

# **CLUSTER ANALYSIS UNTUK MEMPREDIKSI TALENTA PEMAIN BASKET MENGGUNAKAN JARINGAN SARAF TIRUAN *SELF ORGANIZING MAPS (SOM)***

**Gregorius Satia Budhi<sup>1</sup>, Liliana<sup>2</sup>, Steven Harryanto**

Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra

Jl. Siwalankerto 121 – 131, Surabaya 60236

E-mail: greg@petra.ac.id<sup>1</sup>, lilian@petra.ac.id<sup>2</sup>

**ABSTRAK:** Dunia bola basket telah berkembang dengan pesat seiring dengan berjalannya waktu. Hal ini ditandai dengan munculnya berbagai macam dan jenis kompetisi dan pertandingan baik dunia maupun dalam negeri. Sehingga makin banyak dilahirkannya pemain berbakat dengan berbagai karakteristik permainan yang berbeda. Tuntutan bagi seorang pelatih/pemandu bakat, untuk dapat melihat secara jeli dalam memenuhi kebutuhan tim untuk membentuk tim yang solid. Dengan dibuatnya aplikasi ini, maka akan membantu proses analisis dan pengambilan keputusan bagi pelatih maupun pemandu bakat. Aplikasi ini menggunakan algoritma *Self Organizing Maps (SOM)* untuk melakukan analisis *cluster*. Data real pemain NBA digunakan untuk keperluan proses training dan data real pemain Indonesia /pemain Universitas Kristen Petra untuk proses testing. Data pemain NBA dipersiapkan dengan melalui proses *cleaning* dan di transformasi ke bentuk yang dapat diolah oleh algoritma SOM. Kemudian data diolah menggunakan algoritma SOM untuk menghasilkan *cluster-cluster* data. Hasil *cluster-cluster* ini ditampilkan dalam bentuk yang mudah untuk dilihat dan digunakan sebagai analisis. Hasil tersebut dapat disimpan pula dalam bentuk file teks. Dengan menggunakan output dari aplikasi ini, yang berupa *cluster* pemain basket, pengambil keputusan dapat melihat statistik tiap *cluster*. Dengan menggunakan statistik tiap *cluster*, pelatih atau pemandu bakat dapat memprediksi statistik dan posisi di lapangan seorang pemain basket yang dites, yang berada pada sebuah *cluster* tertentu. Informasi ini dapat membantu pelatih atau pemandu bakat dalam pengambilan keputusan.

**Kata kunci:** *cluster analysis, Self Organizing Maps (SOM), pemain basket*

**ABSTRACT:** *Basketball World has grown rapidly as the time goes on. This is signed by many competition and game all over the world. With the result there are many basketball players with their different playing characteristics. Demand for a coach or scout to look for or search great players to make a solid team as a coach requirement. With this application, a coach or scout will be helped in analyzing in decision making. This application uses Self Organizing Maps algorithm (SOM) for Cluster Analysis. The real NBA player data is used for competitive learning or training process and real player data from Indonesian or Petra Christian University Basketball Players is used for testing process. The NBA Player data is prepared through cleaning process and then is transformed into a form that can be processed by SOM Algorithm. After that, the data is clustered with the SOM algorithm. The result of that clusters is displayed into a form that is easy to view and analyze. This result can be saved into a text file. By using the output / result of this application, that are the clusters of NBA player, the user can see the statistics of each cluster. With these cluster statistics coach or scout can predict the statistic and the position of a testing player who is in the same cluster. This information can give a support for the coach or scout to make a decision.*

**Keywords:** *cluster analysis, Self Organizing Maps (SOM), basketball player*

## **PENDAHULUAN**

Olah raga basket lahir pada tahun 1891 digagas oleh Dr. James A. Naissmith. Olah raga ini bertipe permainan. Pada awalnya ditujukan untuk menyegarkan pikiran dan perasaan orang yang terlibat dalam permainan itu. Namun hal ini berubah sejak munculnya berbagai liga bola basket di dunia, seperti liga basket di Amerika yaitu *National Basketball*

*Association (NBA)* dan juga *Indonesia Basketball League (IBL)* di Indonesia. *Basketball* berubah menjadi suatu permainan profesional yang harus ditangani secara profesional pula. Hal ini mengharuskan pengurus tim terutama pencari bakat (*scout*) dan pelatih (*coach*) untuk teliti dalam melihat bakat seorang pemain basket dan memutuskan posisinya di lapangan serta berapa lama pemain tersebut dimainkan. Salah satu cara untuk mendukung keputusan

tersebut adalah dengan melakukan analisis pada data statistik dari pemain tersebut.

Secara umum, sebuah tim yang solid adalah tim yang terdiri dari 15 pemain dengan skill diatas rata-rata. Kategori kemampuan (*skill*) dari masing-masing pemain dapat diputuskan dengan menganalisa statistik selama pemain bersangkutan bertanding. Data statistik ini sangat detail, mulai dari ukuran fisik sampai perhitungan *foul*/pelanggaran dan perolehan point. Dari analisis statistik tentang karakteristik pemain ini dapat ditentukan berbagai macam skill, yang nantinya menjadi dasar pengambilan keputusan. Keputusan tersebut dapat berupa usulan perekrutan pemain oleh pemandu bakat. Selain itu hasil analisis dapat menjadi dasar bagi pelatih guna memutuskan penempatan posisi pemain serta lama pemain bermain pada saat pertandingan.

Peneliti mencoba membuat sebuah aplikasi *cluster analysis* menggunakan metode jaringan saraf tiruan *Self Organizing Map* guna membantu analisis dan pengambilan keputusan seorang pelatih atau seorang pencari bakat, sehingga performa kerja dari sebuah tim diharapkan akan meningkat dan membaik. Sulitnya melakukan analisis manual menyebabkan terjadinya kesalahan yang dilakukan oleh seorang pelatih maupun seorang pemandu bakat dalam melakukan tugasnya. Hal ini menyebabkan kinerja dari tim menurun dan membuat mereka semakin terpuruk berada di papan bawah klasemen kompetisi.

**PENGARUH STATISTIK PADA KUALITAS PEMAIN BASKET**


Statistik seorang pemain basket dapat berbicara tentang kualitas bahkan ciri/karakter permainan dari pemain itu. Seorang pemain *guard* memiliki angka yang lebih besar pada statistik *Three Point*, *Assist*, *Stea*, dan *Turnover*. Pengaruh pada empat *field* ini terjadi karena peranan seorang *guard* pada umumnya adalah sebagai pembawa bola sekaligus pengatur serangan dan sebagai orang bertahan pertama pada saat terjadi *turnover* atau pertukaran bola. Berbeda dengan pemain *guard*, dua pemain posisi lainnya pasti sedikit bertolak belakang dengan ciri khas statistik pemain *guard*. *Forward* lebih dominan pada *Field goal*, *Three point* dan *Rebound*. Sedangkan pemain pada posisi center dominan pada *Field goal*, *Free throw*, *Rebound* dan *Block* [10].

Angka-angka dominan sesuai dengan posisi masing-masing inilah yang biasanya dijadikan tolak ukur kualitas pemain. Pengaruh besar juga dari segi tinggi dan berat badan pemain. Selain mempengaruhi karakter permainan, tinggi dan berat badan juga mempengaruhi peluang seorang pemain untuk mem-


peroleh poin-poin yang mengisi statistiknya. Seperti misalnya: untuk pemain posisi *guard*, biasanya cenderung memiliki postur tubuh yang lebih kecil daripada 2 posisi lainnya, yaitu *forward* dan *center*. Hal ini dikarenakan pemain *guard* membutuhkan kecepatan dan ketangkasan lebih untuk memecah pertahanan lawan dan menciptakan peluang bagi rekan setimnya untuk mencetak angka [10]. Contoh perbandingan statistik pemain dari 3 posisi yang berbeda (*guard*, *forward*, dan *center*) dapat dilihat pada Gambar 1.

 Name : Kobe Bryant Born : 8/23/1978 Number : 8 Height : 198 cm Weight : 99.8 kg Position : Guard	Games	66	Total Rebounds	5.9
	As Starting Line Up	66	Assists	6
	Minutes	40.7	Steals	1.3
	Field Goals %	43.3	Blocks	0.8
	3 Points %	33.9	Turnovers	4.09
	Free Throws %	81.6	Fouls	2.6
	Offensive Rebounds	1.4	Points	27.6
	Defensive Rebounds	4.5		

 Name : Chris Webber Born : 3/1/1973 Number : 4 Height : 208 cm Weight : 111.1 kg Position : Forward	Games	67	Total Rebounds	9.1
	As Starting Line Up	67	Assists	4.7
	Minutes	35.4	Steals	1.4
	Field Goals %	43.3	Blocks	0.79
	3 Points %	34.1	Turnovers	2.72
	Free Throws %	79.4	Fouls	3.2
	Offensive Rebounds	1.9	Points	19.5
	Defensive Rebounds	7.2		

 Name : Yao Ming Born : 9/12/1980 Number : 11 Height : 226 cm Weight : 134.3 kg Position : Center	Games	80	Total Rebounds	8.4
	As Starting Line Up	80	Assists	0.8
	Minutes	30.6	Steals	0.43
	Field Goals %	55.2	Blocks	2
	3 Points %	0	Turnovers	2.45
	Free Throws %	78.3	Fouls	3.7
	Offensive Rebounds	2.6	Points	18.3
	Defensive Rebounds	5.8		

**Gambar 1. Perbandingan Statistik 3 Pemain dengan Posisi Berbeda**

Adapun macam-macam statistik pemain basket dapat dilihat pada Tabel 1.

**Tabel 1. Statistik pemain basket [1, 3]**

Istilah	Deskripsi
<i>Height</i>	Tinggi badan pemain bsket (dalam centimeter)
<i>Weight</i>	Berat badan pemain basket (dalam kilogram)
<i>Games</i>	Jumlah pertandingan yang telah dialami pemain dalam 1 musim kompetisi.
<i>Games Started</i>	Jumlah pertandingan dimana pemain dipasang sebagai pemain inti pada pertandingan.
<i>Minutes</i>	Durasi waktu dimana pemain bermain dalam pertandingan.
<i>Field Goal Percentage</i>	Persentase keberhasilan usaha seorang pemain untuk mencetak poin pada pertandingan, baik itu perolehan 2 poin

Istilah	Deskripsi
	atau 3 poin.
3 Point Field Goal Percentage	Persentase keberhasilan usaha seorang pemain untuk mencetak poin melalui tembakan 3 angka pada pertandingan.
Free Throw Percentage	Persentase keberhasilan seorang pemain dalam mengeksekusi tembakan bebas atau tembakan hukuman pada pertandingan.
Offensive Rebound	Perolehan bola netral atau bola mentah hasil tembakan tim serang pada saat posisi menyerang pada pertandingan.
Defensive Rebound	Perolehan bola netral atau bola mentah hasil tembakan tim serang pada saat posisi bertahan pada pertandingan.
Total Rebound	Jumlah perolehan bola netral atau bola mentah yang dilakukan seorang pemain pada pertandingan.
Assist	Hasil <i>passing</i> atau operan bola yang membuahkan poin dalam pertandingan
Steal	Pencurian bola dari tangan tim lawan dalam pertandingan
Block	Hasil menggagalkan tembakan pemain dari tim lawan dalam pertandingan.
Turnover	Kesalahan-kesalahan yang dilakukan seorang pemain dalam pertandingan.
Foul	Jumlah pelanggaran yang dilakukan seorang pemain dalam pertandingan
Point	Jumlah angka yang dicetak oleh seorang pemain dalam pertandingan.

**KNOWLEDGE DISCOVERY DAN DATA MINING**

*Knowledge Discovery* secara sederhana dapat dikatakan sebagai proses meng-ekstrak atau menggali (*mining*) pengetahuan/informasi yang berharga (*interesting knowledge*) dari sejumlah besar data baik yang disimpan di dalam *database*, *data warehouse* maupun media penyimpanan informasi lainnya. Sementara itu *data mining* merupakan salah satu tahap yang terdapat di dalam *Knowledge Discovery* [2].

**CLUSTER ANALYSIS**

Analisis *Cluster* adalah upaya menemukan sekelompok obyek yang mewakili suatu karakter yang sama atau hampir sama (*similar*) antar satu obyek dengan obyek lainnya pada suatu kelompok dan memiliki perbedaan (*not similar*) dengan obyek-obyek pada kelompok lainnya. Tentunya persamaan dan perbedaan tersebut diperoleh berdasar informasi yang diberikan oleh obyek-obyek tersebut beserta

hubungan (*relationship*) antar obyek. Dalam berbagai kesempatan, *clustering* juga sering disebut sebagai *Unsupervised Classification* [6, 11].

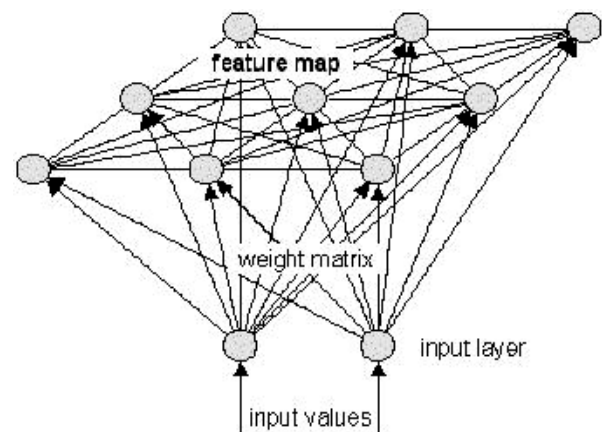
**MIN-MAX NORMALIZATION**

Metode normalisasi ini menghasilkan transformasi *linier* pada data asal. Bila *minA* dan *maxA* adalah nilai minimum dan maksimum dari sebuah atribut *A*, *Min-max Normalization* memetakan sebuah nilai *v* dari *A* menjadi *v'* dalam *range* nilai minimal dan maksimal yang baru, *new\_minA* dan *new\_maxA* [2]. Rumus *Min-max Normalization* dapat dilihat pada persamaan 1.

$$v' = \frac{v - \text{min}A}{\text{max}A - \text{min}A} * (\text{new\_max}A - \text{new\_min}A) + \text{new\_min}A \quad (1)$$

**ALGORITMA SELF ORGANIZING MAPS**

Algoritma *Self Organizing Map* (SOM) merupakan suatu metode jaringan saraf tiruan yang diperkenalkan oleh Professor Teuvo Kohonen pada tahun 1980an. SOM merupakan salah satu bentuk topologi dari *Unsupervised Artificial Neural Network* (*Unsupervised ANN*) dimana dalam proses pelatihannya tidak memerlukan pengawasan (*target output*). SOM digunakan untuk mengelompokkan (*clustering*) data berdasarkan karakteristik/fitur-fitur data [4, 5, 11]. Arsitektur dari SOM dapat dilihat pada Gambar 2.



**Gambar 2. Arsitektur SOM**

Berikut adalah algoritma SOM [7, 9, 10]:

1. Inisialisasi *neuron input* :  $x_1, x_2, \dots, x_i$ .
2. Inisialisasi *neuron output* (lapisan output) sebanyak  $j \times 1$  :  $y_{1j}, y_{2j}, \dots, y_{ij}$ .
3. Mengisi bobot antar *neuron input* dan *output*  $\mu_{ij}$  dengan bilangan *random* 0 sampai 1.

4. Mengulangi langkah 5 sampai dengan langkah 8 hingga tidak ada perubahan pada bobot *map* atau iterasi telah mencapai iterasi maksimal.
5. Pemilihan salah satu *input* dari vektor input yang ada.
6. Penghitungan jarak antar vektor input terhadap bobot ( $d_{ji}$ ) dengan masing-masing *neuron* output dengan rumus pada persamaan 2.
7. 
$$d_{ji} = \sum_{i=1}^n (\mu_{ijl} - x_i)^2 \quad (2)$$
8. Dari seluruh bobot ( $d_{ji}$ ) dicari yang paling kecil. Index dari bobot ( $d_{ji}$ ) yang paling mirip disebut *winning neuron*.
9. Untuk setiap bobot  $\mu_{ijl}$  diperbaharui bobot koneksinya dengan menggunakan rumus yang dapat dilihat pada persamaan 3.
10. 
$$\mu_{ijl}(t+1) = \mu_{ijl}(t) + \gamma(t) * h_{ijbc}(t) * (x_i(t) - \mu_{ijl}(t)) \quad (3)$$
11. Simpan bobot yang telah konvergen.

**Learning Rate ( $\gamma(t)$ )**

Digunakan untuk menunjukkan bagaimana adaptasi pembelajaran terhadap data. Fungsi skalar adaptasi bernilai  $0 \leq \gamma(t) \leq 1$ . Semakin besar nilai  $\gamma(t)$ , semakin cepat bobot koneksi beradaptasi semakin besar pengaruh vektor input terhadap perubahan bobot koneksi yang terjadi. *Learning rate* ini semakin lama akan semakin mengecil, berkurang seiring berjalannya waktu/iterasi. Semakin dekat *learning rate*  $\gamma(t)$  mendekati 0, perubahan bobot akan semakin kecil dan vektor-vektor input dapat dipetakan dengan baik [8, 9]. Fungsi  $\gamma(t)$  yang digunakan adalah penurunan konstan, dapat dilihat pada persamaan 4.

$$\gamma(t) = (1 - \frac{t}{T}) / (1 + \alpha) \quad (4)$$

**Neighborhood Function**

Fungsi  $h_{ijbc}(t)$  adalah fungsi tetangga (*Neighborhood Function*), yang bernilai  $0 \leq h_{ijbc}(t) \leq 1$ . Fungsi ini memberi pengaruh perubahan bobot secara proporsional dari *neuron best matching* ke *neuron-neuron* tetangganya. Semakin lama nilai dari fungsi ini semakin kecil, berkurang seiring berjalannya waktu/iterasi sampai mendekati 0. Semakin lama pengaruh dari perubahan bobot semakin menyempit, akhirnya hanya *neuron best matching* yang di-pengaruhi. Fungsi ini dipengaruhi oleh *Learning Rate* [8, 9]. *Neighborhood Function* dapat dilihat pada persamaan 5.

$$h_{ijbc}(t) = \exp\left(-\frac{\|r_{ij} - r_{bc}\|^2}{2\sigma^2(t)}\right) \quad (5)$$

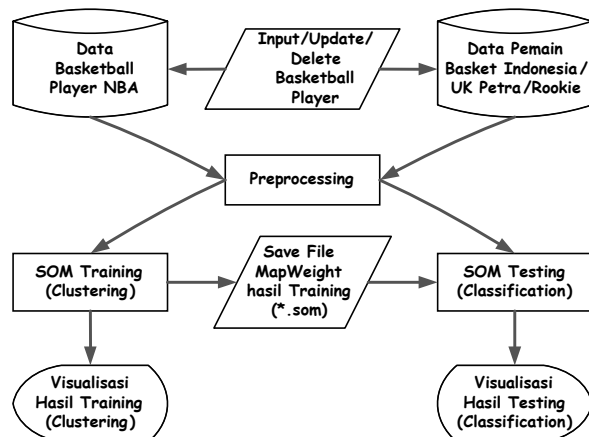
**Testing**

Berikut ini adalah langkah – langkah testing SOM untuk analisis *cluster* [5, 11]:

1. Inialisasi bobot  $\mu_{ijl}$  dengan bobot hasil *training* (*load last convergent weight*)
2. Bentuk *cluster-cluster* dari vektor data *training* (yang digunakan sebelumnya) dengan cara sama dengan proses *training*, tetapi tanpa meng-*update* bobot koneksi.
3. Definisikan spesifikasi *output* dari tiap *cluster* berdasarkan spesifikasi *output* dari tiap data *training* dalam *cluster* tersebut.
4. Masukkan vektor data baru yang akan dites dan lakukan proses *clustering* untuk semua vektor data baru.
5. *Update* spesifikasi *ouput* tiap data baru dengan spesifikasi *output* dari tiap *cluster* dimana dia berada.

**DESAIN APLIKASI**

Perencanaan aplikasi secara keseluruhan dapat dilihat di blok diagram pada Gambar 3.



**Gambar 3. Blok Diagram dari Aplikasi**

Aplikasi yang dibuat ini memiliki tiga macam proses dan empat macam *interface* yang dijelaskan berikut.

**Proses – Proses di dalam Aplikasi**

Aplikasi ini memiliki tiga proses yaitu:

- a. Preprocessing: Berfungsi untuk mentransfer data mentah (*Raw Data*) yang berasal dari dari *data training* (Pemain Basket NBA) atau *data testing* (Pemain Basket IBL/UK Petra / *Rookie*) ke dalam *array* dua dimensi. Data mentah ini berupa statistik pemain basket. Di dalam proses ini *user*, yaitu pelatih atau pemandu bakat, dapat memilih beberapa hal:

- Untuk proses *training (clustering)*, dapat dipilih musim pertandingan NBA mana yang diproses (tahunnya) dan juga jenis statistik apa saja yang digunakan untuk proses *training*. Selanjutnya semua pemain NBA pada musim yang dipilih akan menjadi vektor *input* proses *training* SOM.
- Untuk proses *testing (classification)*, dapat dipilih pemain-pemain IBL dan UK Petra mana saja yang akan diklasifikasikan. Selanjutnya pemain-pemain yang dipilih akan menjadi bagian dari vektor *input* proses *testing*.

Data yang telah ditransfer ke dalam *array* selanjutnya dinormalisasi menggunakan metode *Max-min Normalization* menjadi *range data* dari 0 sampai dengan 1 dan disimpan kembali pada *array* dua dimensi yang telah disediakan.

b. Proses *SOM Training (Clustering)*: Tahap ini berfungsi untuk mengelompokkan (*clustering*) data semua pemain NBA pada musim yang telah dipilih berdasarkan statistiknya ke dalam “map-map” dari SOM. Algoritma yang digunakan untuk tahap *training* ini dapat dilihat pada bab 2.4. Setelah *training* selesai dilakukan isi *array* bobot koneksi antara *neuron input* dan *neuron output (map)* yang telah konvergen disimpan (*save*) ke dalam file \*.som. Selain bobot, juga disimpan semua *setting* yang telah ditentukan oleh *user* untuk proses *training* ini, yaitu: tahun musim pertandingan dan macam-macam jenis statistik yang digunakan dalam *training*. Hasil *training* ini dapat pula dilihat pada *interface* “Visualisasi Hasil Training”.

c. Proses *SOM Testing (Classification)*: berfungsi untuk mengklasifikasikan *data* pemain IBL, UK Petra atau pemain baru yang dipilih *user* ke dalam map-map (*cluster-cluster*) yang ada. Secara detail langkah-langkah pada proses ini adalah sebagai berikut:

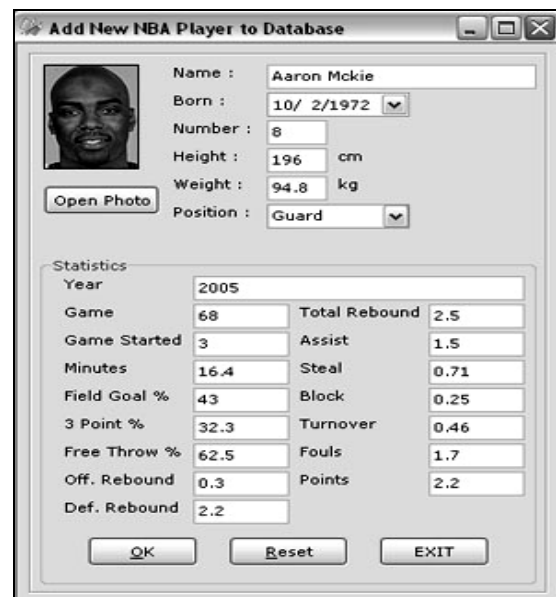
1. *User* memilih dan panggil (*load*) file \*.som yang diinginkan. Setelah file tersebut dipanggil, maka bobot yang ada di dalam *file* dimasukkan pada *array* bobot koneksi antara *neuron input* dan *neuron output (map)* SOM. Kemudian data statistik pemain NBA sesuai dengan *setting* musim dan jenis statistik yang ada dalam *file* dipersiapkan pada *preprocessing* dan dimasukkan pada vektor *input*.
2. *User* memilih pemain basket IBL, UK Petra atau pemain baru yang akan di klasifikasi (*testing*) dari data pemain IBL/UK Petra/Rookie. Setelah dipilih, maka dilakukan *preprocessing* pada data-data yang dipilih dan hasilnya dimasukkan dalam vektor *input*.

3. Sistem melakukan klasifikasi pada semua data dalam vektor *input* dengan mengukur kedekatan jarak tiap data dengan bobot koneksi tiap *neuron output (map)* SOM menggunakan persamaan 2. Dari hasil pengukuran jarak tadi semua data pada vektor input ditandai sebagai bagian dari sebuah ‘map’ SOM (*cluster*) tertentu. Selanjutnya hasil klasifikasi dapat dilihat pada *interface* “Visualisasi Hasil Testing”.

**Interface yang dipakai aplikasi**

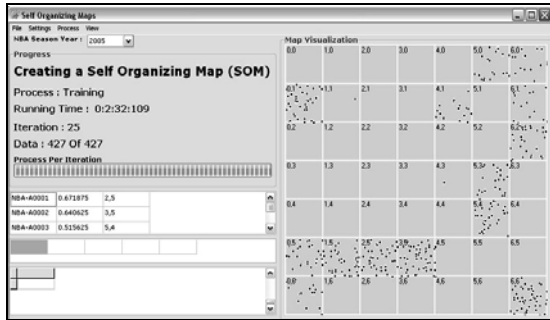
Secara umum kegunaan dari *interface* adalah untuk berkomunikasi dengan *user*. Pada aplikasi ini digunakan empat macam *interface* yaitu:

- a. *Interface Input/Update/Delete Basketball Player*, digunakan untuk memasukkan, memperbaharui atau menghapus data pemain basket pada database. Contoh *interface* dapat dilihat pada Gambar 4.



**Gambar 4. Contoh tampilan interface Input/Update/Delete Basketball Player**

- b. *Interface Setting dan Select*, sebenarnya ada beberapa macam *interface* pada kelompok ini. Secara umum fungsi dari *interface* ini digunakan untuk mempersiapkan *setting-setting* dari SOM dan memilih data-data yang akan di *training* ataupun di *testing*.
- c. *Interface Visualisasi Hasil Training*, digunakan untuk menampilkan hasil *training*. Pada *interface* ini disajikan pula perhitungan probabilitas posisi dari semua pemain yang tergabung pada map/*cluster* yang sama dan juga rata-rata statistik dari pemain-pemain tersebut. Contoh tampilan *interface* ini dapat dilihat pada Gambar 5 dan Gambar 6.



Gambar 5. Visualisasi posisi data pada tiap map setelah proses *training*.



Gambar 6. Visualisasi data member untuk sebuah map/cluster

d. *Interface* Visualisasi Hasil *Testing*, digunakan untuk menampilkan hasil *testing*. Pada *interface* ini disajikan perbandingan antara *data player testing* dan *similar NBA player* pada map/cluster yang sama. Selain itu ditampilkan pula perhitungan probabilitas posisi dari semua pemain NBA yang tergabung pada map/cluster yang sama dan juga rata-rata statistiknya. Contoh *interface* ini dapat dilihat pada Gambar 7.



Gambar 7. Visualisasi Perbandingan *Data Testing* dengan "*Similar NBA Data*"

## PENGUJIAN APLIKASI

Pengujian Aplikasi ini dilakukan pada *hardware* dan *software* dengan spesifikasi sebagai berikut:

- Processor* : Pentium IV 1,5 GHz
- Memory* : 512 MB
- HardDisk* : 20 GB
- Operating System*: MS Windows XP
- Database* : MS SQL Server 2000

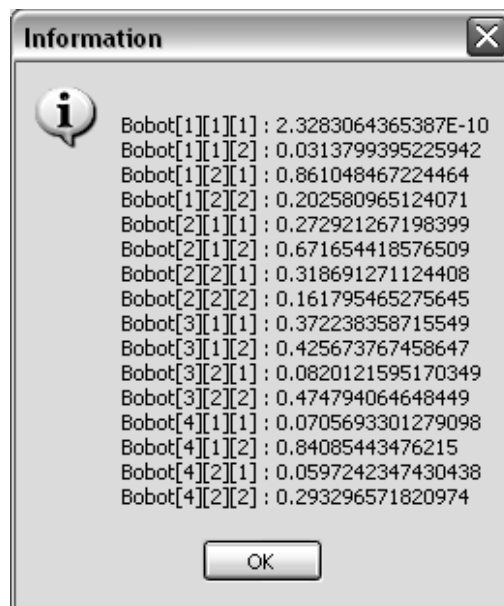
## Pengujian Kebenaran Proses SOM

Karena inti dari aplikasi ini adalah Jaringan Saraf Tiruan SOM, maka perlu diuji apakah Algoritma SOM yang diciptakan oleh Teuvo Kohonen telah diprogram dengan benar pada aplikasi ini. Karena iterasi dari proses *training* SOM sangat banyak, tidaklah mungkin pengujian dilakukan sampai tahap akhir proses. Namun karena proses *training* itu berbentuk iterasi, dapat diasumsikan bila iterasi pertama benar, iterasi-iterasi selanjutnya juga benar.

Berikut ini adalah *setting* SOM yang digunakan untuk uji kebenaran proses *training*:

- *Data Training*: NBA Season 2006
- Dimensi Map SOM: 2 x 2
- *Statistics Field* yang digunakan:
  1. *Height*
  2. *Weight*
  3. *Field Goal Percentage*
  4. *3 Point Field Goal Percentage*

Jalannya pengujian adalah sebagai berikut (sesuai dengan algoritma SOM): Setelah langkah 1 dikerjakan, aplikasi melakukan langkah 2, yaitu radomisasi bobot koneksi antara *neuron input* dan *neuron output*. Hasilnya dapat dilihat pada Gambar 8.



Gambar 8: Hasil Randomisasi Bobot

Langkah 3 algoritma adalah *looping* untuk langkah langkah selanjutnya. Pada langkah 4 diambil satu vektor *input* teratas yaitu data dari “Derrick Coleman” dengan statistik sebagai berikut (setelah dinormalisasi):

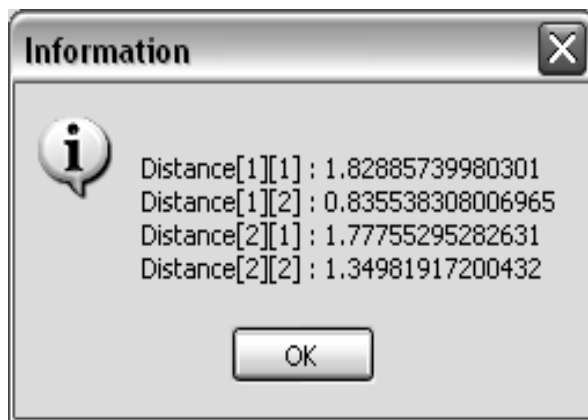
- *Height*: 0.696969696969697
- *Weight*: 0.918650793650794
- *Field Goal*: 1
- *3 Point Goal*: 0.799979600163198

kemudian data tersebut dijadikan *neuron input*.

Pada langkah 5 dihitung jarak antara *neuron input* dengan masing-masing bobot koneksinya pada setiap *Neuron Output (Maps)* menggunakan persamaan 2. Hasil perhitungan manual dapat dilihat pada Tabel 2, sementara hasil perhitungan aplikasi dapat dilihat pada Gambar 9.

**Tabel 2: Hasil Perhitungan *Distance* Secara manual**

Tipe Jarak	Nilai
Utk Map1-1 ( $d_{11}$ )	1.8288
Utk Map1-2 ( $d_{12}$ )	0.8355
Utk Map2-1 ( $d_{21}$ )	1.7776
Utk Map2-2 ( $d_{22}$ )	1.3498

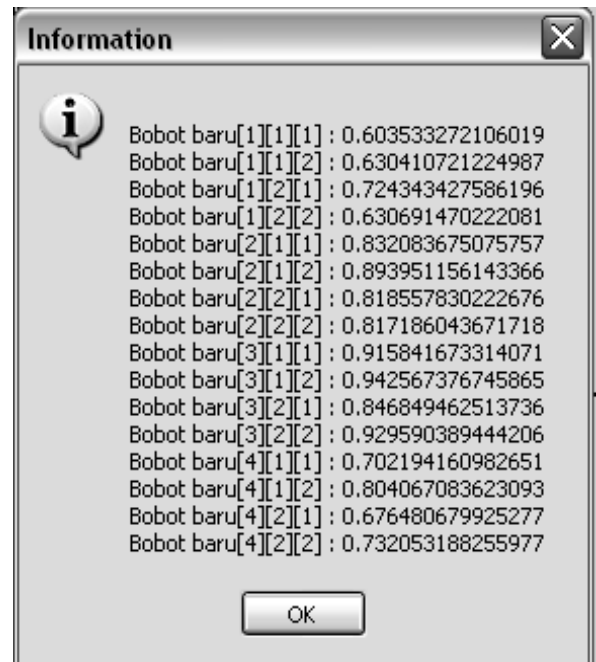


**Gambar 9. Hasil perhitungan *distance* oleh aplikasi**

Pada langkah ke-6 ditentukan “*winning neuron*” untuk vektor *input* 1, yaitu “Map 1-2”, karena map ini yang memiliki jarak/*distance* terkecil dengan vektor *input* 1. Langkah 7 melakukan *update* bobot untuk masing-masing map sesuai dengan persamaan 3. Secara manual perhitungannya adalah sebagai berikut: Bila nilai fungsi *gamma* awal  $\gamma(0) = 1$ , maka untuk vektor *input* 1,  $\gamma(1) = 1/10 = 0.9$ . Selanjutnya perhitungan bobot baru secara *manual* dapat dilihat pada Tabel 3 dan hasil perhitungan dari aplikasi dapat dilihat pada Gambar 10.

**Tabel 3. Hasil Perhitungan bobot baru secara manual**

Nama bobot koneksi	Nilai $h_{ijbc}(t)$	Nilai bobot baru
$\mu_{111}$	0.9622	0.6035
$\mu_{112}$	1	0.6304
$\mu_{121}$	0.9257	0.7243
$\mu_{122}$	0.9622	0.6307
$\mu_{211}$	0.9622	0.8321
$\mu_{212}$	1	0.8939
$\mu_{221}$	0.9257	0.8186
$\mu_{222}$	0.9622	0.8172
$\mu_{311}$	0.9622	0.9158
$\mu_{312}$	1	0.9426
$\mu_{321}$	0.9257	0.8468
$\mu_{322}$	0.9622	0.9296
$\mu_{411}$	0.9622	0.7022
$\mu_{412}$	1	0.8041
$\mu_{412}$	0.9257	0.6765
$\mu_{422}$	0.9622	0.7321



**Gambar 10. Hasil *update* bobot**

Sampai di sini pengujian perbandingan hasil aplikasi dan *manual* selesai dilakukan. Setelah

langkah 7, alur algoritma ber-iterasi (looping) kembali ke langkah 5. Demikian seterusnya sampai bobot koneksi konvergen didapat atau iterasi maksimal dicapai.

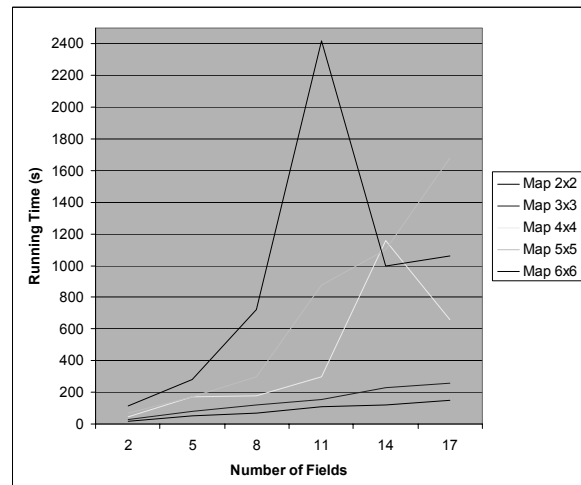
Dari hasil perbandingan perhitungan aplikasi dan *manual*, dapat diasumsikan bahwa program SOM untuk *training* telah dibuat dengan baik. Proses *testing* tidak perlu diuji kebenarannya karena prosesnya sama dengan sebagian dari proses *training*.

### Pengujian Kecepatan Proses

Pada pengujian ini diuji kecepatan aplikasi saat melakukan proses *training* SOM. Proses *testing* tidak perlu dites kecepatan prosesnya karena proses ini hanya melakukan 1 kali iterasi dan tidak perlu melakukan perhitungan untuk *update* bobot koneksi *neuron input* dan *output* (bobot koneksi untuk *testing* di *load* dari file \*.*som*). Hasil uji kecepatan proses dapat dilihat pada Tabel 4 dan Gambar 11.

**Tabel 4. Hasil Pengujian kecepatan proses diurutkan berdasarkan Dimensi Map**

No.	Jumlah Field	Dimensi Map	Jumlah Iterasi	Running Time
1.	2	2x2	27	20s
2.	5	2x2	38	49s
3.	8	2x2	38	1m 10s
4.	11	2x2	45	1m 47s
5.	14	2x2	42	2m 3s
6.	17	2x2	42	2m 28s
7.	2	3x3	26	30s
8.	5	3x3	39	1m
9.	8	3x3	38	1m 59s
10.	11	3x3	38	2m 37s
11.	14	3x3	45	3m 51s
12.	17	3x3	42	4m 18s
13.	2	4x4	31	48s
14.	5	4x4	53	2m 50s
15.	8	4x4	37	2m 59s
16.	11	4x4	46	4m 56s
17.	14	4x4	144	19m 20s
18.	17	4x4	68	10m 58s
19.	2	5x5	34	1m 11s
20.	5	5x5	38	2m 52s
21.	8	5x5	43	4m 59s
22.	11	5x5	94	14m 36s
23.	14	5x5	94	18m 20s
24.	17	5x5	119	27m 59s
25.	2	6x6	34	1m 52s
26.	5	6x6	41	4m 40s
27.	8	6x6	57	12m 3s
28.	11	6x6	162	40m 17s
29.	14	6x6	63	16m 38s
30.	17	6x6	55	17m 39s



**Gambar 11. Grafik Jumlah Field Terhadap Running Time Pada Setiap Map SOM**

Dari analisis terhadap Tabel 4 dan Gambar 11 secara umum terlihat bahwa kenaikan jumlah atribut / *field* dan dimensi map akan mempengaruhi lamanya waktu proses. Namun terjadi anomali/ketidaknormalan di sini yaitu:

- Pada Map 4 x 4, waktu yang dibutuhkan untuk *training* data dengan jumlah *field* 14 lebih tinggi/lambat dibanding jumlah *field* yang lebih banyak.
- Pada Map 6 x 6, waktu yang dibutuhkan untuk *training* data dengan jumlah *field* 11 lebih tinggi/lambat dibanding jumlah *field* yang lebih banyak.

Setelah dilakukan percobaan lebih lanjut, yaitu dengan mengganti *field*/atribut tentang *Offensive Rebounds*, *Defensive Rebounds*, dan *Total Rebounds* dengan *field* yang lain, anomali itu hilang. Dari sini dapat disimpulkan bahwa ada sesuatu pada ketiga *field* tersebut. Setelah dilihat ke dalam *record-recordnya*, ditemukan bahwa ketiga *field* tersebut memiliki sebaran nilai *data* yang banyak dan luas. Hal ini menyebabkan sulitnya *data* yang ada untuk dikelompokkan dalam *cluster-cluster*. Imbas dari problem ini, *training* SOM membutuhkan waktu yang lebih lama untuk mencapai bobot koneksi konvergen.

### Pengujian oleh Calon Pengguna

Pengujian ini digunakan untuk melihat dimensi map SOM mana yang paling cocok untuk digunakan mengelompokkan (*clustering*) data pemain basket NBA yang di *training*. Metode pengujian dilakukan dengan *survey* ujicoba aplikasi kepada beberapa orang responden yang memiliki pengetahuan yang dalam tentang basket dan mempunyai pengalaman selama bertahun-tahun di dalam dunia basket. Daftar responden tersebut dapat dilihat pada Tabel 5. Sementara hasil *survey* dapat dilihat pada Tabel 6.



**Tabel 5. Daftar Responden untuk survey dimensi map SOM yang tepat.**

No	Nama	Status / Jabatan
1.	Paulus	Pelatih Tim Basket U.K. Petra
2.	Raymond	Manager Tim Basket U.K. Petra
3.	Bayu	Asisten Manager Tim Basket U.K. Petra
4.	Oei A Kiat	Pelatih Tim IBL Klub Bima Sakti Malang
5.	Arifin	Pelatih Tim Pemula Klub Bima Sakti Malang
6.	Sayuti	Pelatih Tim Junior Klub Bima Sakti Malang
7.	Lena	Pelatih Tim Junior Putri Klub Sahabat Surabaya
8.	Wiby	Pelatih Tim Basket Klub Basudewo Blitar
9.	Hendri	Pelatih Tim Basket STTS
10.	Budianto	Pelatih Tim Basket Klub Rajawali Sakti Surabaya
11.	Lily Rosanna	Pemain Senior & Kapten Tim Basket Putri U.K.P
12.	Benny	Pemain Senior & Kapten Tim Basket Putra U.K.P
13.	Irfan	Pemain Senior Tim Basket Putra U.K. Petra
14.	Teguh	Pemain Senior Tim Basket Putra U.K. Petra
15.	Edwin	Pemain Senior Tim Basket Putra U.K. Petra
16.	Edward	Pemain Senior Tim Basket Putra U.K. Petra
17.	Steven Harryanto	Pemain Senior Tim Basket Putra U.K. Petra

**Tabel 6. Hasil survey dari 17 responden**

Respon den	Lebar Dimensi Map SOM				
	2x2	3x3	4x4	5x5	6x6
Ke-1	2	3	3	4	5
Ke-2	2	3	3	4	4
Ke-3	2	3	3	4	4
Ke-4	3	3	4	5	5
Ke-5	2	2	4	4	4
Ke-6	2	3	3	4	5
Ke-7	2	2	3	5	5
Ke-8	2	3	4	4	4
Ke-9	3	4	4	5	5
Ke-10	1	2	3	4	5
Ke-11	3	3	3	5	5
Ke-12	2	4	4	4	5
Ke-13	2	3	3	4	5
Ke-14	3	3	4	4	5
Ke-15	2	3	4	5	5
Ke-16	2	2	4	4	5
Ke-17	2	3	3	4	5

Ket. Nilai: 1 → Sangat Tidak Tepat s/d  
5 → Sangat Tepat

Dari hasil rekap survey menggunakan rumus 6 berikut ini:

$$\text{Nilai} = \left( \left( \sum_1^{\text{jml\_responden}} \text{nilai\_responden} \right) / (5 * \text{jml\_responden}) \right) * 100 \% \tag{6}$$

yang dapat dilihat pada Tabel 7, didapat fakta bahwa rata-rata responden sepakat bahwa semakin besar dimensi map SOM, semakin baik mutu cluster yang dihasilkan.

**Tabel 7. Hasil Rekap Survey**

Dimensi Map	Nilai
6 x 6	95%
5 x 5	81%
4 x 4	69%
3 x 3	54%
2 x 2	44%

**KESIMPULAN DAN SARAN**

Kesimpulan yang dapat ditarik adalah sebagai berikut:

1. Aplikasi dapat menampilkan hasil *classification* melalui proses *testing* SOM, dimana pemain yang di tes dikelompokkan ke dalam *cluster* tertentu yang mempunyai kemiripan karakteristik dengannya. Informasi karakteristik *cluster* dan informasi tentang probabilitas posisi pemain-pemain dalam *cluster*, akan memudahkan pengambilan keputusan yang perlu dilakukan oleh pelatih (*coach*) dan pemandu bakat (*scout*).
2. Dari pengujian, dapat disimpulkan bahwa kenaikan jumlah atribut/*field* yang dipakai sebaiknya disertai dengan kenaikan dimensi map pada SOM saat proses *training/competitive learning/ clustering*. Hal ini dimasukkan untuk mendapatkan *cluster-cluster* yang berkualitas.
3. Dari pengujian dapat disimpulkan bahwa, selain bergantung pada jumlah data, lebar dimensi map SOM, jumlah *field*/atribut yang digunakan, kecepatan proses *training* SOM (*clustering*) juga dipengaruhi oleh jenis *field* yang dipakai, sebaran nilainya serta kombinasi dari *field-field* yang digunakan.
4. Dari hasil *survey* terhadap calon pengguna aplikasi yaitu pelatih basket, manager tim, asisten manager tim dan beberapa pemain senior yang dapat memberi masukan kepada pelatih, dapat disimpulkan bahwa bahwa semakin besar dimensi map SOM, semakin baik mutu *cluster* yang dihasilkan.

Saran pengembangan lebih lanjut adalah:

1. Ada baiknya jika aplikasi ini didistribusikan ke dalam bentuk *website*.
2. Mengingat fungsi dari aplikasi ini yang cukup berguna bagi pengambil keputusan pada sebuah tim olahraga permainan, disarankan agar aplikasi ini dapat dimodifikasi untuk cabang olahraga permainan jenis lain seperti sepakbola, *baseball* atau lainnya.

#### DAFTAR PUSTAKA

1. Stiroh, Kevin J. *Playing for Keep: Pay and Performance in NBA*, June 2003, <http://www.nba.com>, February 2006.
2. Data Statistik Pemain, <http://www.basketindonesia.com>, February, 2006.
3. NBA Players Statistics, <http://www.nba.com>, February 2006.
4. Han, Jiawei, and Micheline Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufmann, 2001.
5. Kainulainen, Jukka, *Clustering Algorithms: Basics and Visualization*, Finland: Helsinki University of Technology, 2002.
6. Vesanto, Juha, and Alhoniemi, Esa, *Clustering of the Self Organizing Map*, IEEE, May 2000.
7. Germano, Tom, *Self Organizing Maps*, <http://davis.wpi.edu/~matt/courses/soms/>, March 1999.
8. Guha, Rajarshi, *Introduction of SOM*, Penn State University, 2003.
9. *The Self Organizing Map (SOM) by Teuvo Kohonen*. <http://www.cis.hut.fi/projects/somtoolbox/theory/somalgorithm.shtml>, August, 2006.
10. Kohonen, Teuvo, *The Self Organizing Map*, IEEE, Sept. 1990. Stiroh, Kevin J. *Playing for Keep: Pay and Performance in NBA*, June 2003, <http://www.nba.com>, February 2006.
11. Setiawan, Kuswara, *Paradigma Sistem Cerdas, Artificial Intelligence*, Bayumedia Publishing, 2003.
12. *Kohonen's Self Organizing Maps (SOM)*, <http://www.willamette.edu/~gorr/classes/cs449/Unsupervised/SOM.html>, August, 2006.