

BAB III

METODOLOGI PENELITIAN

Pada bab III akan dibahas tentang metode penelitian dan metode *Self Organizing Map* yang digunakan oleh penulis untuk proses *clustering* (pengelompokan).

3.1 Metodologi Penelitian

Metode penelitian yang digunakan pada skripsi ini merupakan metode penelitian dasar yang mengkaji teori tentang *Self Organizing Map* (SOM) kemudian mengkonstruksi algoritma *Self Organizing Map* ke dalam bahasa pemrograman R dan diaplikasikan pada suatu data. Data yang digunakan merupakan data sekunder yang diperoleh dari Badan Pusat Statistik yang tersedia di *www.bps.co.id*. Data tersebut merupakan data Indeks Pembangunan Manusia di Provinsi Aceh Tahun 2013. Selanjutnya data tersebut akan digunakan untuk proses simulasi pembentukan klaster dalam penelitian ini.

Metode yang digunakan untuk pengelompokan objek adalah metode *Self Organizing Map*. Metode ini digunakan karena menghasilkan akurasi yang lebih baik dalam mengelompokkan objek ke dalam kelompok yang cocok dari pada metode *K-Means* dan metode hirarki serta menunjukkan hasil yang baik saat menggunakan himpunan data yang kecil daripada himpunan data yang besar.

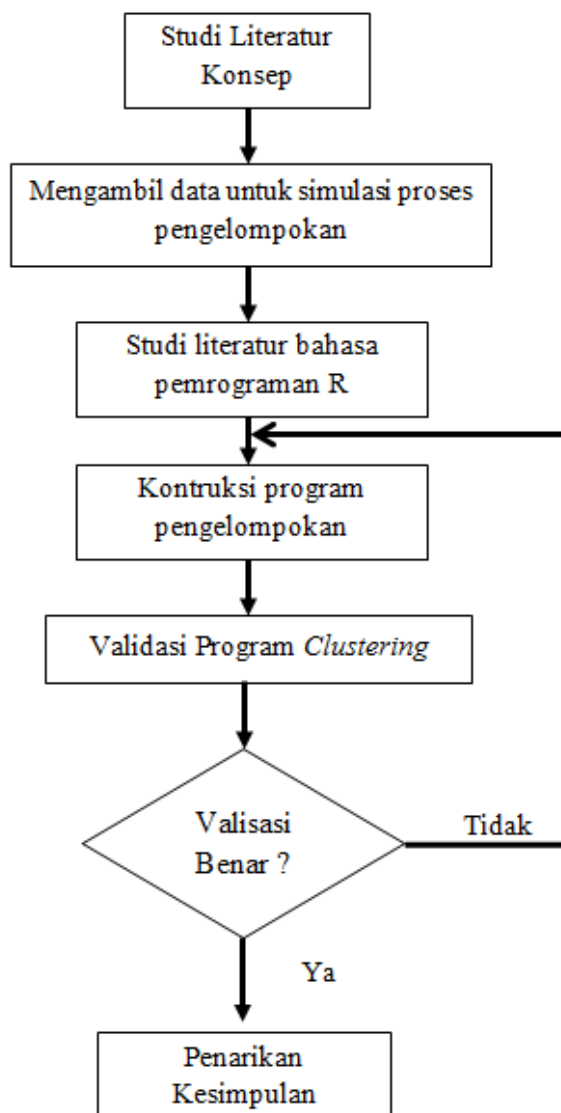
Proses pengelompokan objek dengan menggunakan metode *Self Organizing Map* akan membutuhkan waktu yang cukup lama serta dapat terjadi kesalahan dalam perhitungannya apabila dilakukan secara manual sehingga akan mempengaruhi hasil. Oleh karena itu, akan dibuat program aplikasi untuk proses pengelompokan objek dengan metode *Self Organizing Map* menggunakan bahasa pemrograman R.

Berikut langkah-langkah penelitian pada skripsi ini adalah:

1. Studi literatur mengenai konsep dasar analisis multivariat, analisis klaster, jaringan saraf tiruan, dan metode *Self Organizing Map*.
2. Studi literatur bahasa pemrograman R.
3. Mengambil data dari Badan Pusat Statistik untuk simulasi program yang telah dibuat.

4. Mengkontruksi program aplikasi untuk pengelompokan objek dengan metode *Self Organizing Map* menggunakan bahasa R. Setelah program aplikasi selesai dikonstruksi maka selanjutnya dilakukan proses validasi yaitu dengan cara membandingkan hasil pengelompokan dengan menggunakan program aplikasi dan perhitungan manual. Tujuan validasi ini adalah untuk mengetahui apakah pengelompokan menggunakan program telah sesuai dengan algoritma.
5. Penarikan kesimpulan yaitu pada tahap ini diambil beberapa kesimpulan yang berkaitan dengan tujuan penelitian.

Langkah-langkah dari metodologi penelitian di atas disajikan dalam bentuk diagram alur berikut :

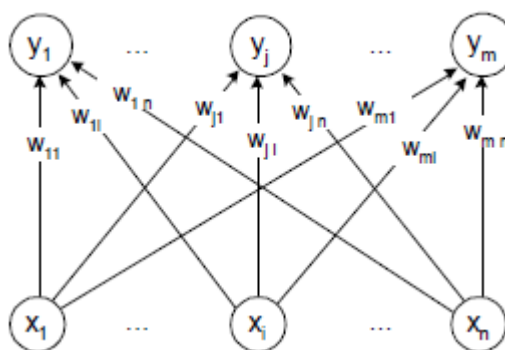


Gambar 3. 1 Diagram Alur Metodologi Penelitian

3.2 Metode Self-Organizing Map

Metode Self-Organizing Map (SOM) merupakan suatu metode Jaringan Syaraf Tiruan yang diperkenalkan oleh Professor Teuvo Kohonen pada tahun 1980-an, sebagai salah satu bentuk topologi dari *Unsupervised Artificial Neural Network* (Unsupervised ANN) dimana dalam proses pelatihannya tidak memerlukan pengawasan atau target keluaran (Anis&Isnanto, 2014, hlm. 48).

Leleury dan Patty (2013, hlm.152) lebih lanjut menjelaskan bahwa pada jaringan SOM terdiri dari dua lapisan (*layer*), yaitu lapisan *input* dan lapisan *output*. Setiap neuron dalam lapisan *input* terhubung dengan setiap neuron pada lapisan *output*. Setiap neuron dalam lapisan *output* merepresentatifkan kelas (klaster) dari *input* yang diberikan. Setiap neuron *output* mempunyai bobot untuk masing-masing neuron *input*. Selama proses *clustering*, klaster (neuron pada lapisan *output*) yang memiliki jarak paling dekat dengan pola input akan terpilih sebagai pemenang dan beserta neuron tetangganya akan memperbaiki bobotnya. Berikut adalah struktur jaringan SOM (Siang, 2004, hlm. 142).



Gambar 3. 2 Struktur Jaringan SOM

Pada algoritma pembelajaran *Self Organizing Map*, akumulasi sinyal yang didapat tidak perlu diaktivasi, karena fungsi aktivasi tidak memberikan pengaruh pada pemilihan *winner neuron* yang akan memperbaiki bobotnya dan bobot tetangganya. (Afif, 2017, hlm.33).

3.2.1 Algoritma Self-Organizing Map

Proses *Self Organizing Map* (SOM) dilakukan dengan mengikuti algoritma berikut (Fausett, hlm. 170, 1994):

a. Masukan :

1. Neuron pada lapisan input $\mathbf{X} = \{\vec{x}_1, \vec{x}_2, \vec{x}_3, \dots, \vec{x}_n\}$; $\vec{x}_i \in R^n$, di mana

$$\vec{x}_i = \langle x_{i1}, x_{i2}, \dots, x_{ip} \rangle$$

Siti Kania, 2018

PROGRAM APLIKASI PENGELOMPOKAN OBJEK DENGAN METODE SELF ORGANIZING MAP
MENGUNAKAN BAHASA R

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

2. Bobot $W = (w_{ij}); i = 1, 2, \dots, m$ dan $j = 1, 2, \dots, p$ di mana p adalah banyaknya variabel dan m adalah banyak kluster diinginkan.
 3. *Learning rate* (α) bernilai $0 < \alpha < 1$
 4. Jumlah iterasi (T).
 5. Galat maksimum (ϵ)
- b. Keluaran : $W = (w_{ij}); i = 1, 2, \dots, m$ dan $j = 1, 2, \dots, p$
Objek yang sudah tercluster
- c. Langkah :
1. Inisialisasi bobot koneksi antara neuron input dan output (w_{ij}) dengan bilangan acak 0 sampai 1, di mana $i = 1, 2, \dots, p$ dan $j = 1, 2, \dots, m$ di mana p adalah banyaknya variabel dan m adalah banyak kluster diinginkan.
 2. Hitung jarak vektor input terhadap bobot koneksi untuk masing-masing neuron output dengan menggunakan rumus :

$$d(j) = \sum_{i=1}^p (\mathbf{w}_{ij} - \mathbf{x}_{ki})^2, k = 1, 2, \dots, n \quad (3.1)$$
 3. Tentukan indeks j sedemikian sehingga d_j minimum.
 4. Perbaharui nilai bobot W_{ij} untuk setiap unit j dengan menggunakan rumus:

$$w_{ij}(\text{new}) = w_{ij}(\text{old}) + h(t)(x_{ni} - w_{ij})(\text{old}) \quad (3.2)$$
 5. Ulangi langkah dua sampai langkah empat untuk setiap vektor input \mathbf{X}
 6. Perbaharui *learning rate* (α) pada saat iterasi ke t ($t=1, 2, 3, \dots, T$) dengan persamaan :

$$\alpha(t+1) = \alpha(1)(1 - \frac{t}{T}) \quad (3.3)$$
 7. Ulangi langkah kelima sampai keenam sampai maksimum iterasi tercapai dan *learning rate* konvergen menuju nol.
 8. Selanjutnya dilakukan proses pengelompokan atau clusterisasi, yaitu dengan cara mencari jarak Euclid antara bobot terakhir yang didapat dengan seluruh objek pengamatan dengan menggunakan persamaan (3.1). Kemudian cari jarak terkecil pada setiap objek pengamatan.

3.2.2 Learning Rate ($\alpha(t)$)

Learning Rate digunakan untuk menunjukkan bagaimana adaptasi pembelajaran terhadap data. Fungsi skalar adaptasi bernilai $0 < \alpha(t) < 1$. Semakin besar nilai $\alpha(t)$, 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* $\alpha(t)$ mendekati 0, perubahan bobot akan semakin kecil dan vektor-vektor input dapat dipetakan dengan baik (Budhi, 2008, hlm.26). Fungsi *Learning Rate* yang digunakan adalah fungsi linear, berikut persamaan dari fungsi linear.

$$\alpha(t + 1) = \alpha(1)\left(1 - \frac{t}{T}\right)$$

di mana $\alpha(1)$ merupakan nilai *learning rate* awal yang ditentukan peneliti.

3.2.3 Neighborhood Function

Fungsi $h(t)$ adalah fungsi lingkungan (*Neighborhood Function*), yang bernilai $0 \leq h(t) \leq 1$. Fungsi ini memberi pengaruh perubahan bobot secara proporsional dari neuron *best matching* ke *neuron - neuron* tetangganya. (Harryanto, 2008, hlm.4). Fungsi lingkungan merupakan fungsi posisi dari dua unit neuron (neuron pemenang dengan neuron lainnya) dengan *Neighborhood Radius* (r) yang diberikan. Terdapat beberapa fungsi lingkungan, dua fungsi lingkungan yang paling umum adalah *Gaussian Neighborhood* dan *bubble neighborhood*. *Gaussian Neighborhood* cenderung lebih dapat diandalkan (inisialisasi yang berbeda cenderung konvergen ke peta yang sama), sedangkan *bubble neighborhood* memiliki kuantisasi kesalahan yang lebih kecil, dan secara komputasi jauh lebih cepat (Lobo, 2009, hlm. 26). Dalam penelitian ini, fungsi lingkungan yang akan digunakan adalah *bubble neighborhood* dan *Neighborhood Radius* sama dengan nol. Berikut fungsi lingkungan *bubble neighborhood*.

$$h(t) = \begin{cases} \alpha(t), & \text{untuk setiap bobot pada neuron pemenang } j \\ 0, & \text{lainnya} \end{cases}$$

3.3 Normalisasi Data

Sebelum dilakukan proses pembelajaran (*training*), data *input* harus dinormalisasi terlebih dahulu. Normalisasi adalah penskalaan terhadap nilai-nilai

input sedemikian sehingga data-data input masuk dalam suatu range tertentu. Pada pembelajaran algoritma *Self Organizing Map* proses normalisasi perlu dilakukan agar rentang nilai pada masing-masing variabel tidak terpaut jauh. Lebih lanjut Larose (2014, hlm. 188) menjelaskan bahwa salah satu kelemahan dari jaringan syaraf tiruan adalah semua nilai variabel dari suatu objek harus dikodekan yaitu mengambil nilai antara nol dan satu, bahkan untuk variabel kategori (Larose, 2014, hlm.188).

Oleh karena hal tersebut, proses normalisasi dilakukan dengan metode *Min-Max Normalization*. Metode *Min-Max Normalization* didefinisikan sebagai berikut :

$$x_i^* = \frac{x_i - \min_{x_i}}{\max_{x_i} - \min_{x_i}}$$

dengan :

- x_i^* : nilai variabel ke-i yang baru setelah dinormalisasi
- x_i : nilai variabel ke-i yang lama sebelum dinormalisasi
- \min_{x_i} : nilai minimum dari variabel ke-i
- \max_{x_i} : nilai maksimum dari variabel ke-i

3.4 Perancangan Program

Pada bagian ini akan dibahas mengenai rancangan data masukan, data keluaran dan algoritma program dari program aplikasi proses pengelompokan menggunakan metode *Self Organizing Map* dengan bantuan bahasa pemrograman R.

3.4.1 Data Masukan

Pada program yang akan dibuat memuat data masukan yang ditunjukkan pada tabel 3.1 berikut ini :

Tabel 3. 1 Daftar Data Masukan

Data	Notasi	Tipe Data
Data yang akan diolah	x	Matriks
Bobot	w	Matriks
<i>Learning Rate</i>	a	Numerik
Jumlah Iterasi	banyakT	Integer
Galat Maksimum	Galat	Numerik

3.4.2 Data Keluaran

Data keluaran pada program yang akan ditampilkan disajikan pada tabel 3.2 berikut ini :

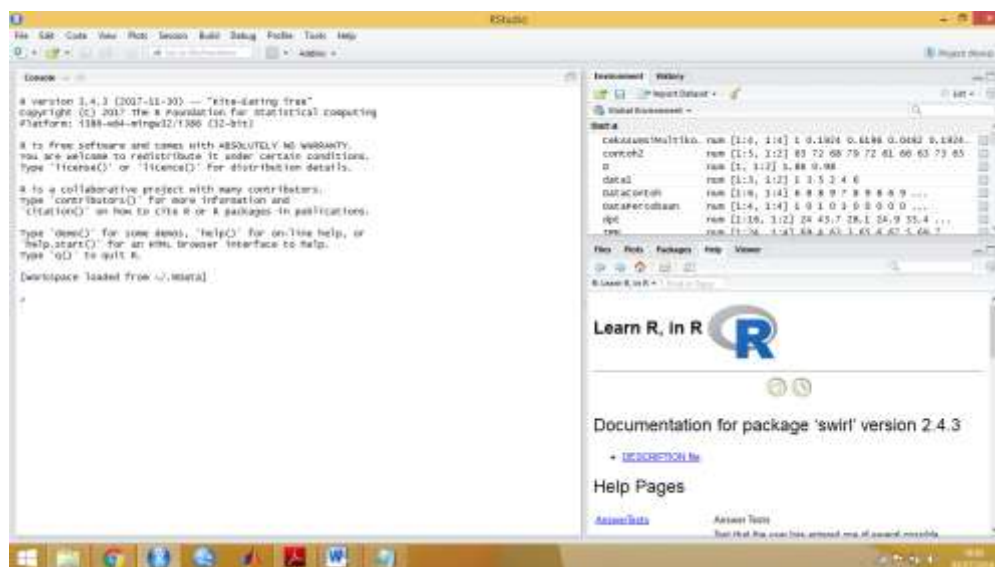
Tabel 3. 2 Daftar Data Keluaran

Data	Notasi	Tipe Data
Bobot Akhir	w	Matriks
Hasil Clustering	hasil	<i>Character</i>

3.4.3 Langkah-Langkah Program dan Pemograman

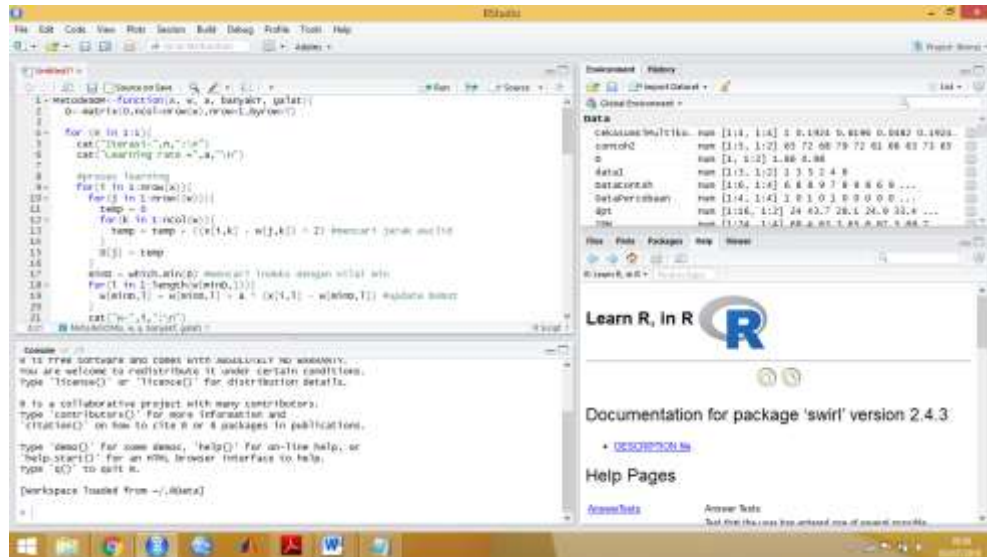
Langkah-langkah yang harus dilakukan untuk mengkontruksi program aplikasi penentuan proses *clustering* dengan menggunakan metode *Self Organizing Map* adalah sebagai berikut:

1. Buka program aplikasi RStudio pada layar *desktop window* sehingga muncul tampilan utama program aplikasi RStudio seperti yang ditunjukkan pada gambar 3.3.



Gambar 3. 3 Tampilan Layar Utama Program Aplikasi RStudio

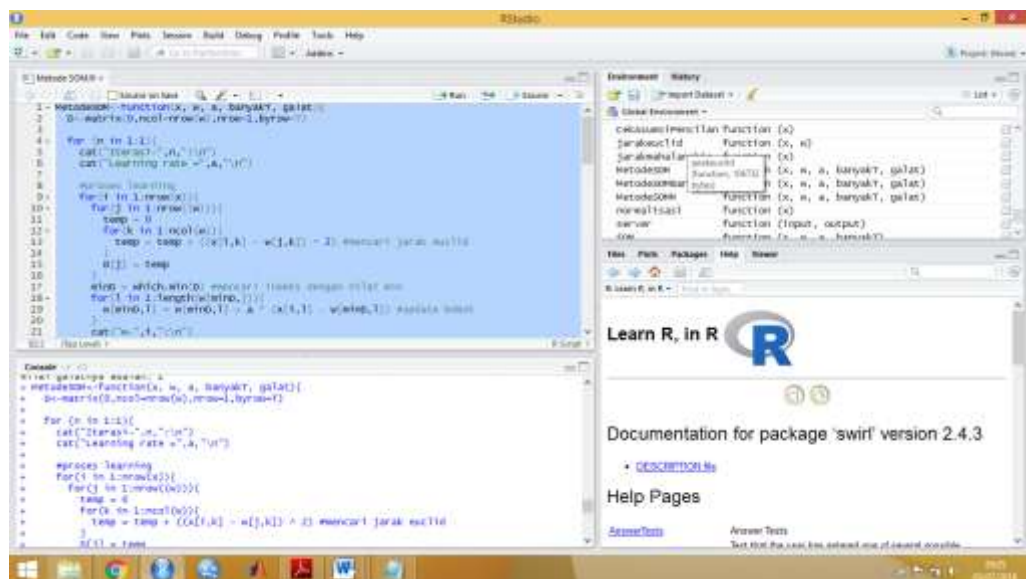
2. Tambahkan '*R Script*' pada Rstudio, kemudian terjemahkan algoritma *Self Organizing Map* ke dalam bahasa pemograman R pada *R Script* tersebut seperti pada gambar 3.4 berikut.



Gambar 3. 4 Mengkontruksi Algoritma *Self Organizing Map* ke dalam Bahasa R

Teks *coding* hasil penerjemahan algoritma *Self Organizing Map* ke dalam bahasa pemrograman R di lampirkan pada lembar lampiran 1.

3. Setelah algoritma *Self Organizing Map* diterjemahkan ke dalam bahasa R, kemudian *save*.
4. Untuk menjalankan program tersebut blok atau tekan Ctrl+A pada semua teks *coding* pada lembar kerja '*R Script*' tersebut kemudian tekan *Run* seperti pada gambar 3.5 berikut.



Gambar 3. 5 Menjalankan Program yang telah Dikontruksi

Siti Kania, 2018

PROGRAM APLIKASI PENGELOMPOKAN OBJEK DENGAN METODE SELF ORGANIZING MAP MENGGUNAKAN BAHASA R

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

Dapat dilihat pada gambar 3.5 tersebut, program tersebut telah dijalankan pada lembar kerja 'Console'.

5. Selanjutnya untuk diaplikasikan ke dalam data, ketik nama program yang telah kita buat pada lembar kerja 'Console' kemudian masukkan data seperti tabel 3.1.