

Pemanfaatan Kohonen SOM Satu Dimensi Untuk Pengklasteran Data Kuesioner Calon Mahasiswa Di Surabaya

Steven Nesdi Ghozali, Kuswara Setiawan
Program Studi Sistem Informasi
Universitas Pelita Harapan Surabaya
Surabaya, Indonesia
stv.nesdi@gmail.com
kuswara.setiawan@uphsurabaya.ac.id

Abstrak—Mahasiswa adalah salah satu unsur penting yang ada di dalam sebuah perguruan tinggi. Jika sebuah perguruan tinggi tidak mempunyai mahasiswa atau mempunyai mahasiswa hanya dalam jumlah sedikit, maka perguruan tinggi tersebut tidak akan bisa menjadi populer dan lebih berkualitas. Universitas Pelita Harapan adalah salah satu perguruan tinggi yang ada di Surabaya. Dalam usaha untuk mencari mahasiswa lebih banyak, tim marketing mengadakan penelitian dengan cara membagikan kuesioner kepada para siswa dari SMA swasta di Surabaya kemudian dilakukan penggolongan terhadap hasil kuesioner tersebut. Jika sudah digolongkan, tim marketing akan menyusun strategi yang tepat untuk dilakukan. Salah satu metode yang bisa dipakai untuk melakukan penggolongan adalah dengan memanfaatkan jaringan saraf tiruan Kohonen SOM (*Self Organizing Map*). Setelah digolongkan, setiap golongan haruslah dilihat ciri-cirinya dan tingkat kepercayaan untuk masing-masing cirinya. Untuk bisa melihat tingkat kepercayaan untuk masing-masing ciri dari tiap golongan, diperlukanlah teori tentang belief yang berasal dari Dempster–Shafer Theory. Setelah data hasil kuesioner tersebut digolongkan, dicari ciri-cirinya, kemudian dicari tingkat kepercayaannya, barulah tim marketing bisa menyusun strategi untuk ke depannya supaya bisa merekrut lebih banyak calon mahasiswa. Berdasarkan hasil uji coba yang sudah dilakukan, dapat disimpulkan bahwa aplikasi ini dapat mengklaster data hasil kuesioner dari calon mahasiswa serta mencari ciri dan belief pada setiap klaster.

Kata kunci – Pengklasteran, Kuesioner, *Kohonen Self Organizing Map (SOM)*, Kecerdasan Buatan.

A. PENDAHULUAN

Di Indonesia, pendidikan di perguruan tinggi sangatlah diperlukan. Hal itu dikarenakan di dalam dunia pekerjaan saat ini, hampir seluruh perusahaan mencari para pekerja yang sudah lulus pendidikan perguruan tinggi dengan indeks prestasi (IP) tertentu karena di perguruan tinggi, para mahasiswa sudah diberikan bekal yang cukup banyak baik pengetahuan maupun pengalaman sehingga para mahasiswa bisa lebih siap untuk memasuki dunia pekerjaan daripada mereka yang tidak menjalani jenjang pendidikan di perguruan tinggi. Oleh karena itu, para siswa yang duduk di bangku SMA banyak yang ingin masuk ke perguruan tinggi dan menjadi mahasiswa agar bisa mendapat pekerjaan dan pendapatan yang lebih layak dan berkualitas.

Mahasiswa adalah salah satu unsur penting yang ada di dalam sebuah perguruan tinggi. Jika sebuah perguruan tinggi tidak mempunyai mahasiswa atau mempunyai mahasiswa hanya dalam jumlah sedikit, maka perguruan tinggi tersebut tidak akan bisa menjadi populer dan lebih berkualitas dan jika itu terus berlangsung dalam jangka waktu yang lama, maka perguruan tinggi tersebut bisa tutup. Oleh karena itu, perlu adanya usaha yang sangat keras

untuk bisa mendapatkan mahasiswa karena jumlah perguruan tinggi yang ada di Indonesia terutama kota-kota besar seperti Surabaya dan Jakarta ada banyak sekali. Untuk bisa melakukan usaha tersebut, dibutuhkan peran dari mahasiswa serta tim marketing yang ada di perguruan tinggi tersebut.

Universitas Pelita Harapan adalah salah satu perguruan tinggi yang ada di Surabaya. Agar universitas bisa mendapatkan lebih banyak mahasiswa, maka diadakanlah sebuah penelitian untuk menentukan strategi marketing yang nantinya harus digunakan agar bisa merekrut banyak siswa. Penelitian tersebut dilakukan dengan cara membagikan kuesioner kepada calon-calon mahasiswa yang berasal dari SMA swasta di Surabaya dan saat ini sedang duduk kelas XI dan XII. Kemudian hasil dari kuesioner tersebut akan dikelompokkan ke dalam beberapa golongan dengan ciri-ciri tertentu di setiap golongannya dan kemudian dari golongan-golongan yang ada, dibentuklah strategi marketing yang baru sehingga jumlah calon mahasiswa yang bisa direkrut bisa semakin meningkat dan hal itu akan membuat universitas menjadi semakin maju, berkualitas, dan populer.

Ketika ingin menggolongkan hasil kuesioner tersebut, tidak mungkin semua hasil

kuesioner itu digolongkan secara manual karena akan sangat memakan banyak waktu, membutuhkan usaha yang besar, dan juga hal itu menjadi sangat susah karena tidak bisa diketahui bagaimana caranya jika ingin menggolongkan hasil kuesioner secara manual. Untuk itu, diperlukan metode yang bisa menggolongkan semua hasil kuesioner tersebut ke dalam beberapa golongan dengan sendirinya. Salah satu metode yang bisa dipakai untuk menyelesaikan masalah tersebut adalah dengan memanfaatkan jaringan saraf tiruan Kohonen SOM (Self Organizing Map). Jaringan saraf tiruan Kohonen SOM (Self Organizing Map) merupakan suatu cara pemetaan pola suatu ciri dengan pengaturan yang dilakukan secara otomatis. Dengan memanfaatkan jaringan saraf tiruan Kohonen SOM, pada akhirnya bisa digolongkanlah hasil kuesioner tersebut tetapi hal itu belumlah cukup. Setelah digolongkan, setiap golongan haruslah dilihat ciri-cirinya dan tingkat kepercayaan untuk masing-masing cirinya. Untuk bisa melihat tingkat kepercayaan untuk masing-masing ciri dari tiap golongan, diperlukanlah teori tentang belief yang berasal dari Dempster-Shafer Theory. Teori Dempster-Shafer adalah suatu teori matematika untuk pembuktian berdasarkan belief functions dan plausible reasoning (fungsi kepercayaan dan pemikiran yang masuk akal), yang digunakan untuk mengkombinasikan potongan informasi yang terpisah (bukti) untuk mengkalkulasi kemungkinan dari suatu peristiwa. Teori ini dikembangkan oleh Arthur P. Dempster dan Glenn Shafer.

Setelah data hasil kuesioner tersebut digolongkan, dicari ciri-cirinya, kemudian dicari tingkat kepercayaannya, barulah tim marketing bisa menyusun strategi untuk kedepannya supaya bisa merekrut lebih banyak calon mahasiswa.

B. METODOLOGI PENELITIAN

Metode yang dipakai dalam penelitian ini adalah metode kualitatif. Metode ini dilakukan dengan melakukan pembagian kuesioner. Tim marketing membuat sebuah kuesioner dan kuesioner tersebut dibagikan kepada siswa-siswa yang ada Surabaya dan siswa-siswa yang dibagikan kuesioner tersebut tidak diambil secara sembarangan. Siswa yang dibagikan kuesioner adalah siswa-siswa yang berasal dari berbagai SMA swasta yang berbeda yang ada di Surabaya. Mereka juga semuanya adalah para siswa yang sedang duduk di kelas XI dan XII yang memiliki rencana untuk melanjutkan studi di universitas dan sudah memiliki perspektif mengenai universitas yang akan dipilih.

Untuk isi kuesionernya, kuesioner yang dibuat adalah kuesioner yang sifatnya tertutup dan semua pertanyaan yang ada di dalam kuesioner adalah pertanyaan yang memakai teknik pengukuran dengan Likert's Summated Rating (LSR). Jadi, para siswa yang mengisi kuesioner nantinya akan

menjawab setiap pertanyaan yang ada di dalam kuesioner dengan angka 1 sampai 5. 1 sampai 5 disini melambangkan tingkat kesetujuan. Angka 1 melambangkan sangat tidak setuju, angka 2 melambangkan tidak setuju, angka 3 melambangkan ragu-ragu, angka 4 melambangkan setuju, dan angka 5 melambangkan sangat setuju. Total jumlah pertanyaan yang ada di dalam kuesioner ada delapan belas pertanyaan. Pertanyaan yang ada di dalam kuesioner adalah seperti yang ada di bawah ini :

1. *I prefer to choose university with clear regularities*
2. *I prefer to choose university based on my own decision / preference*
3. *I prefer to choose university based on my parents' consideration*
4. *I prefer to choose university which is able to support my success in the future*
5. *I prefer to choose university with a good collectivism*
6. *I prefer to choose university where students can have a close relationship with lectures / staff*
7. *I prefer to choose university with a good quality*
8. *I prefer to choose university that provides acceleration program*
9. *I prefer to choose university which is reputable and outstanding*
10. *I prefer to choose university based on my friends' suggestion*
11. *I prefer to choose university based on my consideration*
12. *I prefer to choose university which is modern*
13. *I prefer to choose university that provides a good curriculum*
14. *I prefer to choose university that provides international standard*
15. *I prefer to choose university that provides a good holistic education*
16. *I prefer to choose university that can ease their students to graduate*
17. *I prefer to choose university with a good reputation*
18. *I prefer to choose university that provides scholarship*

Untuk jumlah kuesioner yang disebar, jumlah kuesioner yang disebar ke SMA-SMA swasta yang ada di Surabaya ada sekitar seribu dua ratus kuesioner tetapi dari seribu dua ratus kuesioner yang telah dibagikan, kuesioner yang kembali hanya berjumlah delapan ratus dua puluh dan yang bisa dipakai untuk penelitian tim marketing hanya enam ratus sembilan puluh data. Hal itu dikarenakan seratus tiga puluh kuesioner lainnya ternyata ada yang diisi oleh siswa bukan dari SMA swasta, siswa yang berasal dari SMA di luar Surabaya, dan ada juga yang diisi oleh bukan siswa SMA.

Setelah kuesioner dibagikan, diisi, dan dikembalikan, maka dilakukan pemrosesan data. Pemrosesan data yang dimaksud disini adalah menjadikan satu semua kuesioner yang telah dikumpulkan lalu menginputkan jawaban mereka ke dalam tabel yang telah dibuat di Microsoft Excel.

Setelah dilakukan penginputan data ke Ms. Excel, maka selanjutnya data yang ada di dalam tabel yang sudah dibuat di Ms. Excel akan disalin ke dalam sebuah *text file (.txt)* yang nantinya akan dimasukkan ke dalam aplikasi. Di dalam aplikasi, nantinya data-data tersebut akan digolongkan ke dalam empat kluster (jumlah ini didasarkan pada permintaan dari pihak marketing), dilihat ciri tiap kluster dan tingkat kepercayaannya.

Untuk pencarian tiap kluster, dipakailah kohonen SOM. Langkah-langkah pengklusteran dengan metode SOM adalah sebagai berikut :

Langkah 1: Neuron pada lapisan input (neuron input) sebanyak n dinotasikan sebagai $x_1, x_2, x_3, \dots, x_n$ dan neuron pada lapisan output (neuron output) sebanyak m dinotasikan sebagai $y_1, y_2, y_3, \dots, y_m$. Bobot koneksi antara neuron input dan output dinotasikan sebagai W_{ij} .

Langkah 2: Inisialisasi bobot koneksi antara neuron input dan output (W_{ij}) dengan bilangan random antara 0 dan 1.

Langkah 3: Ulangi langkah empat sampai langkah tujuh hingga konvergen (perubahan bobot relatif kecil/lebih kecil dari batas toleransi) atau *cycle* (langkah empat sampai tujuh) telah dilakukan sebanyak jumlah yang telah ditentukan.

Langkah 4: Pilih salah satu vektor input x secara acak (yang juga bilangan real *random* antara 0 dan 1) yang hendak diklusterkan dan diinputkan ke neuron input.

Langkah 5: Hitung jarak vektor input terhadap bobot koneksi di untuk masing-masing neuron output dengan menggunakan rumus:

$$d_j = \sum_{i=1}^n (W_{ij} - X_i)^2 \quad (2.1)$$

Dimana :

d_j : jarak vektor input terhadap koneksi ke output. Semakin kecil angka pada d_j , maka itu berarti jarak antara input dengan output akan semakin pendek.

W_{ij} : bobot koneksi antara neuron input dan output

X : input

Langkah 6: Cari indeks $b = j$ dimana d_j minimum, neuron output b disebut sebagai *best matching unit*.

Langkah 7: Untuk setiap W_{ij} perbaharui bobot koneksi dengan menggunakan rumus :

$$w_{ij}(t + 1) = w_{ij}(t) + \gamma(t)h_{ib}(t)(x_i(t) - w_{ij}(t)) \quad (2.2)$$

Dimana :

$w_{ij}(t + 1)$: bobot koneksi yang baru

$w_{ij}(t)$: bobot koneksi yang lama

$\gamma(t)$: fungsi skalar adaptasi (antara 0-1)

$x_i(t)$: input

$h_{ib}(t)$: fungsi tetangga

Fungsi jarak (fungsi tetangga) SOM ini memakai rumus *phytagoras*. Rumus jarak berubah menjadi :

$$h_{ib}(t) = \exp\left(-\frac{|(r_i - r_b)|^2}{\sigma(t)^2}\right) \quad (2.3)$$

Dimana :

$h_{ib}(t)$: fungsi tetangga (antara 0-1). Fungsi ini berubah dari waktu ke waktu, sema-kin lama semakin kecil, mendekati 0. Artinya, semakin lama pengaruh dari perubahan bobot semakin menyempit hingga pada suatu saat hanya neuron *best matching* saja yang dipengaruhi.

r_i : posisi neuron output ke-i

r_b : posisi neuron *best matching*

$\sigma(t)$: fungsi pengontrol lebar tetangga (bernilai 1-4)

Sedang fungsi $\gamma(t)$ menggunakan :

$$\gamma(t) = \gamma(0)\left(1 - \frac{t}{T}\right) \quad (2.4)$$

Dimana :

$\gamma(t)$: nilai skalar adaptasi. Semakin besar nilai skalar adaptasi, maka semakin cepat bobot koneksi bera-daptasi atau dengan kata lain semakin besar pengaruh vektor input terhadap terhadap perubahan

bobot koneksi yang terjadi. Nilai keluaran fungsi ini berubah dari waktu ke waktu, yaitu semakin lama semakin mendekati 0, sehingga perubahan bobot semakin lama semakin kecil dan vektor-vektor input dapat dipetakan dengan baik

- $\gamma(0)$: nilai skalar adaptasi pertama (mula-mula)
- t : iterasi saat ini
- T : total iterasi yang dilakukan.

Untuk pencarian tingkat kepercayaan, bisa dilakukan pencarian dengan memakai teori Dempster-Shafer. Secara matematika, rumus untuk mencari *belief* adalah dengan menggunakan rumus di bawah ini :

$$Bel(A) = \sum_{B \subseteq A} m(B) \quad (2.5)$$

Dimana :

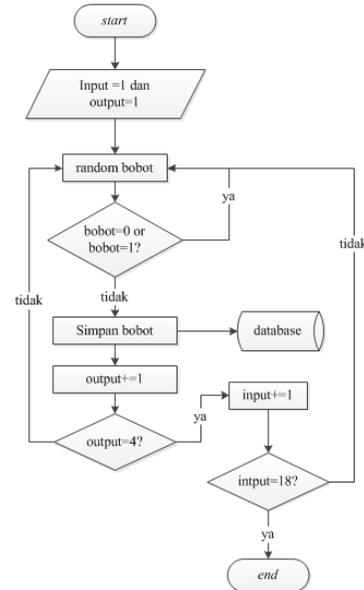
$Bel(A)$: tingkat kepercayaan

$m(B)$: *basic assignment*

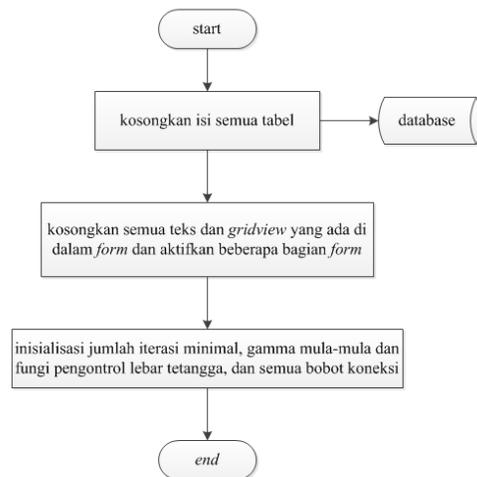
Setelah penyalinan ke *text file (.txt)*, maka didesainlah aplikasi pengklasterannya. Aplikasi terdiri dari tahap *training* dan *testing* dan menggunakan 4 tabel untuk *database*-nya. Tabel-tabel tersebut adalah tabel data *training*, tabel data *testing*, tabel bobot, dan tabel keterangan data. Tabel Data *Training* digunakan untuk menyimpan semua data *training* yang diinputkan oleh *user*. Tabel ini terdiri dari *field* Id , *field* input (1 sampai 18), dan *field* golongan. Tabel Bobot digunakan untuk menyimpan bobot mula-mula dan bobot akhir setelah *training* pada data dilakukan. Tabel Keterangan Data digunakan untuk menyimpan keterangan-keterangan data seperti status *training* dan jumlah data *training*. Tabel Data *Testing*, digunakan untuk menyimpan semua data *testing* yang diinputkan oleh *user*. Tabel ini terdiri dari *field* Id , *field* inputan (1 sampai 18), dan *field* golongan.

Untuk desain aplikasinya, di dalam aplikasi, terdapat dua proses besar yaitu *training* dan *testing*. Dalam proses *training*, mula-mula aplikasi akan melakukan reset dengan sendirinya. Dalam proses reset ini, aplikasi mengosongkan isi tabel data *training* dan bobot di *database*, mengosongkan semua teks dan *gridview* yang ada di dalam *form*, dan juga mengaktifkan beberapa bagian *form*. Setelah itu, aplikasi menginisialisasi jumlah iterasi, gamma mula-mula dan fungsi pengontrol lebar tetangga (lihat kembali rumus 2.3) dan semua bobot koneksi yang digunakan (setelah diinisialisasi, bobot ini disimpan di dalam tabel bobot di *database*).

Untuk bobot koneksi, bobot koneksi yang dibentuk ada 72 (18 input x 4 output) dan merupakan angka *random* antara 0 – 1 (bilangan desimal dengan angka di belakang koma yang berjumlah tiga). Sedangkan untuk gamma, fungsi pengontrol lebar tetangga, dan jumlah iterasi, nilai gamma di-*set* dengan angka 1, jumlah iterasi di-*set* dengan angka 5600, dan fungsi pengontrol lebar tetangga di-*set* dengan angka 4. *Flow chart* untuk proses reset ini bisa dilihat pada gambar 1 dan 2



GAMBAR 1 *Flow chart* pembuatan bobot untuk proses reset



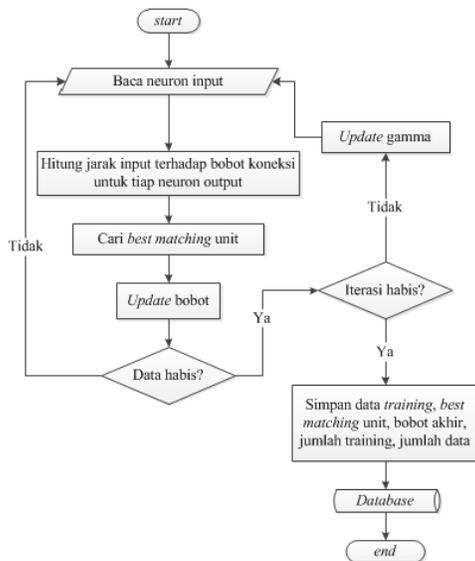
Gambar 2 *Flow chart* proses reset

Setelah reset selesai dilakukan, data hasil kuesioner yang telah dimasukkan ke dalam tabel di Ms. Excel lalu disalin di *text file (.txt)* akan dipilih oleh *user* untuk diklasterkan. Jika *text file* tersebut telah dipilih, maka *file* akan dicoba dibuka dahulu di

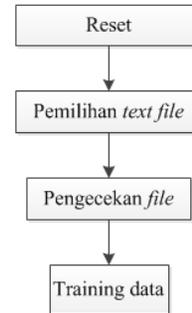
aplikasi untuk memastikan bahwa *text file* bisa diproses. Setelah itu, dilakukan proses *training*.

Pada proses *training*, data yang ada pada *text file* dibaca per baris. Dalam setiap baris, ada 18 angka (merupakan neuron input) yang satu sama lain dipisahkan dengan 'tab'. Untuk setiap baris, setiap ditemukan angka, angka tersebut dimasukkan ke variabel input dan akan dihitung jarak vektor input terhadap bobot koneksi di untuk masing-masing neuron output dengan menggunakan rumus yang ada pada rumus (2.1).

Setelah itu, dicarilah *best matching unit* (lihat kembali langkah enam yang ada di Bab II). Kemudian, semua bobot koneksi di-*update* dengan rumus pada rumus (2.3) yang mana rumus tersebut juga memakai rumus (2.4) dan rumus (2.5). Setelah semua data *training* di-*training* untuk pertama kali, dilakukan pengulangan *training*. Pengulangan *training* terus dilakukan hingga jumlah iterasinya sudah sesuai dengan jumlah iterasi yang diminta. Jika pengulangannya sudah berhenti, aplikasi akan menyimpan seluruh data *training* serta hasil pengklasterannya (berupa *best matching unit* dari setiap data *training*) ke tabel data *training*, bobot akhir ke tabel bobot, dan jumlah iterasi, jumlah data *training* ke dalam tabel keterangan data. Proses *training* secara sederhana bisa dilihat pada *flow chart* proses *training* pada gambar 3 dan keseluruhan dari proses *training* ini bisa dilihat juga dari bagan proses yang ada pada gambar 4.



Gambar 3 Flow chart proses *training*



Gambar 4 Bagan proses keseluruhan tahap *training*

Jika sudah selesai diklasterkan, *user* bisa melihat hasil pengklasterannya serta detail-detailnya (jumlah data yang ada pada suatu kluster, data mana saja yang termasuk dalam suatu kluster, total skala perolehan tiap input pada suatu kluster, dan juga *belief* dari tiap input di dalam suatu kluster). Untuk bisa mendapatkan total skala perolehan tiap input, dilakukan perhitungan dengan cara melakukan penjumlahan nilai input dari semua data dalam satu kluster pada setiap input. Cara untuk mendapat total skala perolehan pada tiap input bisa dilihat pada contoh tabel di bawah ini.

Tabel 1 Cara mencari total skala perolehan pada tiap input

Data	Input 1	Input 2	Input 3
1	2	5	4
4	4	3	2
6	3	2	5
Total skala	9	10	11

Sedangkan untuk mendapatkan nilai *belief*, nilai *belief* bisa didapatkan dengan cara menghitung *basic assignment* pada setiap input. Jika sudah menemukan nilai *basic assignment*, nilai *basic assignment* itulah yang merupakan *belief* dari setiap input. Jika seluruh nilai *belief* tersebut dijumlahkan, maka dihasilkan nilai *belief* = 1 atau 100 persen (nilai dari *belief* bisa bernilai 0 sampai 1 atau 0 sampai 100 persen). Cara mendapatkan nilai dari *basic assignment* adalah dengan membagi salah satu dari skala total dengan total dari total skala yang ada. Contoh untuk menghitung *basic assignment* ada pada tabel 2

Tabel 2 Cara mencari *basic assignment* pada tiap input

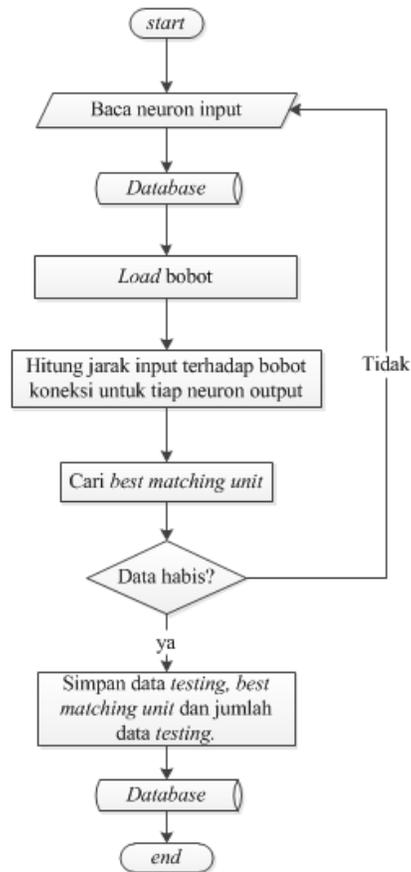
	Input 1	Input 2	Input 3
Skala total	9	10	11
<i>Basic assignment</i>	0,3 atau 30%	0,33 atau 33,33%	0,37 atau 36,67%

Untuk memastikan bahwa aplikasi sudah berjalan dengan baik, maka dilakukan *testing* untuk melihat apakah hasil pengklasteran data yang

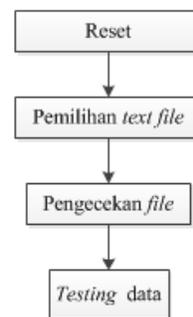
dimasukkan untuk *testing* tersebut sama atau tidak dengan hasil kluster yang sudah terbentuk pada saat *training*. *Testing* ini menggunakan bobot akhir yang telah dibentuk dari hasil *training* dan data yang sama dengan data yang dimasukkan untuk *training* (cara untuk memasukkan data sama dengan memasukkan data waktu *training*, yaitu dengan membuka *text file* kemudian mencoba membukanya). Setelah itu, dilakukan proses *testing*.

Pada proses *testing*, mula-mula, aplikasi akan melakukan reset dengan sendirinya. Dalam proses reset ini, aplikasi mengosongkan isi tabel data *testing*, mengosongkan semua teks dan *gridview* yang ada di dalam *form*, dan juga mengaktifkan beberapa bagian *form*. Setelah itu, aplikasi mengambil bobot yang ada di tabel bobot guna melakukan proses perhitungan jarak vektor input terhadap bobot koneksi. Setelah itu, untuk setiap baris dalam *text file* yang terpilih, setiap ditemukan angka, angka tersebut dimasukkan ke variabel input dan akan dihitung jarak vektor input terhadap bobot koneksi di untuk masing-masing neuron output dengan menggunakan rumus (2.1). Setelah itu, dicarilah *best matching* unit (lihat kembali langkah enam yang ada di Bab II).

Jika sudah dicari *best matching* unitnya, data *testing* serta *best matching* unitnya disimpan di dalam tabel data *testing* di dalam *database* dan jumlah data *testing*-nya disimpan dalam tabel keterangan data. Guna dari penyimpanan data *testing* dan jumlah data *testing* adalah untuk bisa melihat kluster dengan detail. Jadi sifat penyimpanan ini adalah penyimpanan sementara. *Flow chart* dari proses *testing* bisa dilihat pada gambar 5 dan bagan proses keseluruhan tahap *testing* bisa dilihat pada gambar 6.



Gambar 5 Flow chart proses *testing*



Gambar 6 Bagan proses keseluruhan tahap *testing*

Setelah dilakukan *testing* dan ternyata berhasil atau hasilnya sudah sama, *user* bisa melihat hasil pengklusterannya serta detail-detailnya. Jika belum sama, berarti terdapat kesalahan pada aplikasi dan itu berarti aplikasi harus diperbaiki.

Jumlah *form* yang dipakai di dalam aplikasi ini ada 7, yaitu :

- 1) *Form* Menu Utama

Form ini adalah menu utama dari aplikasi. Terdiri dari 3 tombol ('Start clustering', 'See clustering result', dan 'Start testing'). Tombol 'Start clustering' digunakan untuk menampilkan form 'Clustering' yang berfungsi untuk memulai pengklasteran pada data. Tombol 'See clustering result' digunakan untuk untuk menampilkan form 'Clustering result' yang berfungsi untuk melihat klaster yang telah terbentuk dari proses *training*. Sedangkan tombol 'Start testing' digunakan untuk untuk menampilkan form 'Testing' yang berfungsi untuk melakukan *testing*. Pada bagian bawah dari form menu utama ini, terdapat dua label. Label yang ditunjuk oleh nomor satu menunjukkan jumlah data yang dipakai untuk *training* terakhir kali sedangkan label yang ditunjuk oleh nomor dua menunjukkan jumlah iterasi yang dipakai terakhir kali dalam *training* data.

Selain dari 3 tombol yang telah disebutkan di atas, di form menu utama ini juga telah disediakan instruksi awal tentang penggunaan aplikasi ini dan link label 'Read more' yang jika diklik akan memunculkan form 'Help'. Jadi dengan begitu, pengguna aplikasi ini tidak akan bingung untuk memakainya.

Main Menu

Instruction :

- If you see menu 'See clustering result' and 'Start testing' are disabled, then you must do clustering first. After you have done clustering, they will be enabled.
- If you see all menu are enabled, it means you have done clustering before (total iteration and total data that you used at previous training will be added at the bottom of this form). [Read more](#)

Start clustering

See clustering result

Start testing

① Last training : data with iteration ②

Gambar 7 Gambar rancangan form menu utama

2) Form Pengklasteran

Form ini digunakan untuk melakukan *clustering*. Form ini terdiri dari *group box* 'Input' dan *group box* 'Output'.

Untuk *group box* 'Input', terdapat *numeric up down* (ditunjukkan oleh nomor satu) yang berfungsi untuk menentukan total iterasi untuk *training* (bernilai 5600 sebagai *default*); tombol 'Browse' untuk mencari *file* yang akan diklaster melalui *open file dialog*; *text box* 'Path' (ditunjukkan oleh nomor dua) untuk memasukkan *file's path* (didapat dari tombol 'Browse'); tombol 'Try to open file' untuk mencoba membuka *file* dan memasukkan isinya ke dalam *rich text box* yang ditunjukkan oleh nomor tiga; dan tombol 'Start training' untuk memulai *training* (ada pada *flow chart* 3.5).

Untuk *group box* 'Output', terdapat *gridview* (ditunjukkan oleh nomor empat) yang nantinya digunakan untuk melihat hasil *clustering* secara keseluruhan; tombol 'See each cluster' untuk menampilkan form 'Clustering result' yang berfungsi untuk melihat klaster yang telah terbentuk dari proses *training*; dan tombol 'Reset training' untuk mereset ulang *training*.

Selain *group box* input dan output, di bagian bawah form juga terdapat *progress bar* (ditunjuk oleh nomor lima) dan label prosentase (ditunjuk oleh nomor enam) yang akan menunjukkan sampai sejauh mana proses yang berlangsung. Kemudian juga ada label status (ditunjuk oleh nomor tujuh) yang menunjukkan status proses saat ini. Pada awal form dibuka, statusnya adalah 'None'. Ketika tombol 'Try to open file' diklik, maka status berubah menjadi 'Ready', dan saat tombol 'Start training' diklik, maka statusnya berubah menjadi 'Preparing'. Hal itu dikarenakan aplikasi akan melakukan persiapan *training*. Selain itu, *value* dari *progress bar* juga akan bertambah dan prosentase juga bertambah. Setelah persiapan *training* selesai dilakukan, maka tulisan di label status akan berubah menjadi 'Training' dan *value* dari *progress bar* berubah menjadi nol lagi dan label prosentase kembali menjadi 0% kemudian *value* dari *progress bar* bertambah kembali dan prosentase juga bertambah. Setelah aplikasi selesai melakukan proses *training*, maka *progress bar* akan penuh dan prosentase menjadi 100% dan label status berubah menjadi 'Done'.

Input

Total Iteration for training: ①

File's Path: ② Browse... ③

Try to open file Start training

⑤

Output

④

See each cluster Reset training

Progress: ⑤ 0% | Status: None ⑥ ⑦

Gambar 8 Gambar rancangan form pengklasteran

3) Form Testing

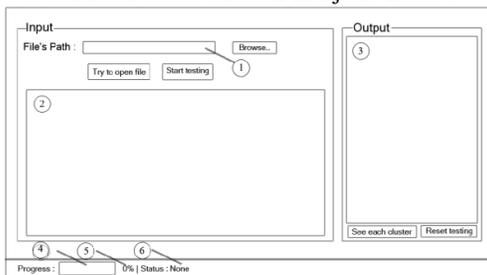
Form ini digunakan untuk melakukan *testing*. Kata 'testing' di sini mempunyai dua pengertian. Yang pertama, kata 'testing' merujuk pada kegunaan form untuk mengetahui apakah hasil klaster yang dihasilkan pada saat *testing* sama atau tidak dengan hasil klaster yang dihasilkan pada saat *training*. Yang kedua, kata 'testing' merujuk pada kegunaan form untuk kedepannya dipakai untuk melihat klaster yang terbentuk dari data yang bukan berasal dari data yang digunakan untuk *training*.

Di dalam form ini, terdapat 2 *group box* (Input dan Output). Di dalam *group box* 'Input', terdapat tombol 'Browse' untuk mencari *file* yang akan di-

test untuk mencari *file* yang akan diklaster melalui *open file dialog*; *text box* 'path' (ditunjukkan oleh nomor satu) untuk memasukkan *file's path* (didapat dari tombol 'Browse'); tombol 'Try to open file' untuk mencoba membuka *file* dan memasukkan isinya ke dalam *rich text box* yang ditunjukkan oleh nomor tiga; dan tombol 'Start testing' untuk memulai *testing*.

Sedangkan di dalam *group box* 'Output', terdapat *gridview* (ditunjukkan oleh nomor empat) yang nantinya digunakan untuk melihat hasil *clustering* secara keseluruhan; tombol 'Reset testing' untuk mereset ulang *form testing*; dan tombol 'See each cluster' untuk menampilkan *form 'Testing result'* yang berfungsi untuk melihat klaster yang telah terbentuk dari proses *testing* untuk melihat secara detail hasil pengklasteran yang dilakukan di *form testing*.

Selain *group box* input dan output, di bagian bawah *form* juga terdapat *progress bar* (ditunjuk oleh nomor empat) dan label prosentase (ditunjuk oleh nomor lima) yang akan menunjukkan sampai sejauh mana proses yang berlangsung. Kemudian juga ada label status (ditunjuk oleh nomor enam) yang menunjukkan status proses saat ini. Pada awal *form* dibuka, statusnya adalah 'None'. Ketika tombol 'Try to open file' diklik, maka status berubah menjadi 'Ready', dan saat tombol 'Start testing' diklik, maka statusnya berubah menjadi 'Testing' dan *value* *progress bar* akan bertambah dan nilai prosentase pada label prosentase juga akan bertambah. Setelah selesai proses *testing*, maka *progress bar* akan penuh dan prosentase menjadi 100% dan label status berubah menjadi 'Done'.



Gambar 9 Gambar rancangan *form testing*

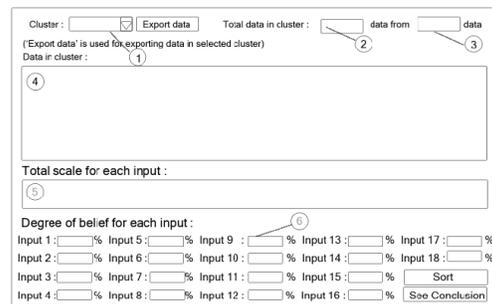
4) Form Clustering Result

Form ini digunakan untuk melihat klaster yang sudah terbentuk dari proses *training* beserta data-datanya. Dari *form* ini juga bisa diketahui besar prosentase 'degree of belief' dari setiap input yang ada.

Form ini bisa diakses dengan dua cara, yaitu dengan mengklik tombol 'See clustering result' yang ada pada *form* menu utama dan juga dengan mengklik tombol 'See each clustering' yang ada pada *form 'Clustering'*.

Form ini terdiri dari *combo box* (ditunjukkan oleh nomor satu) untuk pemilihan klaster; *text box* 'total data in cluster' (ditunjuk oleh nomor dua) untuk melihat total data yang ada pada klaster yang sedang terpilih; *text box* 'total data' (ditunjuk oleh nomor tiga) untuk menunjukkan total data yang ada; *gridview* 'data' (ditunjukkan oleh nomor empat) untuk melihat data tiap klaster; *gridview* 'scale' (ditunjuk oleh nomor lima) untuk melihat total skala perolehan untuk untuk tiap input yang ada; *text box* 'prosentase' (salah satunya ditunjukkan oleh nomor enam) untuk melihat besar prosentase 'degree of belief' untuk tiap input yang ada; tombol 'See conclusion' untuk menampilkan *form 'Conclusion'* untuk melihat kesimpulan pada klaster yang saat ini sedang dipilih oleh *user*; dan tombol 'Sort' yang merupakan tombol *toggle*. Jika tombol ini diklik, maka tulisan di tombol akan berubah dari 'Sort' menjadi 'Unsort'. Jika tombol 'Sort' diklik, maka aplikasi akan mengurutkan bagian *degree of belief* berdasarkan *degree of belief*, mulai dari 'degree of belief' yang terbesar sampai terkecil. Jika tombol 'Unsort' diklik, maka aplikasi akan mengurutkan bagian *degree of belief* berdasarkan urutan input, mulai dari input satu hingga input delapan belas.

Jika *user* ingin meng-*export* data hasil pengklasteran, maka *user* tinggal mengklik tombol 'Export data' dan kemudian aplikasi akan meng-*export* data dalam klaster terpilih kedalam bentuk Excel.



Gambar 10 Gambar rancangan *form training result*

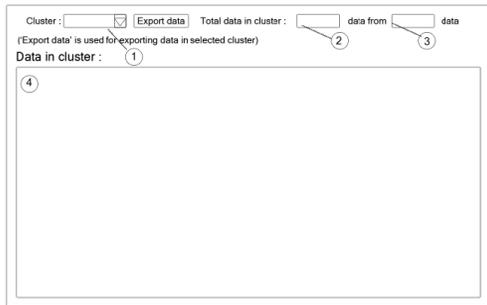
5) Form Testing Result

Form ini kegunaannya hampir sama dengan *form* hasil pengklasteran untuk *training*, hanya saja, di dalam *form* ini, tidak dicantumkan total skala untuk tiap input dan juga tingkat kepercayaan yang ada pada tiap input. *Form* ini digunakan untuk melihat klaster yang sudah terbentuk dari proses *testing* beserta data-datanya. Untuk pengaksesan *form*, *form* ini hanya bisa ditampilkan dengan cara mengklik tombol 'See each clustering' yang ada pada *form 'Testing'*.

Form ini terdiri dari *combo box* (ditunjukkan oleh nomor satu) untuk pemilihan klaster; *text box* 'total data in cluster' (ditunjuk oleh nomor dua)

untuk melihat total data yang ada pada kluster yang sedang terpilih; *text box* 'total data' (ditunjuk oleh nomor tiga) untuk menunjukkan total data yang ada; *gridview* 'data' (ditunjukkan oleh nomor empat) untuk melihat data tiap kluster.

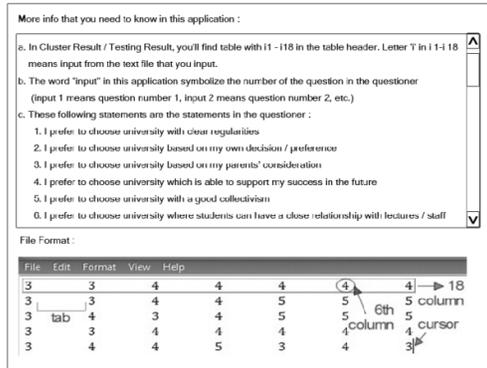
Jika *user* ingin meng-*export* data hasil *testing*, maka *user* tinggal mengklik tombol 'Export data' dan kemudian aplikasi akan men-*export* data dalam kluster terpilih kedalam bentuk Excel.



Gambar 11 Gambar rancangan *form testing result*

6) Form Help

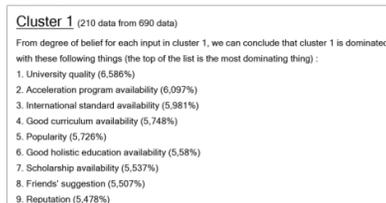
Form ini bisa diakses dengan mengklik *link* label 'read more' yang ada di *form* menu utama. *Form* ini hanya berisi keterangan tambahan untuk *user* agar bisa lebih bisa menggunakan aplikasi.



Gambar 12 Gambar rancangan *form help*

7) Form Conclusion

Form ini berisi kesimpulan dari kluster yang saat ini sedang dipilih oleh *user*.



Gambar 13 Gambar rancangan *form conclusion*

C. HASIL DAN PEMBAHASAN

Sebelum dilakukan pengujian terhadap aplikasi ini, terlebih dahulu dilakukan pelatihan jaringan *SOM* satu dimensi yang telah dibuat. Kemudian setelah bobot ter-*update* dan kemudian disimpan ke *database*, dilakukan uji coba dengan cara memasukkan data yang sama dengan data yang digunakan untuk *training* untuk menguji hasilnya.

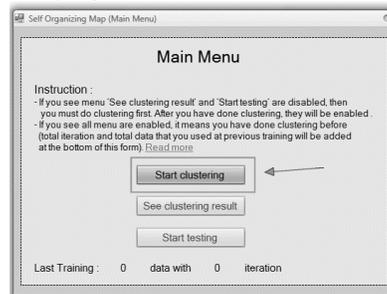
Sebelum melakukan *training*, mula-mula *user* dibawa oleh aplikasi menuju *form* menu utama seperti gambar 14.



Gambar 14 Gambar *Form* Menu Utama

Karena *user* belum pernah melakukan *training* sebelumnya, maka tombol yang aktif hanyalah tombol 'Start clustering' dan untuk bagian bawah *form*, jumlah data dan iterasi hanya betuliskan angka nol dikarenakan di *database* belum ada data apapun beserta keterangannya. Karena yang aktif hanya tombol 'Start clustering' maka *user* hanya bisa melakukan *training* data untuk saat ini. Jika *user* ingin mengetahui lebih lanjut mengenai aplikasi ini, *user* bisa mengklik *link* label 'read more' untuk memunculkan *help*.

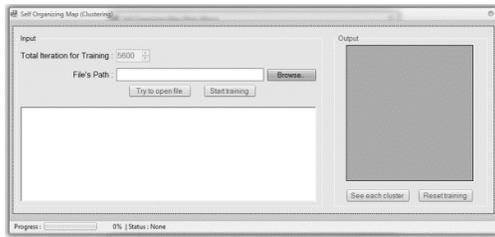
Untuk bisa melakukan pelatihan, *user* terlebih dahulu diharuskan mengklik tombol 'Start clustering' yang ada di *form* menu utama.



Gambar 15 Langkah untuk melakukan pelatihan jaringan

Ketika tombol 'Start clustering' diklik, maka aplikasi akan masuk ke dalam *form* 'Clustering' dan langsung mereset aplikasi kembali ke *setting* awal dan memunculkan notifikasi berupa

message box bahwa aplikasi sudah selesai direset. Jika tombol 'OK' pada notifikasi diklik, maka aplikasi akan menampilkan tampilan form untuk clustering.

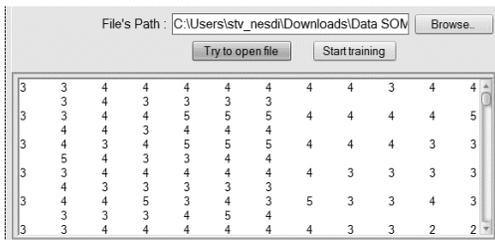


Gambar 16 Tampilan form 'clustering'

Dalam form ini, user harus memilih file dengan cara mengklik tombol 'Browse'. Ketika tombol 'Browse', aplikasi akan membuka open file dialog dan user memilih file.

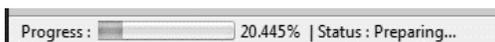
Pada open file dialog ini, user memilih file (hanya text file berformat .txt saja yang bisa dipilih) yang akan dipakai untuk training.

Setelah memilih file, user kembali ke form 'Clustering'. Di form 'Clustering', user diharuskan mencoba untuk membuka file terlebih dahulu dengan mengklik tombol 'try to open file' untuk melihat apakah file yang ingin diproses sudah benar atau tidak.



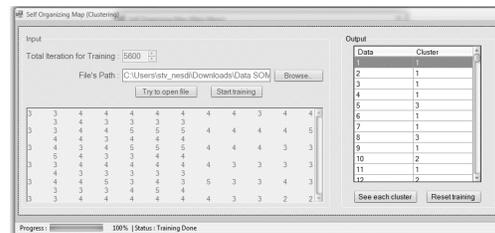
Gambar 17 Tampilan saat file dibuka

Jika file yang dimasukkan sudah benar, maka user bisa mengklik tombol 'Start training' untuk melakukan training data. Ketika tombol tersebut diklik, aplikasi memunculkan notifikasi berupa message box yang memberitahukan bahwa user harus menunggu beberapa saat untuk pemulaian training karena aplikasi sedang melakukan persiapan. Setelah itu, aplikasi akan melakukan persiapan dan tandanya bisa dilihat pada status dan progress bar pada form. Setelah persiapan selesai, maka aplikasi memunculkan notifikasi yang memberitahukan bahwa persiapan sudah selesai dan selanjutnya memulai proses training.



Gambar 18 Tampilan progress dan status bar

Dalam melakukan training, aplikasi membutuhkan waktu beberapa lama untuk melakukan training sampai selesai sehingga user harus menunggu beberapa saat (prosesnya bisa dilihat pada gambar 18 juga hanya saja statusnya menjadi training). Setelah selesai melakukan training, aplikasi memunculkan notifikasi berupa message box yang memberitahukan bahwa proses training telah selesai dilakukan. Jika tombol 'OK' diklik, maka tampilan form 'Clustering' akan menjadi seperti pada gambar 19.



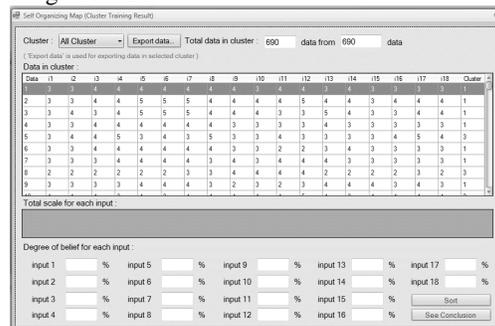
Gambar 20 Tampilan setelah training dilakukan

Pada tampilan sekarang, user sudah tidak bisa melakukan training data lagi karena group box 'Input' sudah tidak aktif lagi. Jika ingin melakukan training ulang, maka user hanya perlu mengklik tombol 'Reset Training' yang ada pada group box 'Output'. Ketika tombol 'Reset Training' diklik, aplikasi menanyakan terlebih dahulu apakah user yakin ingin melakukan reset atau tidak. Jika user memilih 'tidak', tampilan aplikasi akan tetap seperti gambar 20. Tetapi jika user memilih 'ya', aplikasi akan menampilkan notifikasi peresetan training dan kemudian tampilannya kembali seperti tampilan pada gambar 16.

Pada gambar 20, user sudah bisa melihat hasil clustering yang ada (terdapat pada table yang ada di bagian kanan form).

Meski sudah bisa melihat hasil clustering dari proses training, tetapi user belum bisa melihat hasil clustering yang ada secara detail. Jika ingin melihat hasil clustering secara detail, user bisa mengklik tombol 'See each cluster' pada form.

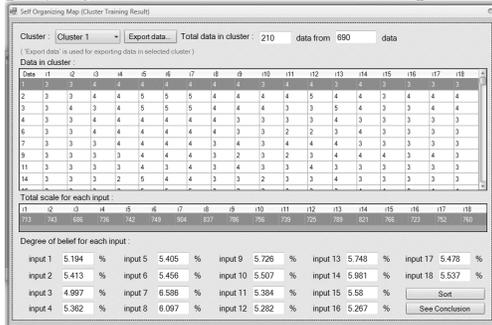
Setelah tombol 'See each cluster' diklik, aplikasi membuka form 'Training Result' seperti pada gambar 21.



Gambar 4.21 Tampilan form 'Training Result'

Pada form, 'Training Result', user mula-mula akan melihat hasil clustering secara keseluruhan seperti yang ditunjukkan pada gambar 20 hanya saja di form ini, ketika user melihat kluster secara keseluruhan, user bisa melihat jumlah data training, detail tiap data training serta keterangan yang menunjukkan data training tersebut termasuk ke dalam kluster yang mana. Jika user ingin melihat detail kluster pada salah satu kluster, maka user harus mengklik *combo box* 'cluster' dan memilih salah satu kluster yang ada pada *combo box* 'cluster'.

Setelah selesai memilih salah satu kluster, aplikasi akan menampilkan keterangan kluster secara detail mulai dari jumlah data yang ada pada kluster terpilih, detail semua data training yang ada dalam kluster terpilih, jumlah skala yang diperoleh tiap input dari dari semua data training yang termasuk dalam kluster terpilih, serta *degree of belief* dari tiap input yang diurutkan berdasarkan urutan input.

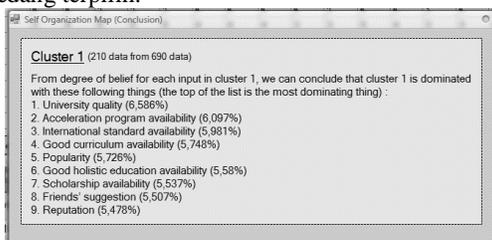


Gambar 22 Tampilan setelah user memilih salah satu kluster

Pada bagian *degree of belief*, jika user ingin mengurutkan bagian *degree of belief* berdasarkan *degree*-nya, maka user bisa mengklik tombol 'Sort' yang ada pada bagian kanan bawah dari form.

Setelah tombol 'Sort' tersebut diklik, maka user bisa melihat bagian *degree of belief* tiap input sudah terurut mulai dari *degree* yang terbesar hingga terkecil dan tombol 'Sort' berubah menjadi tombol 'Unsort' yang fungsinya untuk mengurutkan kembali *degree of belief* berdasarkan urutan input.

Jika tombol 'See Conclusion' diklik, maka akan ditampilkan kesimpulan dari kluster yang sedang terpilih.



Gambar 23 Contoh kesimpulan dari kluster yang terpilih

Jika tombol 'Export data' diklik, maka aplikasi akan memunculkan *Save File dialog* dan kemudian user akan memilih lokasi penyimpanan hasil peng-export-an. Setelah itu, aplikasi akan menyimpan hasil pengklasteran ke dalam file Excel.

Jika form 'Training Result' ini ditutup, maka aplikasi akan menampilkan kembali form 'Clustering' dengan tampilan seperti pada gambar 20. Jika form 'Clustering' ditutup, maka user akan kembali ke menu utama dengan tombol 'See clustering result' dan tombol 'Start testing' yang sudah aktif. Selain itu, di bagian bawah dari form menu utama, pada label untuk jumlah data training dan jumlah iterasi yang dilakukan, angkanya sudah bukan angka nol lagi. Angkanya sudah menjadi angka yang sesuai dengan jumlah data training yang dipakai selama proses training dan juga jumlah iterasi yang diinputkan oleh user pada form 'Clustering'.



Gambar 24 Tampilan menu utama setelah training dilakukan

Jika user ingin melihat kembali hasil kluster yang telah dihasilkan sebelumnya di proses training, maka user bisa mengklik tombol 'See clustering result'. Jika tombol itu diklik, maka aplikasi akan menampilkan kembali form 'Clustering Result'. Jika user mengklik kembali tombol 'Start training' kembali, maka aplikasi akan memberi notifikasi (berupa *message box*) bahwa sebelumnya sudah dilakukan training dan juga aplikasi menanyakan apakah user ingin melakukan training ulang atau tidak.

Jika user memilih 'tidak', maka aplikasi akan membawa user menuju ke form menu utama. Jika user memilih 'ya', maka aplikasi akan mereset aplikasi kemudian menampilkan notifikasi peresetan training dan membawa user menuju form 'Clustering'. Ketika user ingin membatalkan training ulang saat user telah memutuskan melakukan training ulang dan masuk form 'Clustering', maka jika form 'Clustering' ditutup, user akan dibawa oleh aplikasi menuju tampilan form seperti pada gambar 14.

Jika user menutup form menu utama dan membuka kembali aplikasi, maka tampilan form menu utama sudah seperti gambar 24.

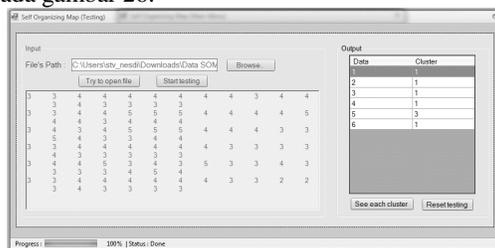
Jika *user* ingin mengetahui kluster dari data *testing* (terdiri dari 18 input juga untuk tiap data dan dipisahkan dengan 'tab' untuk tiap input), *user* bisa mengklik tombol 'Start testing'. Ketika tombol 'Start testing' diklik, maka aplikasi akan menampilkan *form* 'Testing' tetapi sebelumnya memunculkan notifikasi (merupa *message box*) peresatan *form* 'Testing' terlebih dulu. Jika tombol 'OK' diklik, maka *user* akan dibawa menuju *form* 'Testing'.



Gambar 25 Tampilan *form testing*

Untuk melakukan *testing*, *user* terlebih dulu memilih *file* yang akan dites. Caranya yaitu dengan mengklik tombol 'Browse'. (Gambar 4.4)

Jika tombol 'Browse' diklik, akan muncul *open dialog box* dan *user* diminta memilih *file* (*text file* berformat *.txt*) untuk *testing* kemudian mencoba untuk membuka *file* yang telah dipilih. Jika *file* yang dimasukkan sudah benar, maka *user* bisa mengklik tombol 'Start testing' untuk melakukan *testing* data. Dalam melakukan *testing*, aplikasi membutuhkan waktu beberapa lama untuk melakukan *testing* sampai selesai sehingga *user* harus menunggu beberapa saat. Setelah selesai melakukan *testing*, aplikasi memunculkan notifikasi berupa *message box* yang memberitahukan bahwa proses *testing* telah selesai dilakukan. Jika tombol 'OK' diklik, maka tampilan *form* 'Testing' akan menjadi seperti pada gambar 26.



Gambar 26 Tampilan *form testing* setelah selesai proses *testing*

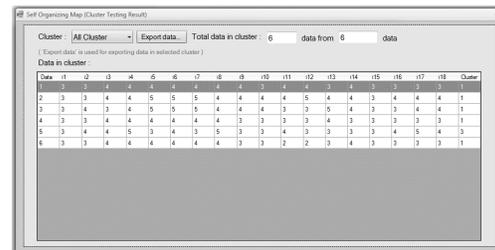
Pada tampilan sekarang, *user* sudah tidak bisa melakukan *testing* data lagi karena *group box* 'Input' sudah tidak aktif lagi. Jika ingin melakukan *testing* ulang, maka *user* hanya perlu mengklik tombol 'Reset testing' yang ada pada *group box* 'Output'. Ketika tombol 'Reset testing' diklik, aplikasi menanyakan terlebih dahulu apakah *user* yakin ingin melakukan reset atau tidak. Jika *user* memilih 'tidak', tampilan aplikasi akan tetap seperti

gambar 26. Tetapi jika *user* memilih 'ya', aplikasi akan menampilkan notifikasi peresetan *form testing* dan kemudian tampilannya kembali seperti tampilan pada gambar 25.

Pada gambar 26, *user* sudah bisa melihat hasil *clustering* yang ada (terdapat pada table yang ada di bagian kanan *form*).

Meski sudah bisa melihat hasil *clustering* dari proses *testing* yang ada, tetapi *user* belum bisa melihat hasil *clustering* secara detail. Jika ingin melihat hasil *clustering* secara detail, *user* bisa mengklik tombol 'See each cluster' pada *form*.

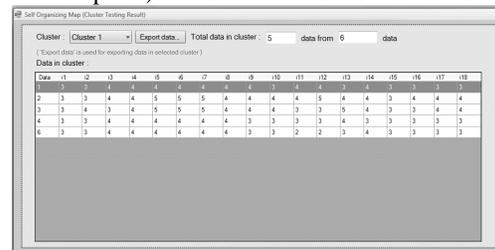
Setelah tombol 'See each cluster' diklik, aplikasi membuka *form* 'Testing Result'.



Gambar 27 Tampilan *form* 'Testing Result'

Pada *form*, 'Testing Result', *user* mula-mula akan melihat hasil *clustering* secara keseluruhan seperti yang ditunjukkan pada gambar 26 hanya saja di *form* ini, ketika *user* melihat kluster secara keseluruhan, *user* bisa melihat jumlah data *testing*, detail tiap data *testing* serta keterangan yang menunjukkan data *testing* tersebut termasuk ke dalam kluster yang mana. Jika *user* ingin melihat detail kluster pada salah satu kluster, maka *user* harus mengklik *combo box* 'cluster' dan memilih salah satu kluster yang ada pada *combo box* 'cluster'.

Setelah selesai memilih salah satu kluster, aplikasi akan menampilkan keterangan kluster secara detail (jumlah data yang ada pada kluster terpilih serta detail semua data *testing* yang ada dalam kluster terpilih).



Gambar 28 Tampilan *form testing result* setelah *user* memilih kluster

Jika tombol 'Export data' diklik, maka aplikasi akan memunculkan *Save File dialog* dan kemudian *user* akan memilih lokasi penyimpanan

hasil peng-export-an. Setelah itu, aplikasi akan menyimpan hasil pengklasteran ke dalam file Excel.

Jika form 'Testing Result' ini ditutup, maka aplikasi akan menampilkan kembali form 'Testing' dengan tampilan seperti pada gambar 26. Jika form 'Testing' ditutup, maka user akan kembali ke menu utama dengan tampilan seperti pada gambar 24.

Setelah aplikasi digunakan oleh tim marketing dan data hasil kuesioner yang dibagikan kepada para siswa SMA dipakai untuk proses pengklasteran, diperoleh hasil berupa berhasil tersebarinya data-data yang telah dimasukkan tersebut ke dalam empat klaster berbeda yang mana keempat klaster tersebut telah ditentukan sebelumnya.

Pada setiap klaster, berdasarkan hasil dari perhitungan untuk memperoleh tingkat kepercayaan dari tiap input, akhirnya pada tiap klaster diambil sembilan input yang mendominasi. Diambilnya kesembilan input itu dikarenakan jika tingkat kepercayaan dari masing-masing sembilan input tersebut dijumlah, maka total nilai *belief* nya sudah mencapai lebih dari 50%. Karena sudah lebih dari 50%, dapat dikatakan bahwa kesembilan input yang diambil tersebut sudah bisa cukup dipercaya mendominasi klaster. Berikut ini adalah hasil analisis terhadap tiap klaster yang ada.

a. Klaster 1

Di dalam klaster pertama, terdapat 210 data. Semua data yang termasuk di dalam klaster yang pertama ini dapat dilihat pada lampiran C. Untuk perolehan total skala tiap input dari klaster yang pertama ini dapat dilihat pada lampiran D. Dari total skala tiap input yang ada, dihitunglah tingkat kepercayaan dari tiap input yang mempengaruhi klaster pertama. Hasil dari perhitungan untuk memperoleh tingkat kepercayaan untuk tiap input bisa dilihat pada lampiran E.

Berdasarkan hasil pencarian *belief* yang ada pada tiap input, dapat disimpulkan bahwa pada klaster satu, sembilan input yang mendominasi (diurutkan berdasarkan nilai *belief* yang paling tinggi) pada klaster ini adalah input ke-7 dengan nilai *belief* sebesar 6,586%, input ke-8 dengan nilai *belief* sebesar 6,097%, input ke-14 dengan nilai *belief* sebesar 5,981%, input ke-13 dengan nilai *belief* sebesar 5,748%, input ke-9 dengan nilai *belief* sebesar 5,726%, input ke-15 dengan nilai *belief* sebesar 5,58%, input ke-18 dengan nilai *belief* sebesar 5,537%, input ke-10 dengan nilai *belief* sebesar 5,507%, dan input ke-17 dengan nilai *belief* sebesar 5,478%. Total dari semua nilai *belief* dari kesembilan input tersebut adalah 52,24%.

Dari analisa terhadap input apa saja yang mendominasi di klaster satu, bisa disimpulkan bahwa klaster satu ini adalah klaster yang terdiri dari

dari siswa-siswa yang lebih mengutamakan hal-hal berikut dalam memilih universitas (diurut mulai dari nilai *belief* yang paling besar) :

- a. Kualitas universitas secara keseluruhan
- b. Ada tidaknya program akselerasi
- c. Ada tidaknya standar internasional
- d. Kurikulum universitas
- e. Kebaikan nama dan keternukaan universitas
- f. Ada tidaknya edukasi holistik
- g. Ada tidaknya beasiswa
- h. Sugesti teman dan sahabat
- i. Reputasi

b. Klaster 2

Di dalam klaster kedua, terdapat 328 data. Semua data yang termasuk di dalam klaster yang kedua ini dapat dilihat pada lampiran C. Untuk perolehan total skala tiap input dari klaster yang kedua ini dapat dilihat pada lampiran D. Dari total skala tiap input yang ada, dihitunglah tingkat kepercayaan dari tiap input yang mempengaruhi klaster kedua. Hasil dari perhitungan untuk memperoleh tingkat kepercayaan untuk tiap input bisa dilihat pada lampiran E.

Berdasarkan hasil pencarian *belief* yang ada pada tiap input, dapat disimpulkan bahwa pada klaster dua, sembilan input yang mendominasi (diurutkan berdasarkan nilai *belief* yang paling tinggi) pada klaster ini adalah input ke-1 dengan nilai *belief* sebesar 6,126%, input ke-3 dengan nilai *belief* sebesar 6,028%, input ke-2 dengan nilai *belief* sebesar 5,933%, input ke-14 dengan nilai *belief* sebesar 5,671%, input ke-8 dengan nilai *belief* sebesar 5,654%, input ke-4 dengan nilai *belief* sebesar 5,65%, input ke-15 dengan nilai *belief* sebesar 5,594%, input ke-9 dengan nilai *belief* sebesar 5,547%, dan input ke-13 dengan nilai *belief* sebesar 5,534%. Total dari semua nilai *belief* dari kesembilan input tersebut adalah 51,737%.

Dari analisa terhadap input apa saja yang mendominasi di klaster dua, bisa disimpulkan bahwa klaster dua ini adalah klaster yang terdiri dari dari siswa-siswa yang lebih mengutamakan hal-hal berikut dalam memilih universitas (diurut mulai dari nilai *belief* yang paling besar) :

- a. Aturan universitas yang berlaku
- b. Tanggapan orang tua
- c. Keputusan pribadi
- d. Ada tidaknya standar internasional
- e. Ada tidaknya program akselerasi
- f. Jaminan bahwa universitas bisa mendukung kesuksesan mahasiswanya di masa depan
- g. Ada tidaknya edukasi holistik
- h. Kebaikan nama dan keternukaan universitas
- i. Kurikulum universitas

c. Klaster 3

Di dalam kluster ketiga, terdapat 107 data. Semua data yang termasuk di dalam kluster yang ketiga ini dapat dilihat pada lampiran C. Untuk perolehan total skala tiap input dari kluster yang ketiga ini dapat dilihat pada lampiran D. Dari total skala tiap input yang ada, dihitunglah tingkat kepercayaan dari tiap input yang mempengaruhi kluster ketiga. Hasil dari perhitungan untuk memperoleh tingkat kepercayaan untuk tiap input bisa dilihat pada lampiran E.

Berdasarkan hasil pencarian *belief* yang ada pada tiap input, dapat disimpulkan bahwa pada kluster tiga, sembilan input yang mendominasi (diurutkan berdasarkan nilai *belief* yang paling tinggi) pada kluster ini adalah input ke-11 dengan nilai *belief* sebesar 6,976%, input ke-13 dengan nilai *belief* sebesar 5,963%, input ke-12 dengan nilai *belief* sebesar 5,933%, input ke-14 dengan nilai *belief* sebesar 5,801%, input ke-10 dengan nilai *belief* sebesar 5,742%, input ke-15 dengan nilai *belief* sebesar 5,684%, input ke-8 dengan nilai *belief* sebesar 5,625%, input ke-9 dengan nilai *belief* sebesar 5,625%, dan input ke-18 dengan nilai *belief* sebesar 5,463%. Total dari semua nilai *belief* dari kesembilan input tersebut adalah 52,812%.

Dari analisa terhadap input apa saja yang mendominasi di kluster tiga, bisa disimpulkan bahwa kluster tiga ini adalah kluster yang terdiri dari dari siswa-siswa yang lebih mengutamakan hal-hal berikut dalam memilih universitas (diurut mulai dari nilai *belief* yang paling besar) :

- a. Tanggapan pribadi
- b. Kurikulum universitas
- c. Kemodernan universitas
- d. Ada tidaknya standar internasional
- e. Sugesti teman dan sahabat
- f. Ada tidaknya edukasi holistik
- g. Ada tidaknya program akselerasi
- h. Kebaikan nama dan ketermukaan universitas
- i. Ada tidaknya beasiswa

d. Kluster 4

Di dalam kluster keempat, terdapat 45 data. Semua data yang termasuk di dalam kluster yang keempat ini dapat dilihat pada lampiran C. Untuk perolehan total skala tiap input dari kluster yang keempat ini dapat dilihat pada lampiran D. Dari total skala tiap input yang ada, dihitunglah tingkat kepercayaan dari tiap input yang mempengaruhi kluster keempat. Hasil dari perhitungan untuk memperoleh tingkat kepercayaan untuk tiap input bisa dilihat pada lampiran E.

Berdasarkan hasil pencarian *belief* yang ada pada tiap input, dapat disimpulkan bahwa pada kluster empat, sembilan input yang mendominasi (diurutkan berdasarkan nilai *belief* yang paling tinggi) pada kluster ini adalah input ke-8 dengan

nilai *belief* sebesar 6,356%, input ke-5 dengan nilai *belief* sebesar 6,356%, input ke-15 dengan nilai *belief* sebesar 6,314%, input ke-13 dengan nilai *belief* sebesar 6,026%, input ke-14 dengan nilai *belief* sebesar 5,861%, input ke-2 dengan nilai *belief* sebesar 5,737%, input ke-18 dengan nilai *belief* sebesar 5,695%, input ke-17 dengan nilai *belief* sebesar 5,613%, dan input ke-16 dengan nilai *belief* sebesar 5,613%. Total dari semua nilai *belief* dari kesembilan input tersebut adalah 53,571%.

Dari analisa terhadap input apa saja yang mendominasi di kluster empat, bisa disimpulkan bahwa kluster empat ini adalah kluster yang terdiri dari dari siswa-siswa yang lebih mengutamakan hal-hal berikut dalam memilih universitas (diurut mulai dari nilai *belief* yang paling besar) :

- a. Ada tidaknya program akselerasi
- b. Kolektivisme universitas
- c. Ada tidaknya edukasi holistik
- d. Kurikulum universitas
- e. Ada tidaknya standar internasional
- f. Keputusan pribadi
- g. Ada tidaknya beasiswa
- h. Reputasi
- i. Kemudahan untuk lulus

D. KESIMPULAN

Dari penelitian ini, didapatkan bahwa Golongan siswa yang mayoritas ada di SMA swasta di Surabaya untuk saat ini adalah siswa golongan kedua (kluster kedua) yang terdiri dari 328 siswa dari 690 siswa (47,54%), kemudian diikuti siswa golongan pertama (kluster pertama) yang terdiri dari 210 siswa dari 690 siswa (30,43%), kemudian diikuti siswa golongan ketiga (kluster ketiga) yang terdiri dari 107 siswa dari 690 siswa (15,51%), dan terakhir diikuti siswa golongan keempat (kluster keempat) yang terdiri dari 45 siswa dari 690 siswa (6,52%).

Sedangkan untuk ke depannya, guna lebih menspesifikkan ciri-ciri calon mahasiswa yang berasal dari SMA swasta yang ada di Surabaya, aplikasi bisa dikembangkan dengan cara memperbanyak jumlah kluster dan memakai kohonen SOM dua dimensi dan aplikasi perlu bisa dibuat lebih fleksibel sehingga tim marketing bisa menambah maupun mengurangi jumlah input serta jumlah kluster.

DAFTAR PUSTAKA

- [1] *Dempster-Shafer theory*. Wikipedia. Available from

http://en.wikipedia.org/wiki/Dempster%E2%80%93Shafer_theory; Internet; accessed January 30, 2014

- [2] Setiawan, Kuswara. *Paradigma Sistem Cerdas (Artificial Intelligence)*. Surabaya: Sekolah Tinggi Teknik Surabaya, 2000.
- [3] J. Klir George & Tina A. Folger. *Fuzzy Sets, Uncertainty, and Information*. United States of America: Prentice –Hall, 1988
- [4] Metode Penelitian. Scribd. Available from <http://www.scribd.com/doc/15852520/28/JENIS-KUESIONER>; Internet; accessed January 30, 2014
- [5] Untung Ariessandi, Muhammad. 2009. "Implementasi Algoritma Dempster Shafer dalam Pembuatan ITS untuk Matakuliah Simulasi dan Permodelan". Universitas Gunadarma
- [6] Caroline, Linda. 2012. "Aplikasi Psikotes Pemilihan Jurusan Kuliah dengan Kohonen *Self Organizing Map (SOM)* Satu Dimensi Berbasis Web". Universitas Pelita Harapan Surabaya
- [7] Dhedha, Cindy. 2012. "Penentuan Konsentrasi Studi Menggunakan Metoda *Self Organizing Maps* Tiga Dimensi.". Universitas Pelita Harapan
- [8] Van Kooij, Rijnardus A. *Menguak Fakta, Menata Karya Nyata*. Jakarta :PT BPK Gunung Mulia, 2008
- [9] A. Churchill, Gilbert. *Dasar-Dasar Riset Pemasara*, Edisi 4, Jilid 1. Jakarta : Erlangga, 2005