

## USULAN METODE INTELLIGENT AHP DAN PENERAPANNYA PADA KLASIFIKASI IRIS FLOWER

### *PROPOSED METHODE INTELLIGENT AHP AND ITS IMPLEMENTATION IN CLASSIFICATION OF IRIS FLOWER*

Rachmat Fahmi Aziz<sup>1</sup>, Mahmud Imrona, DRS., M.T.<sup>2</sup>, Dawam Dwi Jatmiko Suwawi, S.T., M.T.<sup>3</sup>

<sup>1</sup>. [rachmat.fahmiaziz@gmail.com](mailto:rachmat.fahmiaziz@gmail.com) <sup>2</sup>. [mahmudimrona@telkomuniversity.ac.id](mailto:mahmudimrona@telkomuniversity.ac.id) <sup>3</sup>. [panggilsayadawam@gmail.com](mailto:panggilsayadawam@gmail.com)  
School of Computing, Telkom Engineering School, Telkom University, Bandung, Indonesia

AHP merupakan sebuah metode pengambilan keputusan menggunakan banyak kriteria yang ditemukan oleh Thomas L. Saaty pada tahun 1980. Teknik metode AHP adalah menguraikan masalah menjadi beberapa kriteria dan alternatif, metode ini telah banyak digunakan dalam penyelesaian masalah pengambilan keputusan selama 27 tahun terakhir dan dilakukan dalam berbagai bidang. Namun dalam memberikan bobot kriteria AHP masih dilakukan secara manual. Pemberian bobot / *weight* secara manual ini masih menjadi masalah utama dalam metode AHP, selain tidak efektif dalam waktu, pembobotan juga dilakukan secara subjektif sehingga *user* harus menerka-nerka sampai didapat pembobotan yang baik dengan nilai  $CR \leq 10\%$ . Apabila penghitungan dan pemberian bobot setiap kriteria dilakukan secara manual, maka akan membutuhkan waktu yang sangat lama, sehingga dibutuhkan solusi bernama *Intelligent AHP* yang bisa melakukan penghitungan dan pemberian bobot setiap kriteria secara *intelligent*. Studi kasus yang digunakan adalah klasifikasi *iris flower* karena mempunyai 4 kriteria yang tepat untuk digunakan dalam klasifikasi menggunakan AHP. Pada penelitian ini didapatkan sebuah algoritma untuk membangkitkan matriks perbandingan berpasangan kriteria, melakukan normalisasi dan menyimpan setiap vektor konsisten yang menghasilkan  $CR \leq 9\%$  secara *intelligent*. Kinerja metode *Intelligent AHP* dalam mengklasifikasikan *iris flower* diukur dengan menghitung ERROR yang dihasilkan. Hasil ERROR terkecil didapat sebesar 5.556% sehingga pengklasifikasian *iris flower* dapat dilakukan.

**Kata kunci :** *AHP, Intelligent AHP, iris flower*

#### 1. Pendahuluan

AHP merupakan sebuah metode pengambilan keputusan menggunakan banyak kriteria (*multi-criteria decision making*) yang ditemukan dan dikembangkan oleh Thomas L. Saaty (1980). Teknik dari metode AHP adalah dengan menguraikan masalah menjadi beberapa kriteria dan alternatif [1]. AHP telah banyak digunakan dalam penyelesaian masalah pengambilan keputusan selama 27 tahun terakhir, penerapannya sudah dilakukan dalam berbagai bidang kehidupan, diantaranya di bidang logistic, bisnis, manufaktur, pemerintahan bahkan militer [2].

Cara kerja AHP adalah dengan membandingkan kriteria satu sama lain dengan menggunakan matriks perbandingan berpasangan, dan kemudian dilakukan pembobotan terhadap tiap-tiap perbandingan kriteria, nilai yang dimasukkan ke dalam perbandingan adalah bobot yang didapat dari penilaian terhadap kriteria dibandingkan dengan kriteria yang lainnya, ada 17 kemungkinan untuk mengisi bobot tersebut, yaitu 1,2,3,...,9 dan sebaliknya yaitu  $\frac{1}{2}, \frac{1}{3}, \dots, \frac{1}{9}$ . Setelah dibandingkan, matriks perbandingan berpasangan tersebut kemudian dinormalisasi dan didapatkan nilai vektor *eigen* dan nilai CR, dari sini sudah didapat urutan *ranking* dari setiap kriteria [1]. Namun tidak semua kombinasi layak digunakan, karena hasil CR yang digunakan harus kurang dari 10% [3].

Selanjutnya, mengolah data alternatif. Pengerjaannya sama dengan mengolah kriteria sebelumnya, hanya saja di sini setiap alternatif dibandingkan satu sama lain berdasarkan kriteria. Jika ada  $n$  kriteria, maka entry yang harus diisi ada  $n/2(n-1)$ . Karena ada total 17 kemungkinan, maka total untuk sebuah matriks perbandingan berpasangan akan ada sebanyak  $17^{(n/2(n-1))}$  [4]. Jika pemberian bobot dilakukan secara manual, maka akan membutuhkan waktu yang sangat lama dan tidak efektif. Sehingga pada tugas akhir ini diusulkan sebuah metode untuk melakukan pembobotan kriteria dan menentukan dan menyimpan kumpulan vektor konsisten yang menghasilkan nilai  $CR \leq 9\%$  secara otomatis.

Metode usulan ini selanjutnya disebut sebagai metode *Intelligent AHP*. Pada penelitian kali ini metode *Intelligent AHP* akan diterapkan untuk mengklasifikasikan *iris flower* karena ketepatannya dengan metode AHP yang memiliki banyak kriteria [5] [6], *dataset iris flower* memiliki 4 kriteria. Pengklasifikasian dilakukan dengan cara mencari nilai *Weight* yang terbaik untuk mendapatkan hasil ERROR yang terbaik.

## 2. AHP

*Analytic Hierarchy Process* (AHP) adalah sebuah metode pengambilan keputusan menggunakan banyak kriteria (*multi-criteria decision making*) yang ditemukan dan dikembangkan oleh Thomas L. Saaty pada tahun 1980. Pada metode AHP, masalah diuraikan menjadi hirarki dari kriteria dan alternatif [1].

Penerapan AHP sudah dilakukan dalam berbagai bidang kehidupan, diantaranya di bidang logistik, bisnis, manufaktur, pemerintahan bahkan militer [2].

## 3. Pemberian Bobot AHP

Pemberian bobot untuk setiap kriteria dilakukan setelah matriks perbandingan berpasangan dibentuk, matriks perbandingan berpasangan ini akan membentuk sebuah matriks  $n \times n$ , sebagai contoh untuk permasalahan pemilihan pesawat di atas, maka akan menghasilkan matriks perbandingan berpasangan dengan ukuran  $6 \times 6$ . Bobot yang dimasukkan adalah penilaian seberapa penting satu kriteria dengan kriteria lainnya, bobot kriteria yang diisikan dapat dilihat pada **Error! Reference source not found.** sebagai berikut [1] [4]:

Tabel 1 Bobot kriteria dan penjelasannya [1]

| <i>Intensisty of importance</i> | <i>Definition</i>   | <i>Explanation</i>  |
|---------------------------------|---|---|
| 1                               | <i>Equal Importance</i>   | <i>Two activities contribute equally to the objective</i>   |
| 2                               | <i>Weak or slight</i>   |   |
| 3                               | <i>Moderate importance</i>  | <i>Experience and judgement slightly favour one activity over another</i>   |
| 4                               | <i>Moderate plus</i>  |   |
| 5                               | <i>Strong importance</i>  | <i>Experience and judgement strongly favour one activity over another</i>   |
| 6                               | <i>Strong plus</i>  |   |
| 7                               | <i>Very strong or demonstrated importance</i>   | <i>An activity is favoured very strongly over another, its dominance demonstrated in practice</i>   |
| 8                               | <i>Very, very strong</i>  |   |
| 9                               | <i>Extreme importance</i>   | <i>The evidence favouring one activity over anoother is of the highest possible order of affirmation</i>  |
| <i>Intensisty of importance</i> | <i>Definition</i>   | <i>Explanation</i>  |
| <i>Reciprocals of above</i>     | <i>If activity I has one of the above non zero numbers assigned to it when compared with activity j, then j has reciprocal value when compared with i</i> | <i>A reasonable assumption</i>  |
| 1.1-1.9                         | <i>If the activities are very close</i>   | <i>Maybe difficult to assign the best value but when compared with other constraunting activities the size of the small numbers would not be too noticeable, yet they can still indicate the relative importance of the activities.</i> |

Untuk pembobotan dilakukan dengan membandingkan tiap-tiap kriteria, pemberian bobot ini dapat berupa penilaian subjektif ataupun penilaian secara *expert*, namun tetap harus ada aturan yaitu nilai CR harus lebih kecil atau sama dengan 10%, apabila  $CR > 10\%$ , maka pembobotan untuk matriks perbandingan berpasangan tidak dapat diterima karena bersifat tidak konsisten [4].

Tahapan selanjutnya adalah membuat matriks perbandingan untuk setiap alternatif berdasarkan tiap kriteria. kemudian setiap alternatif dibandingkan satu sama lain dalam bentuk matriks perbandingan berpasangan

berdasarkan setiap kriteria, misalnya ada 7 kriteria, maka matriks perbandingan berpasangan untuk membandingkan alternatifnya ada 7 buah.

Misal ada n alternatif, maka untuk alternatif ke-i nilainya [4]:

$$w_1u_{1j} + w_2u_{2j} + \dots + w_nu_{nj} \tag{1}$$

Dimana :

- $w_j$  = bobot untuk kriteria ke-j
- $u_{ji}$  = nilai utilitas dari kriteria ke-j untuk alternatif ke-i
- $n$  = jumlah kriteria

Tahap AHP :

- Menentukan nilai  $W_j$
- Membuat matriks perbandingan berpasangan
- Normalisasi matriks berpasangan
- Menentukan nilai bobot (vektor eigen), untuk nilai eigen maksimum menjadi eigen maksimum ( $\lambda_{maks}$ ).
- Menghitung konsistensi rasio

$$CI = \frac{CI}{RI}$$


---


$$RI = \frac{\lambda_{maks} - n}{n - 1} \tag{2}$$

RI diperoleh dari tabel

Entry matriks perbandingan berpasangan adalah sebanyak 17 buah, yaitu angka-angka yang sudah dijelaskan di atas, namun untuk beberapa kasus hal ini bisa digantikan dengan langsung membandingkan nilai aslinya, seperti nilai ujian, harga, dll.

#### 4. Random Inconsistency pada AHP

RI merupakan *Random Inconsistency Index* yang diperoleh dari Tabel II-3, Thomas L. Saaty dan William T. Tran telah mengemukakan bahwa nilai RI akan bernilai 0 saat perbandingannya 1 dan 2, dan akan naik dan berbentuk *asymptot* menuju angka 15, dengan batas 1,59. Sehingga untuk perbandingan 16 ke atas, nilai RI akan tetap di angka 1,6 [4]. Detail nilainya dapat dilihat pada **Error! Reference source not found.** di bawah.

Tabel 2 Nilai RI [4]

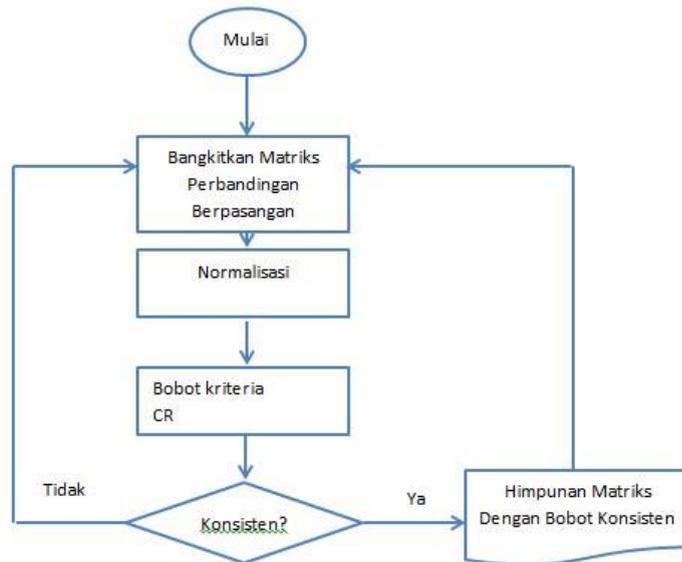
|       |   |   |      |      |      |      |      |     |      |      |      |      |      |      |      |
|-------|---|---|------|------|------|------|------|-----|------|------|------|------|------|------|------|
| Order | 1 | 2 | 3    | 4    | 5    | 6    | 7    | 8   | 9    | 10   | 11   | 12   | 13   | 14   | 15   |
| R.I.  | 0 | 0 | 0.52 | 0.89 | 1.11 | 1.25 | 1.35 | 1.4 | 1.45 | 1.49 | 1.52 | 1.54 | 1.56 | 1.58 | 1.59 |

#### 5. Perancangan sistem

Berbeda dengan AHP yang biasa dilakukan untuk mencari *ranking* terbaik untuk memilih alternatif yang terbaik, *intelligent AHP* kali ini digunakan untuk melakukan klasifikasi, dengan asumsi bahwa nilai  $A_i$  yang didapat dari nilai asli (nilai asli nanti akan disebut  $U_i$ ) dikalikan dengan nilai vektor konsisten, akan mendapatkan nilai  $A_i$  dengan *range* yang unik untuk menentukan *range* kelas tiap-tiap spesies *iris flower*, maka tidak dilakukan perbandingan berpasangan untuk nilai alternatif.

$$A_i = U_{i1}W_{i1} + U_{i2}W_{i2} + U_{i3}W_{i3} + U_{i4}W_{i4} \tag{3}$$

Selanjutnya didapat nilai  $A_i$  yang akan dijadikan rentang nilai untuk menentukan spesies *iris flower*. Di dalam tugas akhir ini, dibangun dua buah program. Program yang pertama dapat melakukan klasifikasi *iris flower* dengan menggunakan masukan berupa *dataset* yang telah ditentukan sebelumnya. Sistem dalam tugas akhir ini menangani pengklasifikasian menggunakan *Intelligent AHP*. Sedangkan program kedua akan digunakan dalam pengaplikasian dalam pengklasifikasian *iris flower* langsung satu-satu.



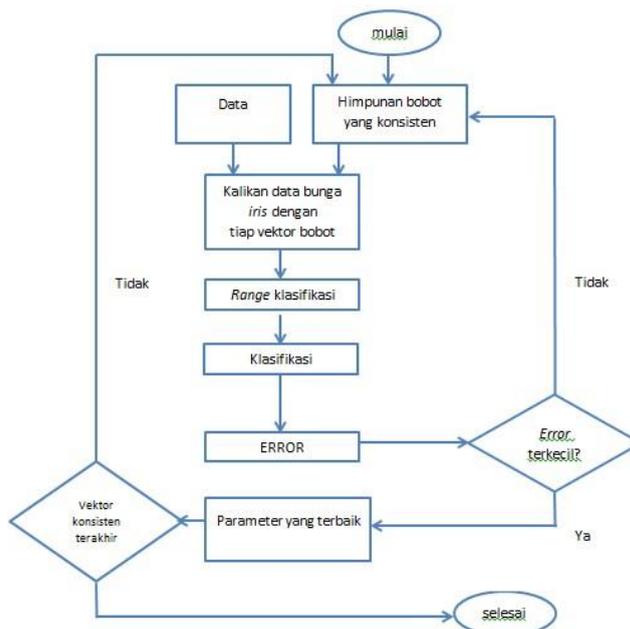
Gambar 1 Detail membangkitkan matriks perbandingan berpasangan

Pertama, matriks perbandingan berpasangan dari kriteria dibangkitkan secara otomatis, hasil dari proses ini adalah sebuah matriks 4 x 4 yang terdiri dari matriks perbandingan tiap-tiap kriteria, yaitu *sepal length*, *sepal width*, *petal length* dan *petal width*.

Setelah itu matriks tersebut dinormalisasi, normalisasi dari matriks tersebut akan menghasilkan vektor eigen dan nilai CR. Apabila nilai CR dari matriks tersebut adalah kurang dari 9%, maka vektor eigen tersebut akan disimpan ke dalam file .csv untuk digunakan ditahap selanjutnya.

Jika CR yang didapat ternyata lebih dari 9%, maka matriks tidak konsisten sehingga proses kembali ke tahap awal yaitu membangkitkan matriks perbandingan berpasangan dengan kombinasi yang baru. Begitu seterusnya sampai kombinasi matriks sudah lengkap.

**Error! Reference source not found.** berikut merupakan *flowchart* dari sistem yang akan dibangun untuk menentukan parameter terbaik. Parameter terbaik yang dihasilkan akan digunakan dalam aplikasi *software* klasifikasi *iris flower*.



$$Error = \frac{\sum_{i=1}^n |w_i - w_i^*|}{\sum_{i=1}^n w_i} \times 100\% \quad (4)$$

Gambar 2 Detail intelligent AHP

Parameter yang dihasilkan adalah berupa weight setiap kriteria, yaitu *sepal length*, *sepal width*, *petal length* dan *petal width*.

Poin terpenting dalam metode *Intelligent* AHP yang membedakannya dengan metode AHP biasa adalah proses membangkitkan matriks perbandingan berpasangan yang dibangkitkan secara otomatis dan *intelligent*. **Error! Reference source not found.** berikut menampilkan *pseudocode* untuk membangkitkan matriks perbandingan dan mengisi bobot kriteria matriks perbandingan berpasangan secara otomatis.

Poin terpenting dalam metode *Intelligent* AHP yang membedakannya dengan metode AHP biasa adalah proses membangkitkan matriks perbandingan berpasangan yang dibangkitkan secara otomatis dan *intelligent*.

## 6. Spesifikasi Sistem

Pada penelitian kali ini akan dibangun 2 buah program, berikut spesifikasi untuk kedua program tersebut;

- a. Untuk program pertama:
  - Sistem bisa menampilkan performansi dari metode yang digunakan
  - Sistem bisa menentukan vektor konsisten yang menghasilkan ERROR terkecil
  - Sistem bisa melihat hasil klasifikasi untuk vektor konsisten tertentu.
- b. Untuk program kedua:
  - Sistem bisa melakukan klasifikasi secara langsung dengan inputan parameter *sepal length*, *sepal width*, *petal length* dan *petal width*.
  - Sistem bisa menampilkan nilai  $W_i$  yang sedang digunakan, dan juga menampilkan *range* nilai  $A_i$  setiap spesies sesuai *data testing* yang dipilih.

## 7. Pengumpulan Data

Data diperoleh dari <http://archive.ics.uci.edu/ml/datasets/Iris> [7] diakses pada bulan Februari 2015. Data yang diambil berupa data dengan total 150 data *iris flower* dengan rincian 50 spesies *iris setosa*, 50 *iris versicolor* dan 50 *iris virginica*, detail dataset *iris flower* dapat dilihat pada lampiran 1. Dataset ini nantinya akan diolah menjadi data training dan beberapa data testing untuk pengujian. Data pengujian terdiri dari *data training* dan *data testing*, dimana *data training* digunakan untuk menentukan nilai *range*  $A_i$  untuk setiap jenis spesies, sedangkan *data testing* digunakan untuk melakukan klasifikasi berdasarkan *range* nilai  $A_i$  yang didapat dari *data training*, seteah itu dihitung besarnya *error* (ERROR) yang terjadi.

### a. Data training

Untuk melakukan pengujian terhadap sistem yang dibangun digunakan *dataset* berisi 90 data *iris flower* yang terdiri dari 30 *iris setosa*, 30 *iris versicolor* dan 30 *iris virginica*, data yang diambil adalah masing-masing 30 data pertama pada *dataset iris flower*. *Data training* yang lengkap dapat di lihat di lampiran 2.

### b. Data testing

Tabel 3 Data testing yang digunakan

| <i>Data testing</i> | Total | <i>Setosa</i> | <i>Versicolor</i> | <i>Virginica</i> |
|---------------------|-------|---------------|-------------------|------------------|
| 1                   | 45    | 15            | 15                | 15               |
| 2                   | 60    | 20            | 20                | 20               |
| 3                   | 75    | 25            | 25                | 25               |
| 4                   | 90    | 30            | 30                | 30               |
| 5                   | 120   | 40            | 40                | 40               |
| <i>Data testing</i> | Total | <i>Setosa</i> | <i>Versicolor</i> | <i>Virginica</i> |
| 6                   | 135   | 45            | 45                | 45               |
| 7                   | 150   | 50            | 50                | 50               |
| 8                   | 50    | 50            | 0                 | 0                |
| 9                   | 50    | 0             | 50                | 0                |
| 10                  | 50    | 0             | 0                 | 50               |

Dari tabel di atas dapat dilihat bahwa *dataset* yang digunakan memiliki kombinasi jumlah total dan jumlah spesies tertentu yang cukup bervariasi.

## 8. Skenario Pengujian

Skenario pengujian untuk menentukan hasil dengan ERROR terkecil sudah dijelaskan pada gambar deskripsi sistem, dimana pengecekan ERROR dilakukan dengan pengujian *data testing* untuk menentukan *range*  $A_i$  tiap spesies, setelah itu langsung dilakukan pengujian untuk *data testing* yang digunakan untuk menentukan nilai ERROR. Pengujian ini berlangsung sebanyak vektor konsisten yang didapatkan. Vektor konsisten didapat dengan menentukan batas CR tertentu untuk membatasi banyaknya vektor konsisten yang dihasilkan.

### a. Pengujian *Data training*

Pengujian terhadap *data training* dilakukan dengan melakukan penghitungan  $A_i$  yang didapat dari perkalian vektor yang konsisten dengan nilai utilitas setiap kriteria dari *data training*.

$$A_i = U_{i1}W_{i1} + U_{i2}W_{i2} + U_{i3}W_{i3} + U_{i4}W_{i4} \quad (1)$$

Penjelasan :

$A_i$  : nilai hasil kali *utilitas data* alternatif dengan *weight* kriteria

$U_i$  : nilai utilitas (*sepal length, sepal width, petal length, petal width*) yang didapat langsung dari data

$W_i$  : nilai *weight* kriteria yang didapat dari matriks perbandingan berpasangan untuk kriteria

Rumus ( $A_i = U_{i1}W_{i1} + U_{i2}W_{i2} + U_{i3}W_{i3} + U_{i4}W_{i4}$ ) (1) didapatkan setelah sebelumnya menggunakan cara lain yaitu sama dengan cara mencari nilai  $A_i$  pada metode AHP biasa, dimana di sini dilakukan pembangkitan matriks perbandingan berpasangan dengan matriks berupa matriks perbandingan berpasangan untuk setiap alternatif, jadi pada kasus ini dilakukan perbandingan berpasangan dengan 90 alternatif sehingga di dapatkan matriks perbandingan berpasangan dengan matriks berukuran  $90 \times 90$ . Selanjutnya didapat nilai  $W_i$  dari vektor konsisten dan nilai  $U_i$  dari matriks  $90 \times 90$  yang dibentuk dan dihitung secara otomatis, dari sini dapat dihitung nilai  $A_i$  yang akan dijadikan rentang nilai untuk menentukan spesies *iris flower*. Namun ternyata pada percobaan dengan menggunakan metode ini menghasilkan error sebesar 100% atau dengan kata lain, tidak bisa melakukan klasifikasi dengan cara ini. Hal ini terjadi karena nilai  $A_i$  yang dihasilkan dengan cara ini hasilnya berbeda apabila jumlah *data testing* dengan *data training* berbeda. Sehingga cara ini diganti dengan langsung mengalikan nilai  $W_i$  dengan nilai utilitas asli dari data yang diolah, sehingga nilai  $A_i$  yang dihasilkan lebih konsisten walaupun jumlah data yang digunakan berbeda. Selain itu, cara ini juga dilakukan dengan mempertimbangkan nilai standar deviasi dari data *iris flower* bernilai kecil, sehingga perhitungan nilai  $A_i$  bisa dilakukan dengan mengalikan langsung nilai *weight* dengan nilai utilitas asli *dataset iris flower*.

Pengujian ini dilakukan untuk menentukan *range* nilai  $A_i$  yang akan digunakan dalam melakukan klasifikasi pada *data testing*. 90 data *iris flower* yang terdiri dari 30 jenis *setosa*, 30 jenis *versicolor* dan 30 jenis *virginica* akan ditentukan nilai  $A_i$ -nya, dari nilai  $A_i$  ini akan didapat *range* tertentu yang akan membedakan setiap spesies. *Range* inilah yang nanti akan digunakan untuk mengklasifikasikan spesies *iris flower* pada *data testing*.

### b. Pengujian *Data testing*

Pengujian terhadap *data testing* hampir sama dengan pengujian *data training*, yaitu dengan mencari nilai dari  $A_i$  setiap spesies, dari sini ditetapkan dengan *range*  $A_i$  yang telah didapat dari *data training* sehingga dapat dilakukan klasifikasi, disini dihitung jumlah persentase ERROR dari hasil klasifikasi. Vektor konsisten yang menghasilkan nilai ERROR terkecil akan digunakan untuk aplikasi yang kedua.

Pada pengujian *data testing* dilakukan sebanyak 10 kali, dengan *dataset testing* yang berbeda-beda. Setiap masukan *dataset testing* ( $U_i$ ) akan dilakukan operasi dengan nilai  $W_i$  yang didapat dari *intelligent AHP*, sehingga didapat nilai  $A_i$  untuk *data testing* dan dilakukan klasifikasi untuk *data testing*, disini dilihat ERROR (persentase jumlah *error*) dan waktu yang diperlukan untuk melakukan operasi.

## 9. Analisis Hasil Pengujian

**a. Kumpulan Vektor Konsisten**

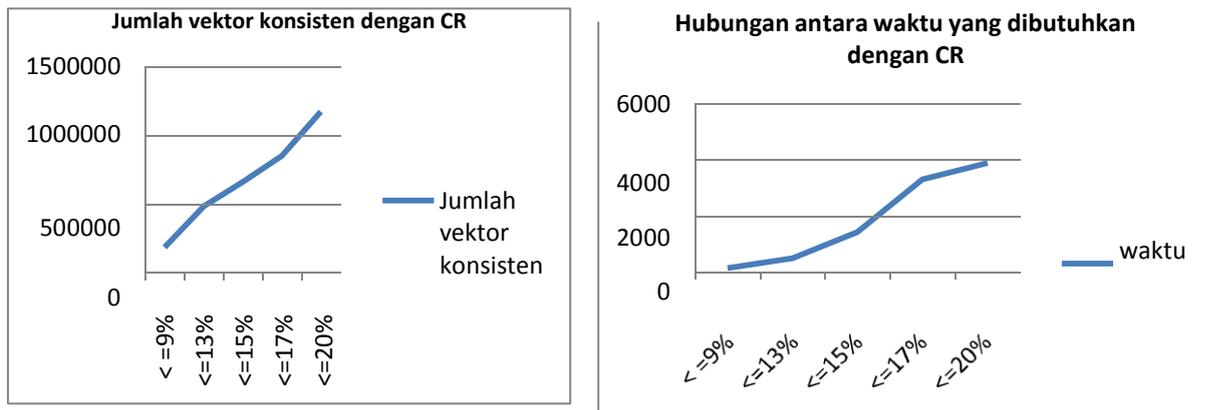
Kumpulan vektor konsisten didapat dari hasil perhitungan  $W_i$  yang dihasilkan dari setiap matriks perbandingan berpasangan yang menghasilkan nilai  $CR \leq 9\%$ . Dari hasil pengujian, didapatkan kumpulan vektor konsisten yang menghasilkan  $CR \leq 9\%$  sebanyak 362.823 buah vektor. Untuk menyelesaikan tahap ini dengan  $CR \leq 9\%$  sistem membutuhkan waktu selama 23 menit 55 detik. Sedangkan untuk  $CR \leq 15\%$ , total waktu yang diperlukan adalah 45 menit 19 detik dan menghasilkan 838.185 buah vektor konsisten, detail hasil analisis kumpulan vektor konsisten dapat dilihat pada Tabel berikut ini.

Tabel 4 Analisis hasil pengujian vektor konsisten dan waktu yang dibutuhkan

| CR          | Jumlah vektor konsisten | Waktu yang dibutuhkan |
|-------------|-------------------------|-----------------------|
| $\leq 9\%$  | 362.823                 | 23 menit 55 detik     |
| $\leq 13\%$ | 661.888                 | 29 menit 46 detik     |
| $\leq 15\%$ | 838.185                 | 45 menit 19 detik     |
| $\leq 17\%$ | 1.030.833               | 76 menit 42 detik     |
| $\leq 20\%$ | 1.351.780               | 86 menit 24 detik     |

Dari tabel di atas, dapat dilihat bahwa kumpulan vektor konsisten yang dihasilkan semakin besar apabila CR yang digunakan semakin besar, hal ini terjadi karena semakin besar CR, maka jumlah kombinasi matriks perbandingan berpasangan yang disimpan akan semakin banyak, dan ini juga berpengaruh terhadap waktu yang diperlukan dalam menyimpan vektor konsisten dimana waktu yang dibutuhkan semakin lama. Untuk lebih jelasnya hubungan antara CR dengan jumlah vektor konsisten yang dihasilkan dan waktu yang dibutuhkan untuk prosesnya, dapat dilihat pada Gambar dan

Gambar berikut ini.



Gambar 3 Hubungan antara CR dengan jumlah vektor konsisten yang dihasilkan

Gambar 4 Hubungan antara CR dengan waktu yang diperlukan untuk proses pembangkitan matriks perbandingan berpasangan

**b. Analisis Range Nilai  $A_i$  Spesies Data training**

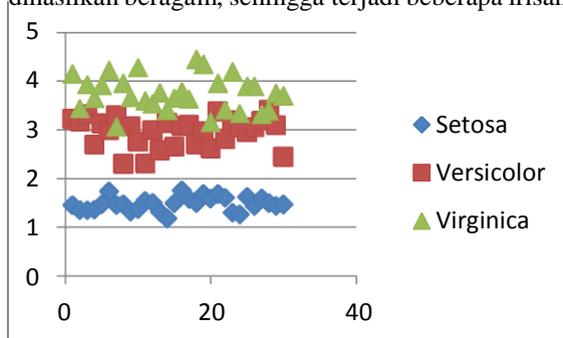
Range nilai  $A_i$  yang didapat dihasilkan dengan cara mengalikan nilai  $W_i$  dengan  $U_i$  data training seperti yang sudah dijelaskan pada scenario di atas. Hasil untuk penentuan nilai range  $A_i$  pada data training dapat dilihat pada Tabel di bawah ini:

Tabel 5 Range nilai  $A_i$  untuk data training

| Vektor konsisten ke- | Range Setosa |          | Range Versicolor |          | Range Virginica |          |
|----------------------|--------------|----------|------------------|----------|-----------------|----------|
|                      | Min          | Max      | Min              | Max      | Min             | Max      |
| 2                    | 1.239466     | 1.842152 | 2.345411         | 3.465214 | 3.099191        | 4.525243 |
| 3                    | 1.295358     | 1.899980 | 2.497767         | 3.687007 | 3.288766        | 4.828496 |
| 8                    | 1.198557     | 1.797152 | 2.245615         | 3.330214 | 2.984646        | 4.341152 |
| 2363                 | 1.712946     | 2.486837 | 2.567281         | 3.749944 | 3.289022        | 4.879026 |
| 2877                 | 2.041501     | 2.934896 | 2.711547         | 3.936442 | 3.409477        | 5.106871 |
| 7178                 | 1.931131     | 2.865253 | 2.040305         | 3.119887 | 2.543149        | 3.890861 |
| 11770                | 2.110642     | 3.140398 | 1.960637         | 3.048916 | 2.365651        | 3.711571 |
| 41196                | 2.394624     | 3.521325 | 2.169073         | 3.309314 | 2.580644        | 3.996366 |
| 61131                | 0.995555     | 1.546884 | 1.932398         | 2.918485 | 2.633371        | 3.767167 |

|       |          |          |          |          |          |          |
|-------|----------|----------|----------|----------|----------|----------|
| 79690 | 0.859989 | 1.373835 | 1.775307 | 2.722136 | 2.441350 | 3.520857 |
|-------|----------|----------|----------|----------|----------|----------|

Dari tabel di atas dapat dilihat bahwa untuk beberapa vektor konsisten terdapat *irisan* untuk *versicolor* dan *virginica*, serta ada juga yang beririsan untuk setiap spesiesnya. Hal ini terjadi karena perbedaan nilai  $W_i$  untuk setiap vektor konsisten, keberagaman nilai  $W_i$  ini menyebabkan nilai  $A_i$  yang dihasilkan beragam, sehingga terjadi beberapa irisan pada *range* nilai  $A_i$ .



|                        |                                     |
|------------------------|-------------------------------------|
| Range $A_i$ setosa     | 1.17244825487013 - 1.75266842532467 |
| Range $A_i$ versicolor | 2.29371956168831 - 3.40483969155844 |
| Range $A_i$ virginica  | 3.05379464285714 - 4.44617491883116 |

Gambar 5 Range  $A_i$  untuk data testing 4 menggunakan vektor konsisten ke-1    Tabel 6 Range  $A_i$  untuk data testing 4 menggunakan vektor konsisten ke-1

**c. Analisis Klasifikasi Data testing**

Di sini akan dibahas beberapa hasil klasifikasi untuk setiap *data testing* berdasarkan operasi dengan beberapa data vektor konsisten yang diambil secara random. Tabel di bawah ini merupakan hasil klasifikasi untuk beberapa vektor konsisten berdasarkan setiap *data testing*.

Tabel 7 Hasil klasifikasi data testing untuk beberapa vektor konsisten

| Vektor konsisten | Data testing | Total data | Hasil klasifikasi |             |           |                | Performansi |       |           |
|------------------|--------------|------------|-------------------|-------------|-----------|----------------|-------------|-------|-----------|
|                  |              |            | Setosa            | Versi-color | Virginica | Not Classified | Benar       | Salah | ERROR (%) |
| 2                | 1            | 45         | 14                | 16          | 13        | 2              | 41          | 4     | 8.89      |
|                  | 2            | 60         | 20                | 26          | 14        | 0              | 54          | 6     | 10.0      |
|                  | 3            | 75         | 25                | 18          | 21        | 1              | 70          | 5     | 6.67      |
|                  | 4            | 90         | 29                | 35          | 23        | 3              | 80          | 10    | 11.11     |
|                  | 5            | 120        | 39                | 46          | 34        | 1              | 113         | 7     | 5.83      |
|                  | 6            | 135        | 44                | 54          | 35        | 2              | 123         | 12    | 8.89      |
|                  | 7            | 150        | 49                | 58          | 40        | 3              | 137         | 13    | 8.66      |
|                  | 8            | 50         | 49                | 0           | 0         | 1              | 49          | 1     | 2.0       |
|                  | 9            | 50         | 0                 | 48          | 0         | 2              | 48          | 2     | 4.0       |
|                  | 10           | 50         | 0                 | 10          | 40        | 0              | 40          | 10    | 20.0      |
| 23187            | 1            | 45         | 34                | 0           | 10        | 1              | 23          | 22    | 48.0      |
|                  | 2            | 60         | 51                | 1           | 8         | 0              | 28          | 32    | 53.33     |
|                  | 3            | 75         | 61                | 1           | 13        | 0              | 38          | 37    | 49.33     |
|                  | 4            | 90         | 73                | 0           | 16        | 1              | 44          | 46    | 51.11     |
|                  | 5            | 120        | 98                | 1           | 21        | 0              | 61          | 59    | 49.16     |
|                  | 6            | 135        | 112               | 1           | 21        | 1              | 65          | 70    | 51.85     |
|                  | 7            | 150        | 122               | 1           | 26        | 1              | 75          | 75    | 50.0      |
|                  | 8            | 50         | 49                | 0           | 0         | 1              | 49          | 1     | 2.0       |
|                  | 9            | 50         | 48                | 1           | 1         | 0              | 1           | 49    | 98.0      |
|                  | 10           | 50         | 25                | 0           | 25        | 0              | 25          | 25    | 50.0      |
| 79690            | 1            | 45         | 15                | 18          | 12        | 0              | 42          | 3     | 6.67      |
|                  | 2            | 60         | 20                | 25          | 15        | 0              | 55          | 5     | 8.33      |
|                  | 3            | 75         | 25                | 29          | 21        | 0              | 71          | 4     | 5.33      |
|                  | 4            | 90         | 30                | 35          | 25        | 0              | 85          | 5     | 5.55      |
|                  | 5            | 120        | 40                | 44          | 36        | 0              | 116         | 4     | 3.33      |
|                  | 6            | 135        | 45                | 52          | 38        | 0              | 128         | 7     | 5.18      |
|                  | 7            | 150        | 50                | 57          | 43        | 0              | 143         | 7     | 4.67      |
|                  | 8            | 50         | 50                | 0           | 0         | 0              | 50          | 0     | 0.0       |
|                  | 9            | 50         | 0                 | 50          | 0         | 0              | 50          | 0     | 0.0       |
|                  | 10           | 50         | 0                 | 7           | 43        | 0              | 43          | 7     | 14.0      |

**d. Analisis Vektor Konsisten yang Menghasilkan ERROR Terkecil**

Seperti yang sudah dijelaskan pada skenario pengujian *data testing*, dilakukan pencarian data vektor konsisten yang menghasilkan ERROR terkecil dari setiap *data testing*, pencarian vektor konsisten

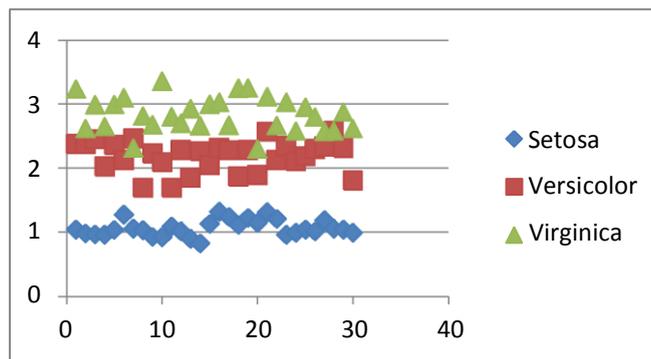
yang menghasilkan nilai ERROR terkecil ini dilakukan secara otomatis menggunakan sistem yang dibangun, hasilnya adalah Tabel berikut.

Tabel 8 Vektor konsisten yang menghasilkan ERROR terkecil untuk setiap data testing dan waktu operasi

| Data testing | Vektor konsisten | CR (%)  | ERROR(%) | Waktu yang dibutuhkan |
|--------------|------------------|---------|----------|-----------------------|
| 1            | 162899           | 7.92570 | 4.45     | 56 seconds            |
| 2            | 219787           | 3.28550 | 6.67     | 52 seconds            |
| 3            | 289939           | 6.89599 | 4.0      | 44 seconds            |
| 4            | 219787           | 3.28550 | 5.55     | 1 minute 12 seconds   |
| 5            | 219787           | 3.28550 | 3.33     | 1 minute 3 seconds    |
| 6            | 219787           | 3.28550 | 5.18     | 45 seconds            |
| 7            | 219787           | 3.28550 | 4.67     | 41 seconds            |
| 8            | 362823           | 8.37538 | 0.0      | 45 seconds            |
| 9            | 361964           | 3.88019 | 0.0      | 46 seconds            |
| 10           | 219787           | 3.28550 | 12.0     | 41 seconds            |

Dari tabel di atas dapat dilihat ada 6 buah *data testing* yang menghasilkan ERROR terkecil dengan menggunakan vektor konsisten yang sama, yaitu vektor konsisten ke-219787. Sedangkan untuk yang lain, yaitu menggunakan *data testing* 1 menghasilkan vektor konsisten ke-162899, *data testing* 3 menghasilkan vektor konsisten ke-289939, *data testing* 8 dan 9 menghasilkan vektor konsisten ke-362823 dan ke-361964. Hal ini terjadi karena untuk *data testing* 1 menggunakan *data testing* yang cukup sedikit, yaitu 45 buah, sedangkan untuk *data testing* 8 dan 9 menggunakan data dengan 1 spesies, sedangkan untuk waktu operasi berkisar antara 41 detik sampai 1 menit 12 detik.

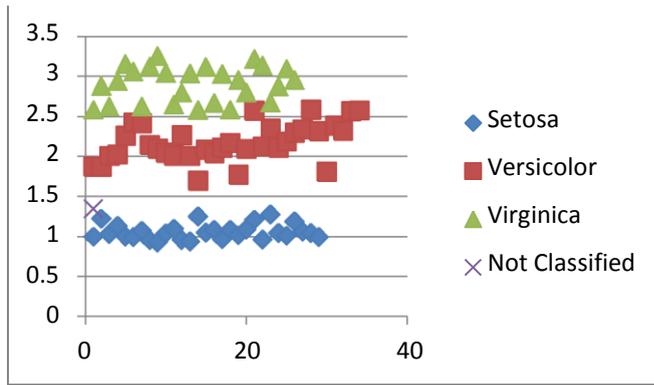
Dari sini dapat disimpulkan bahwa vektor konsisten yang terbaik adalah vektor konsisten ke-219787, dan nantinya akan digunakan sebagai patokan utama untuk nilai  $W_i$  pada aplikasi kedua. Gambar 1 di bawah merupakan gambaran/visualisasi dari *data testing* 4 yang menghasilkan ERROR terkecil, yaitu vektor konsisten ke-219787 yang menghasilkan nilai ERROR sebesar 5.556%.



Gambar 1 Klasifikasi data training menggunakan vektor konsisten ke-219787

Dari grafik di atas dapat dilihat bahwa *range setosa* tidak ada *irisan* dengan spesies lain, sehingga *range* nilai  $A_i$  untuk *setosa* sudah jelas, sedangkan untuk nilai *versicolor* dan *virginica* masih terdapat beberapa *irisan*. Untuk hasil klasifikasi pada *data testing*, dapat dilihat pada Gambar 2 dan Tabel 1 di bawah:

|                   |        |
|-------------------|--------|
| benar             | 85     |
| salah             | 5      |
| <i>Setosa</i>     | 29     |
| <i>Versicolor</i> | 34     |
| <i>Virginica</i>  | 26     |
| Not Classified    | 1      |
| ERROR             | 5.556% |



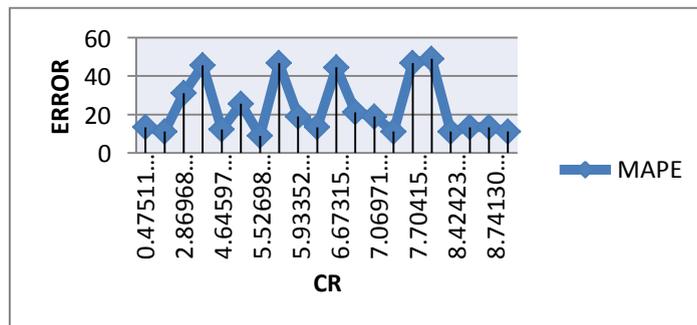
Gambar 2 Klasifikasi data testing menggunakan vektor konsisten ke-219787

Tabel 1 Klasifikasi data testing menggunakan vektor konsisten ke-219787

Hasil klasifikasi untuk *data testing* menggunakan vektor konsisten ke-219787 pada *data testing* 4 menghasilkan 1 kesalahan untuk klasifikasi *setosa*, dan 4 kesalahan untuk *virginica*.

**e. Analisis Hubungan CR dengan ERROR**

Karena 6 di antara *data testing* yang menghasilkan ERROR terkecil bernilai sama, sehingga nilai CR nya juga sama. Selanjutnya perlu diselidiki apakah CR mempengaruhi besarnya nilai ERROR, maka dilakukan analisis untuk beberapa vektor konsisten pada *data testing* 4 dengan menggunakan nilai CR yang berbeda-beda. Hubungan CR dengan ERROR dapat dilihat pada Gambar 3 berikut.



Gambar 3 Hubungan CR dengan ERROR untuk data testing 4

Grafik di atas merupakan grafik yang menggambarkan hubungan antara CR dengan ERROR. Dari grafik di atas, dapat dilihat bahwa besarnya CR ternyata tidak mempengaruhi ERROR, hal ini terjadi karena yang mempengaruhi ERROR adalah nilai vektor konsisten untuk keempat kriteria. Untuk lebih jelasnya dapat dilihat di Tabel 2 berikut:

**Tabel 2 Hubungan ERROR dengan weight kriteria vektor konsisten**

| CR (%)      | ERROR (%) | sepal L  | sepal W  | petal L  | petal W  |
|-------------|-----------|----------|----------|----------|----------|
| 8.56647516  | 5.555556  | 0.042286 | 0.155985 | 0.233653 | 0.568075 |
| 3.28550432  | 5.555556  | 0.101867 | 0.071767 | 0.082637 | 0.743729 |
| 8.52853689  | 5.555556  | 0.045163 | 0.146946 | 0.207371 | 0.60052  |
| 5.027259849 | 5.555556  | 0.046476 | 0.160703 | 0.234316 | 0.558504 |
| 8.947578488 | 5.555556  | 0.048702 | 0.129297 | 0.180481 | 0.64152  |
| 5.852404893 | 11.111111 | 0.035476 | 0.203229 | 0.401102 | 0.360193 |
| 7.081089946 | 11.111111 | 0.035452 | 0.198739 | 0.470117 | 0.295692 |
| 5.852404893 | 11.111111 | 0.035476 | 0.203229 | 0.360193 | 0.401102 |
| 8.987349056 | 11.111111 | 0.042358 | 0.350362 | 0.418996 | 0.188285 |
| 2.620744601 | 12.222222 | 0.049171 | 0.451615 | 0.425299 | 0.073915 |
| 8.149711231 | 40        | 0.045903 | 0.518265 | 0.135372 | 0.300461 |
| 8.903199775 | 51.111111 | 0.058955 | 0.579968 | 0.082314 | 0.278763 |
| 5.700970305 | 54.444444 | 0.058201 | 0.593171 | 0.053865 | 0.294763 |

|             |          |          |          |          |          |
|-------------|----------|----------|----------|----------|----------|
| 4.451225263 | 55.55556 | 0.058884 | 0.599037 | 0.055595 | 0.286484 |
| 7.893939512 | 62.22222 | 0.107592 | 0.625297 | 0.045363 | 0.221748 |

Dari tabel di atas, dapat dilihat bahwa untuk menghasilkan nilai ERROR yang kecil, prioritas pada nilai vektor konsisten berada di *Petal width*, dimana vektor konsisten dengan nilai *petal width* paling besar akan menghasilkan nilai ERROR sebesar 5.5556% yang merupakan ERROR terkecil. Berbeda dengan nilai prioritas yang tidak menempatkan *petal width* sebagai prioritas utama, akan menghasilkan ERROR yang besar. Dari sini dapat diambil kesimpulan bahwa yang mempengaruhi nilai ERROR adalah kombinasi nilai vektor konsisten yang mempunyai nilai *Petal width* paling besar.

**f. Analisis Waktu Operasi Data testing**

Di sini akan dibahas beberapa hasil klasifikasi untuk setiap *data testing* berdasarkan operasi pada program Java yang telah dibuat.

Tabel 3 Waktu operasi pada data testing untuk beberapa vektor konsisten

| Vektor konsisten | Data testing | Total data | Performansi |       |            | Waktu operasi       |
|------------------|--------------|------------|-------------|-------|------------|---------------------|
|                  |              |            | Benar       | Salah | ERROR (%)  |                     |
| 2                | 1            | 45         | 41          | 4     | 8.89       | 1 minute 8 seconds  |
|                  | 2            | 60         | 54          | 6     | 10.0       | 1 minute 6 seconds  |
|                  | 3            | 75         | 70          | 5     | 6.67       | 1 minute 24 seconds |
|                  | 4            | 90         | 80          | 10    | 11.11      | 1 minute 46 seconds |
|                  | 5            | 120        | 113         | 7     | 5.83       | 41 seconds          |
|                  | 6            | 135        | 123         | 12    | 8.89       | 42 seconds          |
|                  | 7            | 150        | 137         | 13    | 8.66       | 43 seconds          |
|                  | 8            | 50         | 49          | 1     | 2.0        | 42 seconds          |
|                  | 9            | 50         | 48          | 2     | 4.0        | 42 seconds          |
|                  | 10           | 50         | 40          | 10    | 20.0       | 49 seconds          |
| 23187            | 1            | 45         | 23          | 22    | 48.0       | 40 seconds          |
|                  | 2            | 60         | 28          | 32    | 53.33      | 48 seconds          |
|                  | 3            | 75         | 38          | 37    | 49.33      | 1 minute 1 second   |
|                  | 4            | 90         | 44          | 46    | 51.11      | 47 seconds          |
|                  | 5            | 120        | 61          | 59    | 49.16      | 40 seconds          |
|                  | 6            | 135        | 65          | 70    | 51.85      | 42 seconds          |
|                  | 7            | 150        | 75          | 75    | 50.0       | 41 seconds          |
|                  | 8            | 50         | 49          | 1     | 2.0        | 52 seconds          |
|                  | 9            | 50         | 1           | 49    | 98.0       | 41 seconds          |
|                  | 10           | 50         | 25          | 25    | 50.0       | 42 seconds          |
| 79690            | 1            | 45         | 42          | 3     | 6.67       | 41 seconds          |
|                  | 2            | 60         | 55          | 5     | 8.33       | 41 seconds          |
|                  | 3            | 75         | 71          | 4     | 5.33       | 41 seconds          |
|                  | 4            | 90         | 85          | 5     | 5.55       | 1 minute 12 seconds |
|                  | 5            | 120        | 116         | 4     | 3.33       | 40 seconds          |
|                  | Data testing | Total data | Benar       | Salah | ERROR (%)  | Waktu operasi       |
|                  | 6            | 135        | 128         | 7     | 5.18       | 1 minute 3 seconds  |
|                  | 7            | 150        | 143         | 7     | 4.67       | 41 seconds          |
|                  | 8            | 50         | 50          | 0     | 0.0        | 50 seconds          |
|                  | 9            | 50         | 50          | 0     | 0.0        | 41 seconds          |
| 10               | 50           | 43         | 7           | 14.0  | 47 seconds |                     |

Dari tabel di atas dapat dilihat bahwa untuk melakukan operasi menentukan ERROR pada vektor konsisten tertentu membutuhkan waktu berkisar antara 40 detik sampai 1 menit 46 detik.

**g. Aplikasi Kedua**

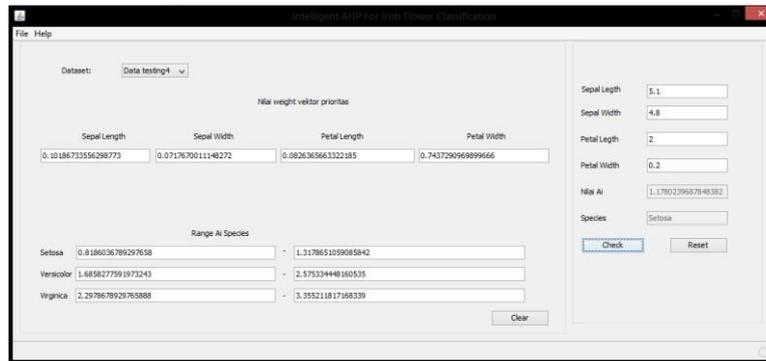
Aplikasi kedua ini adalah pengaplikasian dari hasil penelitian di program yang pertama di atas, dimana di sini dibangun sebuah aplikasi Java sederhana yang bisa mengklasifikasi spesies *iris flower* secara langsung dengan input berupa nilai *sepal length*, *sepal width*, *petal length* dan *petal width*. Pada aplikasi ini juga ditambahkan fitur untuk mengubah nilai  $W_i$  yang digunakan berdasarkan vektor konsisten terbaik yang didapat dari tiap-tiap *data testing*, Tabel 4 berikut merupakan daftar nilai  $W_i$  yang digunakan:

Tabel 4 Nilai  $W_i$  vektor konsisten terbaik

| Data testing | Vektor konsisten | ERROR(%) | $W_i$ Sepal length  | $W_i$ Sepal width    | $W_i$ Petal length  | $W_i$ Petal width   |
|--------------|------------------|----------|---------------------|----------------------|---------------------|---------------------|
| 1            | 162899           | 4.45     | 0.07948446901935274 | 0.07367051553098065  | 0.7198528215970077  | 0.12699219385265897 |
| 2            | 219787           | 6.67     | 0.10186733556298773 | 0.0717670011148272   | 0.0826365663322185  | 0.7437290969899666  |
| 3            | 289939           | 4.0      | 0.12680803486682474 | 0.040948651507777906 | 0.39843197224321875 | 0.43381134138217864 |
| 4            | 219787           | 5.55     | 0.10186733556298773 | 0.0717670011148272   | 0.0826365663322185  | 0.7437290969899666  |
| 5            | 219787           | 3.33     | 0.10186733556298773 | 0.0717670011148272   | 0.0826365663322185  | 0.7437290969899666  |

|    |        |      |                     |                     |                     |                     |
|----|--------|------|---------------------|---------------------|---------------------|---------------------|
| 6  | 219787 | 5.18 | 0.10186733556298773 | 0.0717670011148272  | 0.0826365663322185  | 0.7437290969899666  |
| 7  | 219787 | 4.67 | 0.10186733556298773 | 0.0717670011148272  | 0.0826365663322185  | 0.7437290969899666  |
| 8  | 362823 | 0.0  | 0.7342436974789915  | 0.11729691876750699 | 0.0668767507002801  | 0.08158263305322128 |
| 9  | 361964 | 0.0  | 0.37516685205784206 | 0.04974972191323693 | 0.04974972191323693 | 0.525333704115684   |
| 10 | 219787 | 12.0 | 0.10186733556298773 | 0.0717670011148272  | 0.0826365663322185  | 0.7437290969899666  |

Seperti yang sudah di jelaskan di atas, bahwa yang menentukan kecilnya nilai ERROR adalah prioritas  $W_i$  pada vektor konsisten paling besar terdapat pada  $W_i$  petal width. Dari tabel di atas terdapat 8 buah vektor konsisten yang memiliki petal width sebagai vektor prioritas terbesar. Gambar 4 berikut adalah gambar *printscreen software* klasifikasi iris flower:



Gambar 4 Software klasifikasi iris flower

## 10. Kesimpulan

1. Metode *intelligent AHP* dapat digunakan untuk melakukan klasifikasi dengan banyak kriteria.
2. Pemberian bobot/weight pada tiap kriteria pada *intelligent AHP* dapat dilakukan secara *intelligent*.
3. Metode *intelligent AHP* dapat menggantikan penilaian subjektif pada pemberian bobot/weight karena bisa menemukan kombinasi bobot yang konsisten secara *intelligent*, hal ini lebih cepat dan efisien dibandingkan dengan menggunakan penilaian subjektif dan *input* manual ke dalam *calculator AHP*.
4. Nilai CR tidak mempengaruhi hasil ERROR. Tetapi ERROR dipengaruhi oleh nilai *weight* vektor konsisten.

## 11. Saran

1. Menggunakan metode yang lain yang lebih sederhana untuk membangkitkan matriks perbandingan berpasangan agar proses learning bisa dilakukan lebih cepat dan efisien.
2. Dalam melakukan normalisasi dan menghitung vektor *eigen* dapat menggunakan metode yang lain agar didapat CR yang paling minimal dan paling konsisten.

## 12. Daftar Pustaka

- [1] Thomas L. Saaty , "How to make a decision: The Analytic Hierarchy Process," *European Journal of Operational Research* 48, pp. 9-26, 1990.
- [2] William Ho, "Integrated analytic hierarchy process - A literature review," *European Journal of Operational Research* 186, pp. 211–228, January 2008.
- [3] Thomas L. Saaty, "Decision making with the analytic hierarchy process," *Int. J. Services Sciences, Vol. 1, No. 1*, pp. 83-98, 2008.
- [4] Thomas L. Saaty and Liem T. Tran, "On the invalidity of fuzzifying numerical judgments in the Analytic Hierarchy Process," *Mathematical and Computer Modelling* 46, pp. 962–975, March 2007.
- [5] Akira Notsu, Hirokazu Kawakami, Yuki Tezuka, and Katsuhiko Honda, "Intergration of Information Based on the Similarity in AHP," *Procedia Computer Science* 22, pp. 1011 – 1020, 2013.

- [6] Seyed AmirHossein Khoshsolat, Nuraini Abdul Aziz, Napsiah Ismail, and Normariah Adam, "Repositioning of Electric Distribution Substations Using Fuzzy Decision Making," *Energy Procedia* 14, pp. 476-482, 2012.
- [7] Slavica Dožić and Milica Kalić, "An AHP approach to aircraft selection process," *Transportation Research Procedia* 3, pp. 165 – 174, July 2014.
- [8] Sir Ronald Fisher. (1993, Maret) UCI Database Repository. [Online]. <http://archive.ics.uci.edu/ml/machine-learning-databases/iris/>
- [9] Sir Ronald Fisher. (2015, Februari) Iris Flower Dataset Wikipedia.com. [Online]. [https://en.wikipedia.org/wiki/Iris\\_flower\\_data\\_set](https://en.wikipedia.org/wiki/Iris_flower_data_set)
- [10] V. P. Darji and R. V. Rao, "Intelligent Multi Criteria Decision Making Methods for Material Selection in Sugar Industry," *Procedia Materials Science* 5, pp. 2585 – 2594, 2014.
- [11] Zhang An, Tang Zhili, and Zhang Chao, "Man-machine Function Allocation Based on Uncertain Linguistic Multiple Attribute Decision Making," *Chinese Journal of Aeronautics* 24, pp. 816-822, June 2011.
- [12] Luo Xiaoli and Chen Dexian, "Research on Risk Evaluation Using RNP Technology for Operation into High Elevation Airports with Critical Terrain," *Procedia Engineering* 17, pp. 125 – 140, 2011.
- [13] Z. J. Li, X. L. Chen, and M. Ding, "Well-to-wheel Energy Consumption and Pollutant Emissions Comparison between Electric and Non-electric Vehicles: a Modeling Approach," *Procedia Environmental Sciences* 13, pp. 550-554, 2012.