
echo "<td> echo $matNormal[$bar][$kol] </td>";
$eigen[$bar]= $eigen[$bar] + $matNormal[$bar][$kol];
}
//menghitung eigen
$eigen[$bar] = $eigen[$bar] / $N;
echo "<td> $eigen[$bar] </td>";
echo "</tr>";
}
```

Vektor Eigen diperoleh dengan membagi $\$eigen[\$bar]$ dengan banyaknya kriteria yaitu N.

2.3.5 Pengujian Konsistensi

Pengujian konsistensi dilakukan dengan tahapan sebagai berikut :

1. Menghitung *lamda* maksimum dilakukan berdasarkan persamaan (3). Dalam penghitungan *lamda* maksimum dibuatlah kode sebagaimana tersaji dalam script berikut :

```
//menghitung lamda max
//-----
$lamdaMax=0;
for ($i=1; $i<=$N; $i++)
{ $lamdaMax = $lamdaMax + ($sumKol[$i] * $eigen[$i]);
}
echo "Lamda Max = $lamdaMax <br/>";
```

Variabel $\$sumKol_k$ adalah penyimpanan hasil penjumlahan dari masing-masing kolom pada matriks perbandingan berpasangan, sedangkan $\$eigen_k$ adalah vektor eigen yang telah diperoleh pada langkah sebelumnya.

2. Hitung Indeks Konsistensi/ *Consistency Index* (CI) berdasarkan persamaan (4) yaitu $CI = (\lambda_{maks} - n) / (n - 1)$ di mana $n =$ banyak kriteria.

```
//menghitung CI (Consistensi Index)
//-----
$CI = ( $lamdaMax - $N ) / ( $N - 1 );
echo "CI = $CI <br/>";
```

3. Hitung Rasio Konsistensi/*Consistency Ratio* (CR) berdasarkan persamaan (5) yaitu $CR = CI / RI$ di mana CI : Indeks Konsistensi/ *Consistency Index* dan RI = Indeks Random/ *Random Index* (tabel 1)

```
//menghitung CR (rumus CR = CI/RI )
//-----
if (($N==1) || ($N==2))
{
    $RI = 0;
    $CR = 0;
}
else
{
    if ($N==3) $RI = 0.58;
    else if ($N==4) $RI = 0.9;
    else if ($N==5) $RI = 1.120;
    else if ($N==6) $RI = 1.240;
    else if ($N==7) $RI = 1.320;
    else if ($N==8) $RI = 1.410;
    else if ($N==9) $RI = 1.450;
    else if ($N==10) $RI = 1.490;
    else if ($N==11) $RI = 1.510;
    else if ($N==12) $RI = 1.480;
    else if ($N==13) $RI = 1.560;
    else if ($N==14) $RI = 1.570;
    else if ($N==15) $RI = 1.590;
    $CR = $CI / $RI;
    echo "CR = $CR <br/>";
}
```

4. Memeriksa konsistensi hirarki. Jika nilainya lebih dari 10%, maka penilaian *comparative judgement* harus diperbaiki. Namun jika rasio konsistensi kurang atau sama dengan 0,1 , maka hasil perbandingan bisa dinyatakan konsisten sebagaimana dijelaskan dalam persamaan (5).

```
if($CR<0.1)
{
    echo " CR < 0.1,
        Konsistensi Diterima, lanjutkan <br/>";
}
else
{
    echo " CR < 0.1, Konsistensi DiTolak,
        perbaiki kembali bobot kriteria <br/>";
}
echo"<input type='button' value='Kembali' onClick='self.history.back()'>";
```

2.3.6 Pembobotan Global dan Lokal atas Sub Kriteria

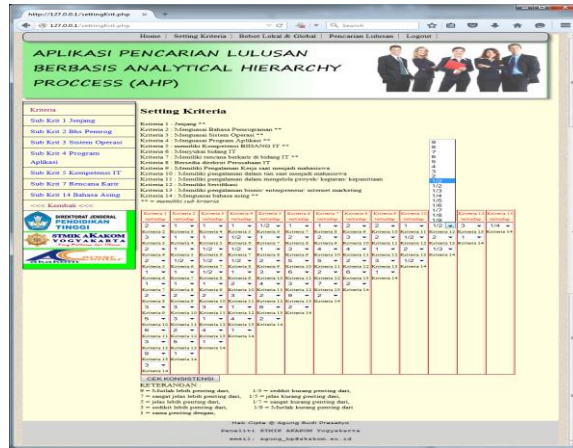
Pembobotan global dilakukan dengan cara mengalikan seluruh bobot lokal (vektor eigen) masing-masing sub kriteria dengan vektor eigen kriteria [5]. Dengan bobot global yang diperoleh selanjutnya akan dilakukan pengurutan kriteria/ sub kriteria secara *descending*.

2.3.7 Perangkingan Alternatif

Bobot dari setiap alternatif (lulusan) untuk setiap kriteria dan sub kriteria diperoleh dari masing-masing lulusan menggunakan skala bobot dengan cara *entry* yang kemudian disimpan dalam bentuk skala CF (*Certainty Factor*) 0 hingga 1. Untuk memperoleh *ranking* atas setiap alternatif dilakukan perkalian antara bobot alternatif dengan bobot kriteria yang dihitung menggunakan persamaan (6). Alternatif yang telah diurutkan selanjutnya dipotong sebanyak kuota lowongan kerja yang diminta oleh pengguna lulusan (M_{Hasil}).

3. HASIL DAN PEMBAHASAN

Gambar 4 memperlihatkan tampilan masukan skala perbandingan berpasangan (*comparative judgement*) oleh pengguna lulusan antara satu kriteria terhadap kriteria yang lain yang meliputi 14 kriteria dan 57 sub kriteria. Di sinilah pengguna lulusan menentukan sendiri skala prioritas atas kriteria yang dikehendaki.



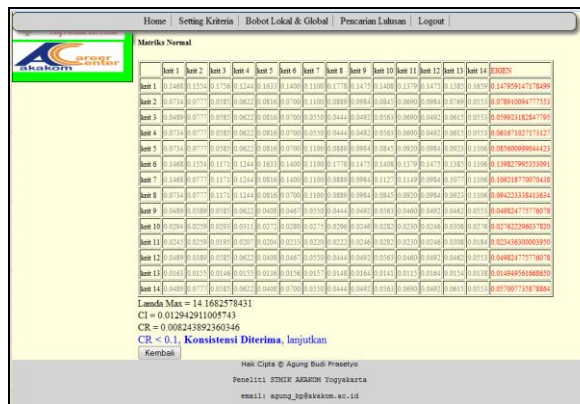
Gambar 4. Masukan skala perbandingan berpasangan oleh pengguna lulusan

Dari *comparative judgement* selanjutnya dibentuk matriks perbandingan berpasangan untuk semua kriteria dan sub kriteria (Gambar 5), demikian pula untuk matriks normal yang menghasilkan vektor eigen (Gambar 6).



Gambar 5. Matriks perbandingan berpasangan

Matriks normal dan vektor eigen (gambar 6) diperoleh setelah dilakukan pembuatan matriks perbandingan berpasangan.



Gambar 6. Matriks Normal dan Vektor Eigen

Pengujian terhadap *kesahihan* hasil perhitungan aplikasi telah dilakukan dengan menggunakan uji coba pencarian lulusan berdasarkan kriteria tertentu. Sebagai pembanding dari pengujian tersebut dilakukan pula pencarian lulusan berdasarkan kriteria yang sama yang dilakukan menggunakan proses penghitungan manual.

Komparasi hasil perhitungan aplikasi atas bobot untuk semua kriteria dan sub kriteria maupun hasil perhitungan manual disajikan dalam tabel 3 dan tabel 4.

Tabel 3. Vektor Eigen untuk Kriteria 1 sampai 14

<i>Kode</i>	<i>Kriteria</i>	<i>Nilai Eigen (hasil aplikasi)</i>	<i>Nilai Eigen (hasil manual)</i>
Krit 1	Jenjang	0.147959147178499	0.147959147639531
Krit 2	Bahasa Pemrograman yang dikuasai	0.078910094777553	0.078910095044474
Krit 3	Sistem Operasi yang dikuasai	0.059923182847795	0.059923183114717
Krit 4	Kompetensi dalam menggunakan Program Aplikasi	0.061671027173127	0.061671027440049
Krit 5	Kompetensi BIDANG IT yang telah dimiliki	0.085600999044423	0.085600999351778
Krit 6	Menyukai bidang IT?	0.139827995353091	0.139827995886934
Krit 7	Ada rencana berkarir di bidang IT	0.109218770070438	0.109218770491037
Krit 8	Bersedia direkrut bekerja oleh Perusahaan IT	0.094223338413634	0.094223338720989
Krit 9	Memiliki Pengalaman Kerja selama menjadi mahasiswa, volunteer, relawan, magang, dsb	0.049824775776078	0.049824775929756
Krit 10	Memiliki pengalaman dalam tim/ organisasi/ komunitas selama menjadi mahasiswa	0.027622296037820	0.027622296114659
Krit 11	Memiliki pengalaman dalam mengelola proyek/ kegiatan/ kepanitiaan	0.023436300003950	0.023436301409579
Krit 12	Memiliki Sertifikasi misalnya: DAT, Oracle, CICSO, JENI, dan lain-lain	0.049824775776078	0.049824775929756
Krit 13	Memiliki pengalaman bisnis/ entrepreneur/ internet marketing (offline maupun online)	0.014949561668650	0.014949561707069
Krit 14	Menguasai bahasa asing	0.057007735878864	0.057007731219673
CR =		0.008243892360345	0.008243890760546

Tabel 4. Vektor Eigen untuk 57 Sub Kriteria

		<i>Bobot Lokal</i>	<i>Eigen Kriteria</i>	<i>Bobot Global (hasil aplikasi)</i>	<i>Bobot Global (hasil manual)</i>
Krit 1	Jenjang				
(J1)	S1	0.75	0.15	0.1109693603835000	0.1109693607296490
(J2)	D3	0.25	0.15	0.0369897867945000	0.0369897869098829
CR =				0	0
Krit 2	Bahasa Pemrograman yang dikuasai				
(PMR1)	Java	0.21	0.08	0.0166724935890020	0.0166724936453693
(PMR2)	C/ C++/ C#	0.16	0.08	0.0128634481578250	0.0128634482013521
(PMR3)	Pascal/ Delphi	0.09	0.08	0.0067560254352360	0.0067560254580830
(PMR4)	Basic/ VB	0.10	0.08	0.0082675967473570	0.0082675967752982
(PMR5)	Php	0.12	0.08	0.0095915276407470	0.0095915276731809
(PMR6)	Asp.net	0.17	0.08	0.0132226834366570	0.0132226834814120
(PMR7)	Python/ Pearl	0.07	0.08	0.0053591858059760	0.0053591858241034
(PMR8)	Prolog	0.04	0.08	0.0035027929083000	0.0035027929201485
(PMR9)	Asembler	0.03	0.08	0.0026743410564850	0.0026743410655268
CR =				0.0444425669	0.044442567
Krit 3	Sistem Operasi yang dikuasai				
(SO1)	Windows	0.35	0.06	0.0207847264098240	0.0207847276715121
(SO2)	Linux	0.25	0.06	0.0147563020625240	0.0147563031022708
(SO3)	Sun	0.03	0.06	0.0016730470367410	0.0016730476415655
(SO4)	Unix	0.08	0.06	0.0046094389931000	0.0046094389453021
(SO5)	Android	0.18	0.06	0.0109039730252270	0.0109039695897769
(SO6)	Windows Phone	0.12	0.06	0.0071956953204030	0.0071956961642892
CR =				0.0967960896	0.096796044
Krit 4	Kompetensi dalam menggunakan Program Aplikasi				
(PA1)	Aplikasi Perkantoran	0.13	0.06	0.0078452256782050	0.0078452257174441
(PA2)	Aplikasi Grafis Desain	0.08	0.06	0.0050928639265690	0.0050928645713072
(PA3)	Aplikasi Multimedia	0.03	0.06	0.0019216432652400	0.0019216430032287
(PA4)	Aplikasi Internet	0.15	0.06	0.0095185417942480	0.0095185417893542
(PA5)	Aplikasi Mobile	0.10	0.06	0.0064017359334430	0.0064017352710267
(PA6)	Aplikasi KeamananJar	0.19	0.06	0.0117885576964470	0.0117885576904664

		<i>Bobot Lokal</i>	<i>Eigen Kriteria</i>	<i>Bobot Global (hasil aplikasi)</i>	<i>Bobot Global (hasil manual)</i>
(PA7)	Aplikasi Database	0.24	0.06	0.0150448874347370	0.0150448888132929
(PA8)	Aplikasi Keu.dan Akt	0.07	0.06	0.0040575714441790	0.0040575705839285
CR =				0.0297164256	0.029716398
Krit 5	Kompetensi BIDANG IT yang telah dimiliki				
(KIT1)	Komunikasi Data dan Jaringan	0.10	0.09	0.0086202805547450	0.0086202811395175
(KIT2)	Keamanan Jaringan	0.05	0.09	0.0044912942619230	0.0044912955222779
(KIT3)	Sist Kendali, Robotika	0.01	0.09	0.0012179975532420	0.0012179980838027
(KIT4)	Programming	0.17	0.09	0.0148040893446810	0.0148040882869071
(KIT5)	Sistem Cerdas	0.09	0.09	0.0073488857194290	0.0073488862129081
(KIT6)	Komputasi Cloud	0.07	0.09	0.0063247642687920	0.0063247647608814
(KIT7)	Komputasi Mobile	0.04	0.09	0.0035955798283010	0.0035955799853728
(KIT8)	Web, Framework, GIS	0.13	0.09	0.0109611250259780	0.0109611233996771
(KIT9)	Data Warehouse dan Data Mining	0.04	0.09	0.0030707830850590	0.0030707830043178
(KIT10)	Basis Data, Sistem Inf	0.15	0.09	0.0128222714371850	0.0128222704700566
(KIT11)	Multimedia dan Pengolahan Citra	0.03	0.09	0.0025738213105690	0.0025738214694441
(KIT12)	Games	0.03	0.09	0.0021565189866000	0.0021565192335580
(KIT13)	Hardware, Instalasi dan Maintenance	0.06	0.09	0.0048113378815030	0.0048113378791161
(KIT14)	Komputer Akuntansi	0.03	0.09	0.0028022497864030	0.0028022499039404
CR =				0.0469082527	0.046908251
Krit 6	Menyukai bidang IT?		0.14	0.13982799535309	0.1398279958869340
Krit 7	Ada rencana berkarir di bidang IT				
(KR1)	Programmer	0.16	0.11	0.0178248657161650	0.0178248663264495
(KR2)	Sistem Analis	0.10	0.11	0.0112173720446490	0.0112173722826364
(KR3)	Quality Assurance	0.08	0.11	0.0084492078488290	0.0084492081127765
(KR4)	Operator	0.06	0.11	0.0069725962412760	0.0069725964995321
(KR5)	Tester	0.07	0.11	0.0073403362078410	0.0073403364675154
(KR6)	Implementor	0.06	0.11	0.0070910826100520	0.0070910837963015
(KR7)	Op. Maintenance	0.11	0.11	0.0114884591943040	0.0114884596933275
(KR8)	Web Developer	0.16	0.11	0.0174571275016870	0.0174571263584662
(KR9)	Desain Grafis	0.04	0.11	0.0047243536180200	0.0047243540865302
(KR10)	IT Support	0.12	0.11	0.0128079995348800	0.0128079972255028
(KR11)	Other	0.04	0.11	0.0038453695522320	0.0038453696419986
CR =				0.0208069430	0.020806931
Krit 8	Bersedia direkrut Perusahaan IT		0.09	0.094223338413634	0.0942233387209892
Krit 9	Memiliki Pengalaman Kerja selama menjadi mahasiswa		0.05	0.049824775776078	0.0498247759297557
Krit 10	Memiliki pengalaman tim/ organisasi/ komunitas selama mhs.		0.03	0.02762229603782	0.0276222961146590
Krit 11	Memiliki pengalaman mengelola proyek/ kegiatan/ kepanitiaan		0.02	0.02343630000395	0.0234363014095792
Krit 12	Memiliki Sertifikasi misalnya: DAT, Oracle, CICSO, JENI, dsb		0.05	0.049824775776078	0.0498247759297557
Krit 13	Memiliki pengalaman bisnis/ enterpreneur/ internet marketing		0.01	0.01494956166865	0.0149495617070691
Krit 14	Menguasai bahasa asing				
(BA1)	Inggris	0.32	0.06	0.0184870102644060	0.0184870096593669
(BA2)	Prancis	0.08	0.06	0.0046719179099670	0.0046719183644265
(BA3)	Portugues	0.03	0.06	0.0017557114998180	0.0017557117287998
(BA4)	Arab	0.16	0.06	0.0091050234409350	0.0091050233957575
(BA5)	Jepang	0.10	0.06	0.0059350466835180	0.0059350464966563
(BA6)	Mandarin	0.24	0.06	0.0139243497482860	0.0139243453374883
(BA7)	Korea	0.05	0.06	0.0031286763319750	0.0031286762371776
CR =				0.0169332746	0.016933263

Perhitungan bobot kriteria yang dilakukan oleh aplikasi sebagaimana tersaji dalam tabel 3 dan 4 mampu memberikan kesesuaian hingga 7 angka di belakang koma jika dibandingkan dengan hasil perhitungan yang dilakukan secara manual.

Hasil perkalian antara bobot alternatif dengan bobot kriteria yang dihitung menggunakan persamaan (6) akan menghasilkan urutan alternatif yang telah disesuaikan dengan banyaknya kuota lowongan kerja yang diminta (M_{Hasil}) seperti tersaji pada Gambar 7 yang menunjukkan hasil pencarian untuk 10 orang lulusan.

No	SCORE URUTAN	NAMA	TEMPAT	TGL LHR	PRODI	TELP	EMAIL	FB
1	0.704625	indra permata daru	slaman	1987-06-22	TI	85725959542	indrarockster@yahoo.com	link
2	0.624236	Rahmad Qadri	Oransbani	1991-04-04	SI	81225104764	amaqodri@gmail.com	link
3	0.621564	CINTHIA AGATA YUSTARINA	LAHAT	1992-02-05	SI	85725555462	thyathyot@gmail.com	link
4	0.620951	Kharina Capriota	Surabaya	1992-01-19	TI	81343528404	nina.capriota@gmail.com	link
5	0.611833	Suprpto	Jambi	1990-02-03	TK	85325390094	kikisuprpto@yahoo.co.id	link
6	0.609834	Yudha Anggara	slaman	1991-01-26	TI	85702466222	yudhaanggara123@gmail.com	link
7	0.59935	FITHRIAH RINI	LANGGE	1984-01-10	SI	85299007332	fithriahriini@gmail.com	link
8	0.596151	Oky Mahendra	Brebes Jawa Tengah	1990-10-24	SI	85642698170	okymahendra@gmail.com	link
9	0.594949	Abdul hayyi numan	lingkok godak	1993-01-28	SI	87738351186	a.hayyi.numan@gmail.com	link
10	0.591125	Gusti Dani Yuzmiri	Katapang	1992-08-26	TI	85245507143	gustidaniyuzmiri268@gmail.com	link

Hak Cipta © Agung Budi Prasetyo
Peneliti SIMIK AKAKOM Yogyakarta
email: agung_bp@akakom.ac.id

Gambar 7. Hasil Pencarian Lulusan

Hasil urutan alternatif inilah yang kemudian menjadi hasil dari aplikasi mesin pencari dalam menemukan lulusan yang memiliki kompetensi sesuai dengan kriteria yang ditetapkan pengguna lulusan.

4. KESIMPULAN

Dari hasil penelitian ini dapat disimpulkan beberapa hal sebagai berikut :

- AHP dapat digunakan sebagai metode pengurutan kriteria calon tenaga kerja dan dapat diterapkan dalam mesin pencari data lulusan.
- Berdasarkan data lulusan yang digunakan, aplikasi mampu melakukan pengurutan data lulusan (alternatif) hingga 198 lulusan berdasarkan 14 kriteria dan 57 sub kriteria yang ditentukan oleh pengguna lulusan.
- Banyaknya lulusan yang ditemukan oleh aplikasi dapat disesuaikan berdasarkan kebutuhan (kuota) yang dikehendaki pengguna lulusan.

DAFTAR PUSTAKA

- Syafiq, A. 2008. "Link And Match, Solusi Atau Ilusi?" URL : <http://materi.uniku.ac.id/tracer%20studi%20diksi/link%20and%20match.html>, diakses tanggal 22 Januari 2013.
- Kusumadewi, S., et al. (2006). *A. Fuzzy Multi -Attribute Decision Making*. Yogyakarta : Graha Ilmu.
- Saaty T.L. 1987. "Uncertainty and rank order in the analytic hierarchy process". *European Journal of Operational Research* 32: 27-37.
- Sinaga J. 2009. "Penerapan Analytical Hierarchy Process (AHP) dalam Pemilihan Perusahaan Badan Usaha Milik Negara (BUMN) sebagai Tempat Kerja Mahasiswa Universitas Sumatera Utara (USU). Medan" : Departemen Matematika Fakultas Matematika Dan Ilmu Pengetahuan Alam Universitas Sumatera Utara.
- Indriyati. 2012. "Aplikasi AHP untuk Penilaian Kinerja Dosen. *Jurnal Masyarakat Informatika*". Volume 3, Nomor 5: 34.
- Alonso, J.A., Lamata, MT. 2006. "Consistency in The Analytic Hierarchy Process: A New Approach". *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol 14, No. 4, 445-459. World Scientific Publishing Company.