

CLUSTERING DATA MAHASISWA MENGGUNAKAN METODE SELF ORGANIZING MAPS UNTUK MENENTUKAN STRATEGI PROMOSI UNIVERSITAS KANJURUHAN MALANG

Nomadeni Fitroh Arno¹

Moh. Ahsan²

¹Teknik Informatika, Universitas Kanjuruhan Malang, denydevil1234@gmail.com

²Teknik Informatika, Universitas Kanjuruhan Malang, ryan.ahsan@gmail.com

Abstrak

Data akademik mahasiswa merupakan data yang dihimpun dari hasil kegiatan proses belajar mengajar selama mengikuti studi di suatu perguruan tinggi. Berdasarkan data yang diperoleh dari Pangkalan Data Pendidikan Tinggi (PDDIKTI), Universitas Kanjuruhan Malang pada 2 tahun terakhir ini terjadi penurunan pendaftaran mahasiswa baru. Untuk menarik minat calon mahasiswa maka dilakukan analisis dari historis data mahasiswa dengan cara melakukan pengolahan data.

Self Organizing Maps (SOM) merupakan salah satu metoda dalam Jaringan Syaraf Tiruan (*Neural Network*) yang menggunakan pembelajaran tanpa pengarahan (*unsupervised learning*). *Input* dari penelitian ini adalah data asal sekolah mahasiswa berdasarkan jurusan yaitu MA, SMA dan SMK, sedangkan *output*-nya adalah dikelompokkan ke dalam 2 *cluster*. Penelitian ini bertujuan untuk membuat aplikasi sistem berbasis *web* dalam menentukan acuan strategi promosi, dengan melakukan *clustering* terhadap sejumlah data mahasiswa berdasarkan asal sekolah (SMA, SMK & MA) menggunakan algoritma *clustering* SOM.

Kata Kunci : Data Mahasiswa, Aplikasi Web, Jaringan Syaraf Tiruan, Self Organizing Maps (SOM).

Abstract

Student academic data is data collected from the results of the teaching and learning activities during studies at a college. Based on data obtained from the database of higher education (PDDIKTI), Kanjuruhan University of Malang on last 2 years the amount of new students decreased. To attract the candidate of students, so the staff analyze from the data history by processing the data.

Self Organizing Maps (SOM) is one of the methods in Neural Networks that use learning without direction (unsupervised learning). Input from this research is original data school students based on majors namely MA, SMA, and SMK, while his output was grouped into 2 clusters. This research aims to create a web-based application system in determining the reference promotion strategy, by performing clustering of a number of student data based on the origin of the school (SMA, SMK & MA) using SOM clustering algorithm.

Keywords : Student Data, Web Applications, Neural Network, Self Organizing Maps (SOM).

1. Pendahuluan

Perguruan tinggi sebagai institusi pendidikan telah memiliki data akademik dan administrasi dalam jumlah yang sangat besar, namun hanya sebagian kecil data tersebut dimanfaatkan (khususnya dalam penyusunan evaluasi diri). Data akademik mahasiswa merupakan data yang dihimpun dari hasil kegiatan proses belajar mengajar selama mengikuti studi di suatu perguruan tinggi. Data tersebut antara lain: data pribadi mahasiswa, data rencana studi, dan data hasil studi (nilai dan indeks prestasi). Jumlah data yang terakumulasi dari tahun ke tahun perlu dilakukan analisis untuk dapat membuka peluang dihasilkannya informasi yang berguna dalam pembuatan alternatif keputusan bagi manajemen perguruan tinggi (Suprawoto, 2016).

Berdasarkan data yang diperoleh dari Pangkalan Data Pendidikan Tinggi (PDDIKTI), jumlah mahasiswa baru Universitas Kanjuruhan Malang (UNIKAMA) pada tahun 2013 berjumlah 12.065 mahasiswa, pada tahun 2014 berjumlah 11.184 mahasiswa sedangkan pada tahun 2015 berjumlah 9.945 mahasiswa. Pada 2 tahun terakhir ini terjadi penurunan pendaftaran mahasiswa baru, pentingnya sebuah strategi promosi untuk menarik minat calon mahasiswa maka dilakukan analisis dari historis data mahasiswa dengan cara melakukan pengolahan data tersebut.

Maka dari itu, pada penelitian kali ini akan mengimplementasikan “Clustering Data Mahasiswa Menggunakan Metode Self Organizing Maps (SOM) Untuk Menentukan Strategi Promosi Universitas Kanjuruhan Malang” yang digunakan untuk membantu pihak admisi Unikama sebagai bahan identifikasi mengolah data mahasiswa Universitas Kanjuruhan Malang agar dapat menentukan strategi promosi berdasarkan asal sekolah mahasiswa yang menjadi target.

2. Tinjauan Pustaka

2.1 Data Mining

Data mining adalah suatu istilah yang digunakan untuk menguraikan penemuan pengetahuan di dalam basis data. Didalamnya juga terdapat proses yang menggunakan teknik statistik, matematika, kecerdasan buatan, dan *machine learning* untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai basis data (Arsyad, 2015).

Secara sederhana, data mining dapat diartikan sebagai proses mengekstrak atau “menggali” pengetahuan yang ada pada sekumpulan data. Banyak orang yang setuju bahwa data mining adalah sinonim dari *Knowledge Discovery in Database* atau yang biasa disebut KDD. Dari sudut pandang yang lain, data mining dianggap sebagai satu langkah yang penting didalam proses KDD. Proses KDD ini terdiri dari langkah-langkah sebagai berikut :

1. *Data Selection*, pengambilan data yang akan dipakai dari sumber data
2. *Data Cleaning*, proses menghapus data yang tidak konsisten dan kotor
3. *Data Transformation*, proses dimana data ditransformasikan menjadi bentuk yang sesuai untuk diproses dalam data mining
4. *Data Mining*, suatu proses yang penting dengan melibatkan metode untuk menghasilkan suatu pola data
5. *Interpretation/Evaluation*, proses untuk menguji kebenaran dari pola data yang mewakili knowledge yang ada didalam data itu sendiri

2.2 Klustering

Baskoro 2010, menyatakan bahwa:

Klustering adalah satu diantara alat bantu pada data mining yang bertujuan untuk mengelompokkan objek-

objek ke dalam kluster-kluster. Kluster adalah sekelompok atau sekumpulan objek-objek data yang similar satu sama lain dalam kluster yang sama dan disimilar terhadap objek-objek data yang berbeda kluster. Objek-objek yang akan dikelompokkan ke dalam satu atau lebih kluster sehingga objek-objek yang berada dalam suatu kluster akan mempunyai kesamaan yang tinggi antara satu dengan yang lainnya.

2.3 Algoritma Self Organizing Maps (SOM)

SOM merupakan perluasan dari jaringan kompetitif yang sering disebut dengan jaringan kohonen. Jaringan kohonen termasuk dalam pembelajaran tak terawasi (*unsupervised learning*). Jaringan ini pertama kali diperkenalkan oleh Teuvo Kohonen pada tahun 1981. Pada jaringan ini, suatu lapisan yang berisi neuron-neuron akan menyusun dirinya sendiri berdasarkan input nilai tertentu dalam suatu kelompok yang dikenal dengan istilah *cluster*. Selama proses penyusunan diri, *cluster* yang memiliki bobot paling cocok dengan pola input (memiliki jarak paling dekat) akan terpilih sebagai pemenang. *Neuron* yang menjadi pemenang beserta *neuron-neuron* tetangganya akan memperbaiki bobot-bobotnya (Ely, 2014).

Terdapat m unit kelompok yang tersusun dalam arsitektur sinyal-sinyal masukan (*input*) sejumlah n . Vektor bobot untuk suatu unit kelompok disediakan dari pola-pola masukan yang tergabung dengan kelompok tersebut. Selama proses pengorganisasian sendiri, unit kelompok yang memiliki vektor bobot paling cocok dengan pola masukan (ditandai dengan jarak *Euclidean* paling minimum) dipilih sebagai pemenang. Unit pemenang dan unit tetangganya diperbaharui bobotnya. Setiap neuron terkoneksi dengan neuron lain yang dihubungkan dengan bobot atau *weight* (Ely, 2014).

Algoritma pembelajaran *unsupervised* pada Kohonen SOM dalam pengelompokan data adalah sebagai berikut:

1. Tetapkan
 - a. Jumlah Variabel
 - b. Jumlah Data
 - c. Jumlah *Cluster*
2. Inisialisasi
 - a. Nilai bobot w_{ij} secara acak dengan nilai (0-1)
 - w_{ij} = bobot dari koneksi antara *node input* ke- i terhadap *node output* ke- j
 - i = nilai *node* pada layer *input*
 - j = nilai *node* pada layer *output*
 - b. Besar ukuran *neighbourhood* awal $N_m(0)$ dengan nilai yang cukup besar tetapi lebih kecil dari jumlah *node* output.
 - m = indeks *node* pemenang
 - $N_m(0)$ = jumlah tetangga / *neighbour* dari *node* pemenang awal
 - c. Parameter $\alpha(t)$ (*learning rate*) dan $\sigma^2(t)$ (koefisien fungsi aktivasi) antara 0 sampai 1
 - d. Parameter $\Omega(\text{epoch})$, yaitu jumlah berapa kali sebuah data dimasukkan ke dalam jaringan untuk proses *training* sebelum ukuran *neighbour* berkurang pada setiap iterasi.

3. Masukkan vektor *input* ke dalam lapisan *input* dan hitung *distance* (d) dari *input* ini ke bobot w dari setiap *node j* dengan persamaan *Euclidean distance*:

$$d_j = \|x - w_j\| = \sqrt{\sum_{i=1}^n (x_i - w_{ij})^2}$$

Dimana:

- x_i = *node input* ke - i
- w_{ij} = bobot dari koneksi antara *node input* ke - i terhadap *node output* ke- j

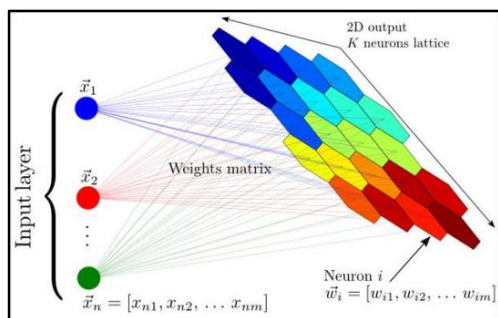
- n = jumlah node pada lapisan *input*
Pilih *node* dengan *distance* terkecil sebagai pemenang m (disebut dengan *best matching unit*)
4. Lakukan *update* vektor bobot pada *node* pemenang m dari *node* tetangganya dengan rumus:
 $w_{ij}(t+1) = w_{ij}(t) + c [x_i = w_{ij}(t)]$
 Dimana:
- $w_{ij}(t+1)$ = bobot koneksi antara *node input* ke- i dan output pada iterasi selanjutnya
 - $C = \alpha(t) \cdot \text{hib}(t)$
 - $\alpha(t) = \exp\left(\frac{-\|r_i - r_m\|}{\sigma^2(t)}\right)$
 - $\text{hib}(t)$ = fungsi tetangga yaitu fungsi *Gaussian* untuk semua *node*; dalam $N_m(t)$
 - $R_i - R_m$ = jumlah *node* atau jarak fisik antara *node* _{i} dan *node* pemenang m dengan jarak *Euclidean*.
5. Lanjutkan dari langkah ketiga untuk Ω epoch, tambahkan 1 pada t , dan kurangi ukuran *neighbourhood*, $\alpha(t)$ dan $\sigma^2(t)$

$$\alpha(t+1) = \alpha(t) * \frac{N_m(t+1)}{N_m(t)}$$

Dimana:

- $\alpha(t+1)$ = *learning rate* pada iterasi berikutnya
- $N_m(t+1)$ = ukuran tetangga pada iterasi berikutnya

Ulangi hingga bobot tidak berubah atau hingga Ω epochs. Setelah itu tes kondisi berhenti.



Gambar 1 Struktur SOM dengan *input* dan output *layer* (Zafar, 2015)

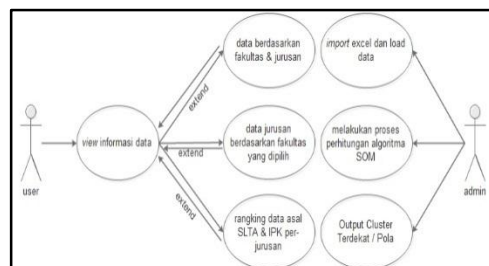
3. Pembahasan

3.1 Analisa Kebutuhan Sistem

Adapun kebutuhan sistem yang dibutuhkan antara lain adalah sebagai berikut :

1. Memberikan *view* informasi pengelompokkan data mahasiswa berdasarkan fakultas, program studi, ranking asal SLTA dan ipk per-jurusan kepada *user*.
2. *Admin* melakukan proses load data dengan menekan tombol yang sudah disediakan oleh sistem.
3. Sistem mampu melakukan perhitungan algoritma SOM secara bertahap melalui setiap iterasi.
4. Sistem mampu membentuk pola menggunakan algoritma SOM.

3.2 Usecase Diagram

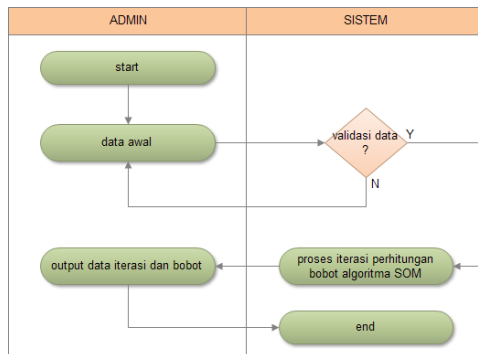


Gambar 2 Usecase diagram

User masuk pada halaman pertama *user* disuguhkan dengan informasi berupa data berdasarkan per-fakultas dan per-jurusan, data jurusan berdasarkan fakultas yang dipilih, dan rangking data berdasarkan asal SLTA & IPK per-jurusan.

Admin menekan tombol *load data* kemudian sistem otomatis melakukan proses *load data* dari *database* lalu dilanjutkan dengan melakukan proses perhitungan algoritma SOM melalui beberapa tahapan dan iterasi untuk mencari masing – masing bobot dan mencari *cluster* yang terdekat. Setelah itu, *admin* dapat melihat *output cluster* terdekat/pola.

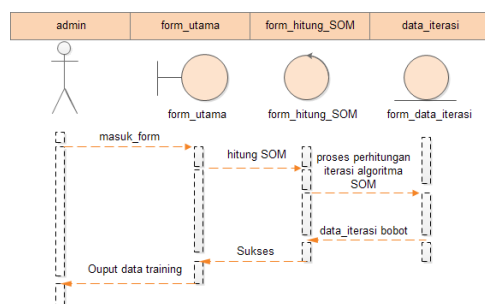
3.3 Activity Diagram Sistem



Gambar 3 Activity diagram perhitungan Algoritma SOM

Dimulai dari *start*, kemudian membaca data awal yang sebelumnya sudah dilakukan proses *load data*. Sistem melakukan validasi data. Setelah itu, sistem melakukan proses iterasi menghitung bobot menggunakan algoritma SOM. Setelah melalui beberapa tahapan iterasi sampai dengan jumlah iterasi, hasil iterasi dan bobot ditampilkan ke dalam *content web*.

3.4 Sequence Diagram Sistem

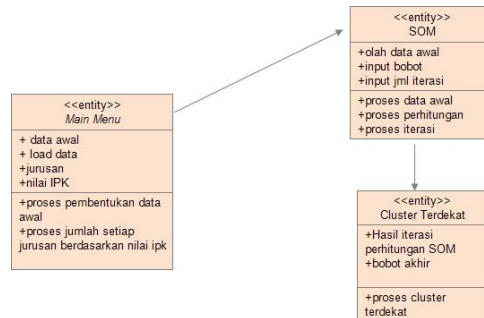


Gambar 4 Sequence Diagram Perhitungan SOM

Proses perhitungan algoritma SOM berdasarkan data *training* / awal. Langkah awal untuk menghitung menggunakan algoritma SOM yaitu melalui parameter jumlah mahasiswa, jurusan, dan asal SLTA. Setelah itu, dilakukan pencarian jumlah data mahasiswa setiap jurusan berdasarkan asal SLTA dengan dikategorikan SMK, SMA, dan MA. Nilai – nilai tersebut

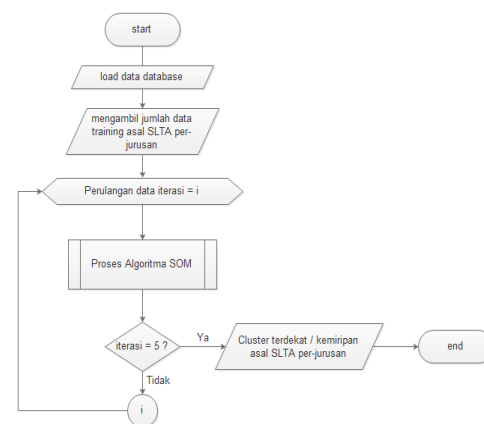
kemudian dihitung dengan bobot awal pada setiap iterasi.

3.5 Class Diagram



Gambar 5 Class Diagram

3.6 Flowchart Sistem



Gambar 6 Flowchart Sistem

Alurnya dari *start* kemudian sistem membaca data dari *database*. Setelah itu, mengambil parameter yang dibutuhkan sebagai data *training* dan melakukan perulangan berdasarkan jumlah iterasi. Dalam *flowchart* diatas menggunakan contoh 5 iterasi kemudian sistem melakukan proses algoritma SOM. Jika sudah mencapai 5 iterasi maka akan mengeluarkan cluster atau pembentukan pola.

3.7 Hasil Dan Pembahasan

- Objek Penelitian : Data mahasiswa berdasarkan asal sekolah (MA, SMA & SMK)

- Metode yang digunakan :
Jaringan syaraf tiruan dengan Algoritma *Self Organizing Maps*
- Bahasa pemrograman : PHP
- Detail data yang akan dilakukan pelatihan
 1. Jumlah data : 18
 2. Jumlah variabel : 3
 3. Jumlah cluster yang diinginkan : 3
 4. Bobot awal : Random
 5. Learning rate (α) : 0,5
 6. Update learning rate ($\alpha(\text{baru})$) : $0,5 * \alpha(\text{lama})$

Berikut adalah sampel 18 data asal sekolah mahasiswa perjurusan yang akan dilakukan pelatihan dengan 3 variabel input yaitu MA, SMA dan SMK.

Tabel 1 Data Training

NO	MA	SMA	SMK
1	17	231	123
2	0	6	2
3	10	112	37
4	4	52	19
5	15	256	198
6	10	226	58
7	11	158	79
8	6	92	23
9	7	199	12
10	3	358	14
11	13	58	28
12	32	441	189
13	9	193	88
14	6	151	22
15	1	92	45
16	6	62	31
17	11	47	85
18	4	56	87

Tabel 2 Data Setelah Dinormalisasi

NO	MA	SMA	SMK
1	0,07	1,00	0,53
2	0,00	1,00	0,33
3	0,09	1,00	0,33
4	0,08	1,00	0,37

5	0,06	1,00	0,77
6	0,04	1,00	0,26
7	0,07	1,00	0,50
8	0,07	1,00	0,25
9	0,04	1,00	0,06
10	0,01	1,00	0,04
11	0,22	1,00	0,48
12	0,07	1,00	0,43
13	0,05	1,00	0,46
14	0,04	1,00	0,15
15	0,01	1,00	0,49
16	0,10	1,00	0,50
17	0,13	0,55	1,00
18	0,05	0,64	1,00

Berdasarkan *input* data normalisasi pada tabel diatas kemudian dilakukan pelatihan jaringan dengan maksimum 10 iterasi dan 18 iterasi dengan menggunakan pemrograman PHP. Berikut adalah hasil *clustering* 18 data dengan maksimum iterasi: 10; *learning rate*: 0,5; dan bobot awal: random. Berikut adalah hasil untuk masing-masing iterasi :

Iterasi 1 data 1

$$d1 = (0.2-0.07)^2 + (0.3-1)^2 + (0.2-0.53)^2 = 0.0169 + 0.49 + 0.1089 = 0.6158$$

$$d2 = (0.2-0.07)^2 + (0.7-1)^2 + (0.1-0.53)^2 = 0.0169 + 0.09 + 0.1849 = 0.2918$$

$$d3 = (0.1-0.07)^2 + (0.3-1)^2 + (1-0.53)^2 = 0.0009 + 0.49 + 0.2209 = 0.7118$$

Terkecil index ke: 1

Iterasi 1 data 2

$$d1 = (0.2-0 + 0.3-1)^2 + (0.2-0.33)^2 = 0.04 + 0.49 + 0.0169 = 0.5469$$

$$d2 = (0.14-0 + 0.85-1)^2 + (0.32-0.33)^2 = 0.0196 + 0.0225 + 0.0001 = 0.0422$$

$$d3 = (0.1-0 + 0.3-1)^2 + (1-0.33)^2 = 0.01 + 0.49 + 0.4489 = 0.9489$$

Terkecil index ke: 1

Iterasi 1 data 3

$$d1 = (0.2-0.09)^2 + (0.3-1)^2 + (0.2-0.33)^2 = 0.0121 + 0.49 + 0.0169 = 0.519$$

$$d2 = (0.07-0.09)^2 + (0.93-1)^2 + (0.33-0.33)^2 = 0.0004 + 0.0049 + 0 = 0.0053$$

$$d3 = (0.1-0.09)^2 + (0.3-1)^2 + (1-0.33)^2 = 0.0001 + 0.49 + 0.4489 = 0.939$$

Terkecil index ke: 1

Iterasi 1 data 4

$$d1 = (0.2-0.08)^2 + (0.3-1)^2 + (0.2-0.37)^2 \\ = 0.0144 + 0.49 + 0.0289 = 0.5333$$

$$d2 = (0.8-0.8)^2 + (0.97-1)^2 + (0.33-0.37)^2 \\ = 0 + 0.0009 + 0.0016 = 0.0025$$

$$d3 = (0.1-0.08)^2 + (0.3-1)^2 + (1-0.37)^2 \\ = 0.0004 + 0.49 + 0.3969 = 0.8873$$

Terkecil index ke: 1

(sampai iterasi ke-10 data ke-18)

Iterasi 10 data 15

$$d1 = (0.2-0.01)^2 + (0.3-1)^2 + (0.2-0.49)^2 \\ = 0.0361 + 0.49 + 0.0841 = 0.6102$$

$$d2 = (0.08-0.01)^2 + (1-1)^2 + (0.37-0.49)^2 \\ = 0.0049 + 0 + 0.0144 = 0.0193$$

$$d3 = (0.09-0.01)^2 + (0.59-1)^2 + (1-0.49)^2 \\ = 0.0064 + 0.1681 + 0.2601 = 0.4346$$

Terkecil index ke: 1

Iterasi 10 data 16

$$d1 = (0.2-0.1)^2 + (0.3-1)^2 + (0.2-0.5)^2 \\ = 0.01 + 0.49 + 0.09 = 0.59$$

$$d2 = (0.08-0.1)^2 + (1-1)^2 + (0.37-0.5)^2 \\ = 0.0004 + 0 + 0.0169 = 0.0173$$

$$d3 = (0.09-0.1)^2 + (0.59-1)^2 + (1-0.5)^2 \\ = 0.0001 + 0.1681 + 0.25 = 0.4182$$

Terkecil index ke: 1

Iterasi 10 data 17

$$d1 = (0.2-0.13)^2 + (0.3-0.55)^2 + (0.2-1)^2 \\ = 0.0049 + 0.0625 + 0.64 = 0.7074$$

$$d2 = (0.08-0.13)^2 + (1-0.55)^2 + (0.37-1)^2 \\ = 0.0025 + 0.2025 + 0.3969 = 0.6019$$

$$d3 = (0.09-0.13)^2 + (0.59-0.55)^2 + (1-1)^2 \\ = 0.0016 + 0.0016 + 0 = 0.0032$$

Terkecil index ke: 2

Iterasi 10 data 18

$$d1 = (0.2-0.05)^2 + (0.3-0.64)^2 + (0.2-1)^2 \\ = 0.0225 + 0.1156 + 0.64 = 0.7781$$

$$d2 = (0.08-0.05)^2 + (1-0.64)^2 + (0.37-1)^2 \\ = 0.0009 + 0.1296 + 0.3969 = 0.5274$$

$$d3 = (0.09-0.05)^2 + (0.59-0.64)^2 + (1-1)^2 \\ = 0.0016 + 0.0025 + 0 = 0.0041$$

Terkecil index ke: 2

Hasil *clustering* dari masing-masing data *input* dengan maksimum 10 iterasi dinyatakan dengan *index* 0 untuk *cluster* 1, *index* 1 untuk *cluster* 2, dan *index* 2 untuk *cluster* 3. Namun hasil yang didapatkan berupa *output* hanya 2 *cluster*.

Tabel 3 Hasil *clustering* 10 iterasi

No	Variabel Input			Cluster		
	MA	SMA	SMK	1	2	3
1	0,07	1,00	0,53		*	
2	0,00	1,00	0,33		*	
3	0,09	1,00	0,33		*	
4	0,08	1,00	0,37		*	
5	0,06	1,00	0,77		*	
6	0,04	1,00	0,26		*	
7	0,07	1,00	0,50		*	
8	0,07	1,00	0,25		*	
9	0,04	1,00	0,06		*	
10	0,01	1,00	0,04		*	
11	0,22	1,00	0,48		*	
12	0,07	1,00	0,43		*	
13	0,05	1,00	0,46		*	
14	0,04	1,00	0,15		*	
15	0,01	1,00	0,49		*	
16	0,10	1,00	0,50		*	
17	0,13	0,55	1,00			*
18	0,05	0,64	1,00			*

Tabel 4 Hasil *clustering* 10 iterasi

No	Variabel Input			Cluster		
	MA	SMA	SMK	1	2	3
1	0,07	1,00	0,53		*	
2	0,00	1,00	0,33		*	
3	0,09	1,00	0,33		*	
4	0,08	1,00	0,37		*	
5	0,06	1,00	0,77		*	
6	0,04	1,00	0,26		*	
7	0,07	1,00	0,50		*	
8	0,07	1,00	0,25		*	
9	0,04	1,00	0,06		*	
10	0,01	1,00	0,04		*	
11	0,22	1,00	0,48		*	
12	0,07	1,00	0,43		*	
13	0,05	1,00	0,46		*	
14	0,04	1,00	0,15		*	
15	0,01	1,00	0,49		*	
16	0,10	1,00	0,50		*	
17	0,13	0,55	1,00			*
18	0,05	0,64	1,00			*

Dari tabel 3 dan 4 diatas dapat dilihat bahwa hasil pelatihan dengan 10 dan 18 iterasi menghasilkan *cluster* dengan anggota yang konsisten.

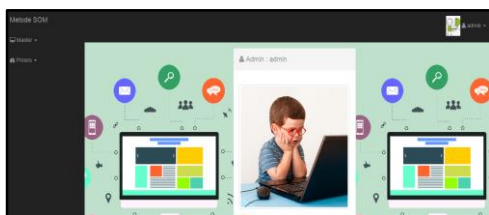
3.8 Desain Interface Sistem Aplikasi Frontend



Gambar 7 Desain Interface Sistem Aplikasi Frontend

Tampilan halaman utama sistem aplikasi *Frontend* ini menampilkan tampilan awal sistem pada sisi *user* saat dibuka. Tampilan menu utama ini digunakan oleh *user* untuk memilih pengelompokan data mahasiswa berdasarkan fakultas maupun berdasarkan jurusan yang dipilih. Terdiri dari menu login, grafik lingkaran berdasarkan fakultas beserta tabel keterangan jumlah mahasiswa / persentasenya dan grafik berdasarkan program studi beserta tabel keterangan jumlah mahasiswa / persentasenya.

3.9 Desain Interface Sistem Aplikasi Backend



Gambar 7 Desain Interface Sistem Aplikasi Backend

Halaman Utama Sistem Aplikasi *Backend* merupakan tampilan utama pada sisi *admin* dimana terdapat beberapa menu sesuai fungsinya masing-masing.



Gambar 9 Perhitungan SOM

Halaman perhitungan *Self Organizing Maps* merupakan halaman yang menampilkan perhitungan data mahasiswa berdasarkan nilai *training* jumlah mahasiswa asal SLTA.

4. Kesimpulan

1. Hasil analisa dan pengolahan data mahasiswa dari aplikasi menggunakan algoritma SOM lebih efisien dalam mengklaster data mahasiswa berdasarkan fakultas, program studi dan asal sekolah (SMA, SMK & MA) dalam bentuk grafik.
2. Implementasi algoritma SOM dapat mengklaster data mahasiswa untuk membantu pihak admisi Unikama dalam menentukan strategi promosi.
3. Setelah didapatkan hasil dari perhitungan *Self Organizing Maps* didapatkan 2 buah *cluster*, *cluster* yang pertama sebanyak 16 buah data dan *cluster* yang kedua hanya 2 buah data.
4. Dari pelatihan yang telah dilakukan anggota yang masuk ke dalam *cluster* pertama adalah :

MA	SMA	SMK
0,07	1,00	0,53
0,00	1,00	0,33
0,09	1,00	0,33
0,08	1,00	0,37
0,06	1,00	0,77
0,04	1,00	0,26
0,07	1,00	0,50
0,07	1,00	0,25
0,04	1,00	0,06
0,01	1,00	0,04
0,22	1,00	0,48
0,07	1,00	0,43
0,05	1,00	0,46
0,04	1,00	0,15
0,01	1,00	0,49
0,10	1,00	0,50

Anggota yang masuk ke dalam *cluster* kedua adalah :

MA	SMA	SMK
0,13	0,55	1,00
0,05	0,64	1,00

Dari tabel tersebut dapat disimpulkan bahwa *cluster* pertama memiliki data lebih dominan dengan variabel SMA dibandingkan dengan *cluster* yang kedua, maka variabel tersebut (SMA) yang akan digunakan sebagai acuan strategi promosi Universitas Kanjuruhan Malang.

5. Saran

1. Diperlukan untuk mengembangkan aplikasi ini dengan metode lain selain *Self Organizing Maps* (SOM), untuk mengetahui hasil yang lebih baik.
2. Perlu adanya pengembangan dari aplikasi ini dengan menambahkan atribut lain seperti kota asal, IPK, dan trend keminatan mahasiswa baru.
3. Diperlukan untuk dikembangkan dengan menggunakan aplikasi lain selain aplikasi yang berbasis *website*.

Daftar Pustaka

Agung, Gregorius. 2014. *MySQL Untuk Pemula*. PT. Elek Media Komputindo Gramedia. Jakarta.

Asropudin, Pipin. 2013. *Kamus Teknologi Informasi Komunikasi*. Titian Ilmu. Bandung

Arsyad Kusumah. 2015. *Penerapan Data Mining Pada Penjualan Produk Benang Di Pt. Bandung Perkasa Jaya Menggunakan Metode Association*. Jurnal Teknik Informatika. Universitas Komputer Indonesia. Bandung.

Baskoro, H. 2010. *Implementasi Algoritma K-Means Menggunakan Data Penyewaan Alat Berat Untuk Melakukan Estimasi Nilai Outcome*. Skripsi Program S1 Fakultas Ilmu Komputer. Universitas Pembangunan Nasional "Veteran". Jakarta.

Ely, Hamzah, Rofiqoh. (2014). *Klasifikasi Kedelai Lokal Berdasarkan Ciri Fisik Menggunakan Jaringan Syaraf Tiruan Algoritma Kohonen Self Organizing Maps*. Jurnal Informatika. Universitas Sebelas Maret. Surakarta.

Jananto Arief. 2010. *Memprediksi kinerja mahasiswa menggunakan teknik data mining (stui kasus data akademik mahasiswa unisbank)*. Tesis Tidak Terpublikasi. Universitas Gajah Mada. Yogyakarta.

Liliweri, Alo. 2011. *Dasar-Dasar Komunikasi Antar Budaya*. Pustaka Belajar. Yogyakarta

Luqman. 2012. *Aplikasi Web Sistem Informasi Penjualan Pada Khazanah Ponsel* Yogyakarta. Sekolah Tinggi Manajemen Informatika Dan Komputer Amikom. Yogyakarta.

Madcoms, Litbang. 2011. *Aplikasi Web Database dengan Dreamweaver dan php-MySQL*. Andi Offset. Yogyakarta

Nugroho, Bunafit. 2013. *Dasar Pemrograman Web PHP-MySQL dengan Dreamweaver*. Gava Media. Yogyakarta.

Riyandwyana Ananda, Suryani Erma, Mukhlason Ahmad. 2012. *Pengembangan Sistem Rekomendasi Peminjaman Buku Berbasis Web Menggunakan Metode Self Organizing Map Clustering Pada Badan Perpustakaan Dan Kearsipan (BAPERSIP) Provinsi Jawa Timur*. Jurnal Teknik Pomits. Surabaya: Institut Sepuluh Nopember (ITS). Edisi Pertama. Volume 1

Suprawoto Totok. 2016. *Klasifikasi Data Mahasiswa Menggunakan Metode K-Means Untuk Menunjang Pemilihan Strategi Pemasaran*. Jurnal Informatika dan Komputer (JIKO). STMIK AKAKOM. Edisi Pertama. Volume 1. Yogyakarta

Zafar Muhammad, *Analisa Program Simulasi Sistem Kendali Modern Self Organizing Maps (SOM)*. 2015. Internet: diunduh tanggal 28 Maret 2016.