

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

## BAB II

### LANDASAN TEORI

#### 2.1 Jaringan Syaraf Tiruan

Jaringan syaraf Tiruan (JST) merupakan representasi buatan dari otak manusia yang selalu mencoba mensimulasikan proses pembelajaran pada otak manusia tersebut, istilah buatan disini digunakan karena jaringan syaraf ini diimplementasikan dengan menggunakan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran (Kusumadewi, 2003 dalam Ranadhi dkk, 2006).

Jaringan syaraf tiruan adalah paradigma pengolahan informasi yang terinspirasi oleh system saraf secara biologis, seperti proses informasi pada otak manusia. Elemen kunci dari paradig ini adalah struktur dari system pengolahan informasi yang terdiri dari sejumlah besar elemen pemrosesan yang saling berhubungan (neuron), bekerja serentak untuk menyelesaikan masalah tertentu. Cara kerja JST seperti cara kerja manusia, yaitu belajar melalui contoh (T.sutojo dkk, 2011)

Jaringan saraf tiruan (JST atau *Artificial Neural Network*), adalah sistem komputasi di mana arsitektur dan operasi diilhami dari pengetahuan tentang sel saraf biologis di dalam otak, yang merupakan salah satu representasi buatan dari otak manusia yang selalu menstimulasi proses pembelajaran pada otak manusia tersebut (Hermawan, dalam Agustin, 2012).

Jaringan syaraf tiruan tidak diprogram untuk menghasilkan keluaran tertentu. Kata lain, penyelesaian permasalahan dengan jaringan syaraf tiruan tidak memerlukan pemrograman. JST terdiri dari sejumlah simpul (*node*) yang merupakan elemen pemroses. Setiap simpul tersebut memodelkan sebuah sel syaraf biologis (*neuron*). Hubungan antar simpul dicapai melalui bobot koneksi (*weight*). Bobot koneksi menentukan apakah sinyal yang mengalir bersifat peredam (*inhibitory connection*). Bobot koneksi yang bersifat meredam dapat dinyatakan, misalnya oleh bilangan negative, sedangkan yang bersifat merangsang oleh



digunakan untuk mengontrol jaringan syaraf tiruan agar menghasilkan efisiensi yang maksimal.

### 2.1.3 Karakteristik Jaringan Syaraf Tiruan

Dengan meniru sistem jaringan biologis (manusia), maka sistem jaringan syaraf tiruan memiliki 3 karakteristik utama (Antoni Siahaan, 2011), yaitu:

#### 1. Arsitektur Jaringan

Merupakan pola keterhubungan antara neuron Keterhubungan neuron-neuron inilah yang membentuk suatu jaringan.

#### 2. Algoritma Jaringan

Merupakan metode untuk menentukan nilai bobot hubungan. Ada dua metode pada algoritma jaringan saraf tiruan, yaitu metode bagaimana JST tersebut melakukan Pelatihan (Pembelajaran) dan metode bagaimana JST tersebut melakukan Pengenalan (Aplikasi).

#### 3. Fungsi Aktivasi

Merupakan fungsi untuk menentukan nilai keluaran berdasarkan nilai total masukan pada neuron. Fungsi aktivasi suatu algoritma jaringan dapat berbeda dengan fungsi aktivasi algoritma jaringan lain.

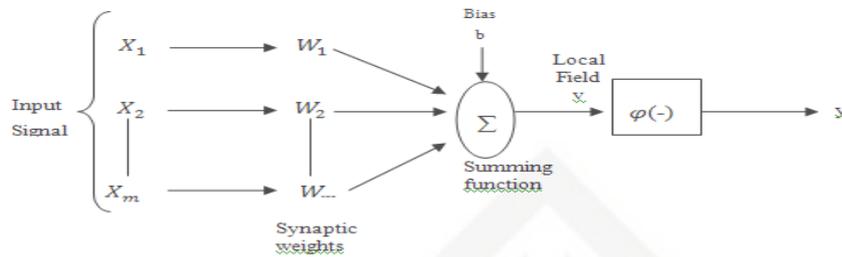
### 2.1.4 Pemodelan dan Konsep Dasar Jaringan Syaraf Tiruan

Jaringan syaraf terdiri atas beberapa *neuron* dan ada hubungan antara neuron–neuron tersebut. *Neuron* adalah sebuah unit pemroses informasi yang menjadi dasar pengoperasian jaringan syaraf tiruan. Syaraf adalah sebuah unit pemroses informasi dengan tiga elemen dasar (Rikki, 2012 dalam Elvia Budianita, 2013) yaitu :

1. Satu set link yang terhubung.
2. Sebuah penjumlah untuk menghitung besarnya penambahan pada sinyal masukan.
3. Sebuah fungsi aktivasi untuk membatasi banyaknya keluaran pada syaraf.

Sebagian besar jaringan syaraf melakukan penyesuaian bobot-bobotnya selama menjalani pelatihan. Pelatihan dapat berupa pelatihan terbimbing

(*supervised training*) di mana diperlukan pasangan masukan-sasaran untuk tiap pola yang dilatihkan. Jenis kedua adalah pelatihan tak terbimbing (*unsupervised training*).



**Gambar 2. 1 Model Jaringan Syaraf Tiruan (Rikki, 2012 dalam Elvia Budianita, 2013)**

Setiap pola-pola informasi input dan output yang diberikan kedalam JST diproses dalam neuron. Neuron-neuron tersebut terkumpul di dalam lapisan-lapisan yang disebut neuron *layers* . Lapisan-lapisan penyusun JST tersebut dapat dibagi menjadi 3 (Puspitaningrum, 2006), yaitu :

1. Lapisan *Input*, unit-unit di dalam lapisan input disebut unit-unit input. Unit-unit input tersebut menerima pola inputan data dari luar yang menggambarkan suatu permasalahan.
2. Lapisan Tersembunyi, unit-unit di dalam lapisan tersembunyi disebut unit-unit tersembunyi. Dimana outputnya tidak dapat secara langsung diamati.
3. Lapisan *Output*, unit-unit di dalam lapisan *output* disebut unit-unit *output*. *Output* dari lapisan ini merupakan solusi JST terhadap suatu permasalahan.

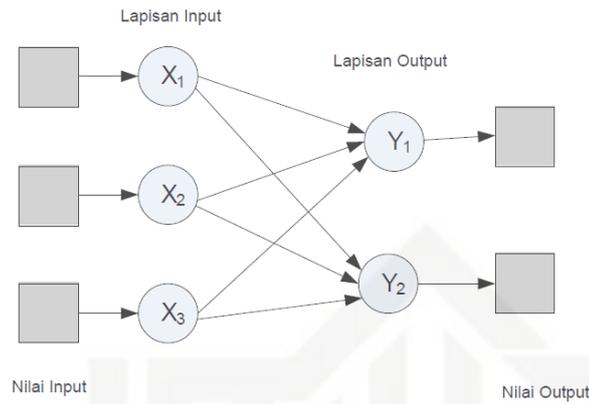
### 2.1.5 Arsitektur Jaringan

JST memiliki beberapa arsitektur jaringan yang sering digunakan dalam berbagai aplikasi. Arsitektur JST tersebut, antara lain (Kusumadewi, 2003 dalam Agustin, 2012).

#### 1. Jaringan Layar Tunggal (*Single Layer Network*)

Jaringan dengan lapisan tunggal terdiri dari 1 layer input dan 1 layer output. Setiap neuron/unit yang terdapat di dalam lapisan/layer input selalu terhubung dengan setiap neuron yang terdapat pada layer output. Jaringan ini hanya menerima input kemudian secara langsung akan mengolahnya menjadi output tanpa harus

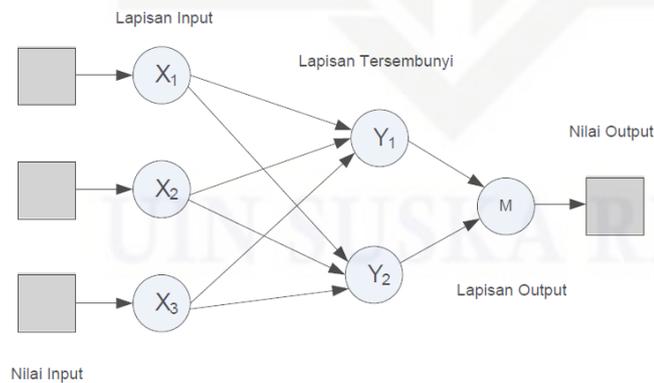
melalui lapisan tersembunyi. Contoh algoritma JST yang menggunakan metode ini yaitu: ADALINE, Hopfield, Perceptron.



**Gambar 2. 2** Arsitektur Layer Tunggal (Hermawan, 2006 dalam Agustin, 2012)

## 2. Jaringan Banyak Lapisan (*Multilayer Net*)

Jaringan dengan lapisan jamak memiliki ciri khas tertentu yaitu memiliki 3 jenis *layer* yakni *layer input*, *layer output*, dan juga *layer tersembunyi*. Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang lebih kompleks dibandingkan jaringan dengan lapisan tunggal. Namun, proses pelatihan sering membutuhkan waktu yang cenderung lama. Contoh algoritma Jaringan Syaraf Tiruan yang menggunakan metode ini yaitu: MADALINE, *backpropagation*, *Neocognitron*.



**Gambar 2. 3** Arsitektur Multilayer (Hermawan, 2006 dalam Agustin, 2012)

### 2.1.6 Algoritma Jaringan Syaraf Tiruan

Pembelajaran/pelatihan jaringan syaraf tiruan (Puspaningrum, 2006 dalam Agustin, 2012).

Dimasukkan n contoh pelatihan kedalam jaringan syaraf tiruan, lakukan:

1. Inisialisasi bobot-bobot jaringan. Set  $I = 1$ .
2. Masukkan contoh ke I (dari sekumpulan contoh pembelajaran yang terdapat dalam set pelatihan) kedalam jaringan pada lapisan *input*.
3. Cari tingkat aktivasi unit-unit input menggunakan algoritma aplikasi
4. Update bobot-bobot dengan menggunakan aturan pembelajaran jaringan.

If → kinerja jaringan memenuhi standar yang ditentukan sebelumnya (memenuhi syarat untuk berhenti).

Then → exit.

5. If  $i = n$  then reset  $i = 1$

Else  $i = i - 1$

Ke langkah 2.

Algoritma aplikasi/inferensi jaringan saraf tiruan (Puspitaningrum, 2006 dalam Agustin, 2012):

Dimasukkan sebuah contoh pelatihan kedalam jaringan saraf tiruan, lakukan:

1. Masukkan kasus kedalam jaringan pada lapisan *input*.
2. Hitung tingkat aktifasi *node-node* jaringan.
3. Untuk jaringan koneksi umpan maju, jika tingkat aktivasi dari semua unit *outputnya* telah dikalkulasi, maka *exit*. Untuk jaringan dengan kondisi balik, Jika tingkat aktivasi dari semua unit *outputnya* menjadi konstan atau mendekati konstan, maka *exit*. Jika tidak, kembali ke langkah 2. Jika jaringannya tidak stabil, maka *exit* dan *fail*.

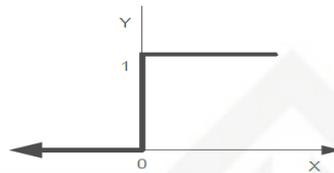
### 2.1.7 Fungsi Aktivasi Jaringan Syaraf Tiruan

Dalam jaringan syaraf tiruan, fungsi aktivasi digunakan untuk menentukan keluaran suatu Neuron. Argument fungsi aktivasi adalah net masukan (kombinasi linier masukan dan bobotnya).

Beberapa fungsi aktivasi yang digunakan adalah:

- a. Fungsi Threshold (batas ambang).

Fungsi Threshold merupakan fungsi threshold biner. Untuk kasus bilangan bipolar, maka angka 0 diganti dengan angka -1. Adakalanya dalam jaringan syaraf tiruan ditambahkan suatu unit masukan yang nilainya selalu 1. Unit tersebut dikenal dengan bias. Bias dapat dipandang sebagai sebuah input yang nilainya selalu 1. Bias berfungsi untuk mengubah threshold menjadi = 0.



**Gambar 2. 4 Fungsi aktivasi Threshold (Kusumadewi, 2004 dalam Agustin, 2012)**

$$F(x) = \begin{cases} 1 \\ 0 \end{cases}$$

Jika  $x \geq a$

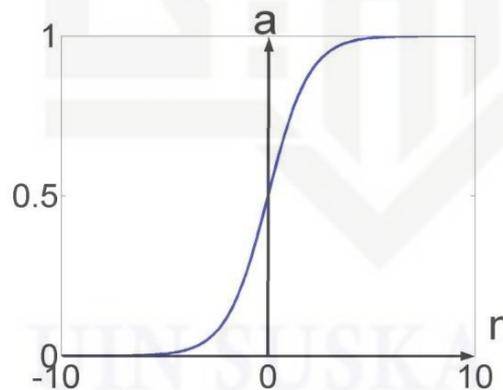
Jika  $x < a$

a. *Fungsi Sigmoid.*

Fungsi ini sering digunakan karena nilai fungsinya yang sangat mudah untuk

di diferensiasikan.

$$F(x) = \frac{1}{1 + e^{-x}}$$

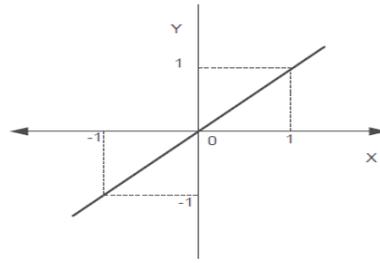


**Gambar 2. 5 Fungsi aktivasi Sigmoid**

b. *Fungsi Identitas.*

Digunakan jika keluaran yang dihasilkan oleh jaringan syaraf tiruan merupakan sembarang bilangan riil (bukan hanya pada range [0,1] atau [1,-1]).

$$Y = X$$



**Gambar 2. 6 Fungsi aktivasi Identitas (Kusumadewi, 2004 dalam Agustin Maria, 2012)**

### Proses Pembelajaran

Fusett (1994) dalam kutipan Puspitaningrum (2006) menjelaskan terdapat dua tipe pembelajaran dalam Jaringan Syaraf Tiruan, yaitu:

a. Pembelajaran terawasi (*supervised learning*)

Metode pembelajaran pada jaringan syaraf disebut terawasi jika *output* yang diharapkan telah diketahui sebelumnya. Maksudnya, sejak awal pembelajaran, output telah ditetapkan terlebih dahulu, dengan nama lainnya adalah target. Perbedaan antara output-actual dengan output-target yang diinginkan digunakan untuk mengoreksi bobot jaringan syaraf tiruan agar jaringan syaraf tiruan dapat menghasilkan jawaban sedekat (semirip) mungkin dengan jawaban yang benar yang telah diketahui oleh jaringan syaraf. Terdapat berbagai tipe pembelajaran terawasi beberapa diantaranya *Hebb Rule*, *Perceptron*, *Delta Rule*, *Backpropagation*, *Heteroassociative Memory*, *Bidirectional Associative Memory (BAM)*, *Learning Vektor Quantization (LVQ)*.

b. Pembelajaran tak terawasi (*unsupervised learning*)

Pembelajaran tak terawasi tidak memerlukan target *output*. Hasil yang seperti apakah yang diharapkan selama proses pembelajaran, pada metode ini tidak dapat ditentukan. Selama proses pembelajaran, nilai bobot disusun dalam suatu *range* tertentu tergantung pada nilai *input* yang diberikan. Tujuan pembelajaran ini adalah mengelompokkan unit-unit yang hampir sama dengan suatu area tertentu. Pembelajaran ini biasanya sangat cocok untuk pengelompokan (klasifikasi) pola. Contoh metode pembelajaran tak terawasi adalah jaringan kohonen (*kohonen network*).

## 2.2 Learning Vector Quantization (LVQ)

*Learning Vector Quantization* (LVQ) adalah suatu metode untuk melakukan pembelajaran pada lapisan kompetitif yang terawasi. Suatu lapisan kompetitif akan secara otomatis belajar untuk mengklasifikasikan vektor-vektor input. Kelas-kelas yang didapatkan sebagai hasil dari lapisan kompetitif ini hanya tergantung pada jarak antara vektor-vektor input. Jika dua vektor input mendekati sama, maka lapisan kompetitif akan meletakkan kedua vektor input tersebut kedalam kelas yang sama (Kusumadewi, 2003)

Menurut Wahab (2008), di kutip oleh (Widodo, dkk 2013), LVQ terdiri dari dua lapis yaitu lapis kompetitif dan lapis linier. Lapis pertama yaitu lapis kompetitif mengklasifikasikan masukan dengan cara kompetitif. Lapis kedua yaitu lapis linear mentransformasikan sub-sub kelas keluaran lapis pertama menjadi kelas-kelas yang telah didefinisikan sebelumnya (target). Setiap sub kelas diwakili oleh suatu neuron pada keluaran lapis kompetitif dan setiap kelas diwakili oleh satu neuron pada keluaran lapis linier. Neuron pada keluaran lapis kompetitif biasanya disebut sebagai *hidden* neuron dan neuron pada keluaran lapis linier disebut sebagai output neuron. Sub kelas pada lapis kompetitif adalah hasil kompetisi pada lapis tersebut sedangkan kelas pada lapis linier adalah kelas yang didefinisikan oleh pengguna (target). Pemrosesan yang terjadi pada setiap neuron adalah mencari jarak terdekat antara suatu vektor masukan ke bobot yang bersangkutan.

Jaringan LVQ mempunyai target yang akan dicapai. Lapisan kompetitif belajar mengenali dan mengklasifikasikan vektor-vektor masukan. Jika ada 2 vektor yang hampir sama, maka lapisan kompetitif akan menempatkan keduanya pada kelas yang sama. Dengan demikian LVQ belajar mengklasifikasikan vektor masukan ke kelas target yang ditentukan oleh pengguna. Gambar 2.2 menunjukkan Arsitektur dari jaringan syaraf LVQ.

Hak Cipta Dilindungi Undang-Undang

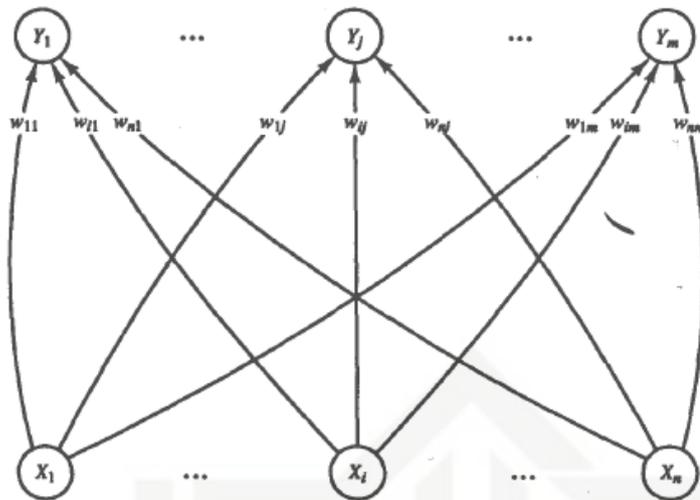
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

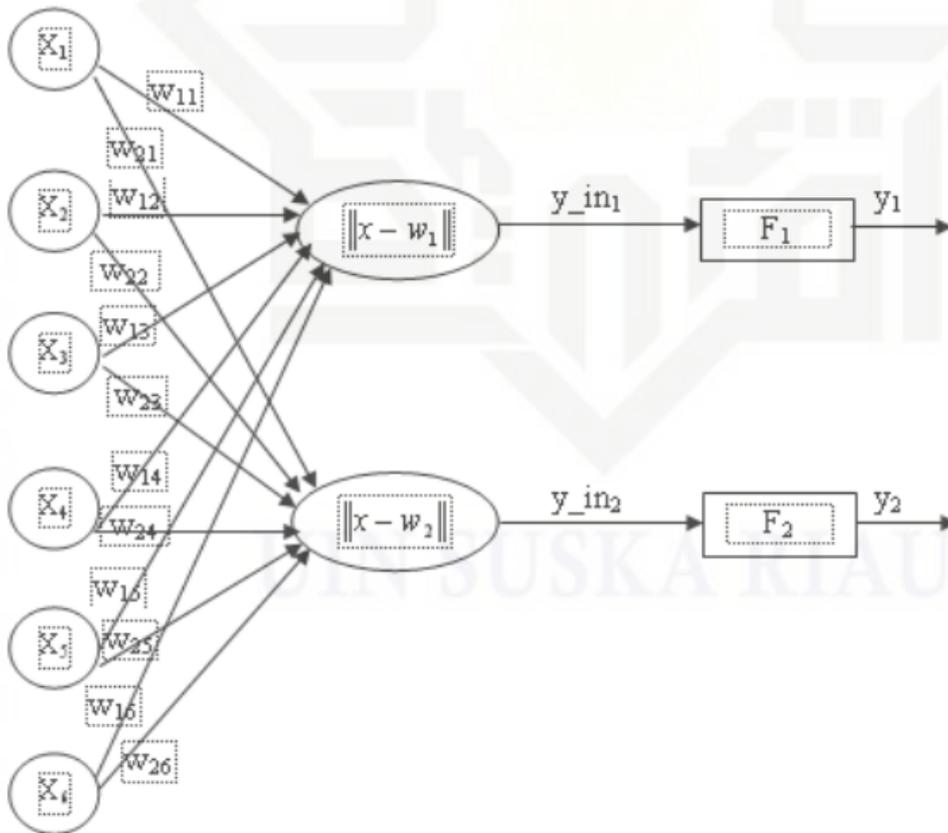
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Hak Cipta Dilindungi Undang-Undang  
Hak Kekayaan Intelektual UIN Suska Riau  
Fakultas Syariah dan Hukum UIN Suska Riau  
Jurnal Ilmiah: *Al-Furqan* Vol. 1 No. 1 Desember 2013  
Pusat Studi Islam UIN Suska Riau  
Universitas Islam Sultan Syarif Kasim Riau



**Gambar 2. 7** Arsitektur Jaringan *Learning Vector Quantization*

Selain itu, Kusumadewi dan Hartati (2006) juga menunjukkan jaringan LVQ dengan 6 unit pada lapisan input dan 2 unit (neuron) pada lapisan output. Arsitektur LVQ tersebut dapat disajikan pada Gambar 2.4.



**Gambar 2. 8** Arsitektur jaringan LVQ dengan 6 unit input dan 2 unit output

### 2.2.1 Algoritma LVQ 1

Pemrosesan yang terjadi pada setiap neuron adalah mencari jarak antara suatu vektor input ke bobot yang bersangkutan ( $w_1$  dan  $w_2$ ).  $W_1$  adalah vektor bobot yang menghubungkan setiap neuron pada lapisan input ke neuron pertama pada lapisan output, sedangkan  $w_2$  adalah vektor bobot yang menghubungkan setiap neuron pada lapisan input ke neuron kedua lapisan output. Fungsi aktivasi  $F_1$  akan memetakan  $y_{in1}$  ke  $y_1 = 1$  apabila  $\|x-w_1\| < \|x-w_2\|$ , dan  $y_1 = 0$  jika sebaliknya. Demikian pula dengan yang terjadi pada fungsi aktivasi  $F_2$ , akan memetakan  $y_{in2}$  ke  $y_2 = 1$  apabila  $\|x-w_2\| < \|x-w_1\|$ , dan  $y_2 = 0$  jika sebaliknya  $|x-w_2| > |x-w_1|$ .

Algoritma LVQ (Kususmadewi, 2002 dikutip oleh Amanu Ramadona, dkk 2004):

1. Tetapkan: Bobot ( $W$ ), Maksimum Epoch (MaxEpoch) *error* minimum yang diharapkan (Eps), *learning rate* ( $\alpha$ )
2. Masukan :
  - a. Data *input* :  $x(m,n)$
  - b. Target berupa kelas :  $T(1,n)$
3. Terapkan kondisi awal:
  - a. Epoch = 0;
  - b. Err = 1.
4. Kerjakan selama : (epoch < MaxEpoch) dan ( $\alpha > Eps$ )
  - a. Epoch = epoch+1;
  - b. Kerjakan untuk  $i= 1$  sampai  $n$ 
    - i. Memilih ( $J$ ) jarak sedemikian hingga  $\|X-W_j\|$  minimum (sebut sebagai  $C_j$ )
    - ii. Perbaiki  $W_j$  dengan ketentuan:
      - Jika  $T = C_j$  maka :
        - $W_j(\text{baru}) = W_j(\text{lama}) + \alpha (x-W_j(\text{lama})) \dots\dots\dots(2.1)$
        - Jika  $C_j \neq T$  maka :  $W_j(\text{baru}) = W_j(\text{lama}) - \alpha (X-W_j(\text{lama})) \dots\dots(2.2)$
  - c. Kurangi nilai  $\alpha$ 
    - $\alpha \text{ baru} = \alpha - (0.1 \times \alpha) \dots\dots\dots(2.3)$

Setelah dilakukan pelatihan, akan diperoleh bobot-bobot akhir ( $W$ ). Bobot-bobot ini nantinya akan digunakan untuk melakukan simulasi atau pengujian. Misalkan dilakukan pengujian terhadap  $np$  buah data. Maka algoritma pengujiannya adalah:

1. Masukkan data yang akan diuji, misal  $X_{ij}$  dengan  $i = 1, 2, \dots, np$  dan  $j = 1, 2, \dots, m$ .
2. Kerjakan untuk  $i=1$  hingga  $np$ 
  - a. Tentukan  $J$  sedemikian hingga  $\|X_{ij}-W_{ij}\|$  minimum
  - b.  $J$  adalah kelas untuk  $X_i$

## 2.2.2 Pengembangan dan Variasi Algoritma LVQ 2

Menurut Kohonen (1990a, 1990b) dikutip oleh Fausett (1994) terdapat beberapa variasi dari algoritma LVQ dasar (LVQ1) yaitu, LVQ2, LVQ2.1, dan LVQ3. Karakteristik pada algoritma LVQ1 adalah hanya vektor referensi terdekat (vektor pemenang) dengan vektor masukan yang diperbaharui. Arah perpindahan vektor tergantung pada apakah vektor referensi memiliki kelas yang sama dengan vektor masukan. Algoritma LVQ yang telah ditingkatkan, vektor pemenang dan vektor *runner up* akan sama-sama belajar bila kondisi tertentu terpenuhi. Idanya adalah bila jarak antara vektor masukan dengan vektor pemenang dan vektor *runner up* kira-kira mempunyai jarak yang sama.

### 2.2.2.1 Learning Vector Quantization 2 (LVQ 2)

Modifikasi pertama adalah LVQ2. LVQ2 adalah sebuah algoritma hasil pengembangan dari algoritma LVQ awal. Kondisi dimana kedua vektor akan diperbaharui jika :

1. Unit pemenang dan *runner up* (vektor terdekat kedua) merepresentasikan kelas yang berbeda
2. Vektor masukan mempunyai kelas yang sama dengan *runner up*
3. Jarak antara vektor masukan ke pemenang dan jarak antara vektor masukan ke *runner up* kira-kira sama.

Kondisi ini diperlihatkan di dalam notasi berikut:

$X$  vektor masukan saat ini

$Y_c$  vektor referensi terdekat dengan  $X$

$Y_r$  vektor referensi terdekat berikutnya dengan  $X$  (*runner up*)

$D_c$  jarak dari X ke  $Y_c$   
 $D_r$  jarak dari X ke  $Y_r$

Vektor referensi dapat diperbaharui jika masuk ke dalam daerah yang disebut *window* ( $\epsilon$ ). *Window* yang digunakan untuk memperbaharui vektor referensi didefinisikan sebagai berikut:

Vektor masukan X akan masuk ke dalam window bila

$$\frac{d_c}{d_r} > 1 - \epsilon, \quad \frac{d_r}{d_c} < 1 + \epsilon, \quad \dots \dots \dots (2.4)$$

dengan nilai  $\epsilon$  tergantung dari jumlah data pelatihan. Berdasarkan Kohonen (1990a) dalam Fausett (1994) nilai  $\epsilon = 0.3$  adalah nilai yang disarankan. Vektor  $Y_c$  dan  $Y_r$  akan diperbaharui bila kondisi 1,2 dan 3 terpenuhi. Vektor  $Y_c$  dan  $Y_r$  diperbaharui dengan menggunakan persamaan :

$$Y_c(t+1) = Y_c(t) - \alpha(t)[X(t) - Y_c(t)] \dots \dots \dots (2.5)$$

$$Y_r(t+1) = Y_r(t) + \alpha(t)[X(t) - Y_r(t)] \dots \dots \dots (2.6)$$

Berikut adalah contoh perhitungan menggunakan jaringan syaraf tiruan LVQ2 untuk mencari vektor bobot masing-masing kelas dengan asumsi terdapat dua kelas yg berbeda (kelas 1 dan 2) dengan data pelatihan ditunjukkan dengan Tabel 2.1 (Nugroho, 2011).

**Tabel 2. 1 Contoh data pelatihan berupa vektor dengan ukuran 1x4**

(x1, x2, x3, x4)	T
1 1 0 0	1
0 0 0 1	2
0 0 1 1	2
1 0 0 0	1
0 1 1 0	2

## Pelatihan

Ambil dua vektor masukan pertama sebagai inisialisasi bobot awal masing-masing kelas

Bobot awal kelas-1  $w_1 = (1, 1, 0, 0)$

Bobot awal kelas-2  $w_2 = (0, 0, 0, 1)$

Learning rate  $\alpha = 0.1$  dengan nilai  $\alpha$  akan dikurangi sebesar  $0.1 \alpha$  pada akhir epoch saat ini

Window  $\epsilon = 0.3$

### Epoch-1

- a. Vektor masukan 0 0 1 1

$$D_1 = \sqrt{((0-1)^2 + (0-1)^2 + (1-0)^2 + (1-0)^2)} = 2$$

$$D_2 = \sqrt{((0-0)^2 + (0-0)^2 + (1-0)^2 + (1-1)^2)} = 1$$

Jarak terdekat adalah  $D_2$  sehingga  $J = 2$  yang menunjukkan vektor masukan masuk kategori 2 ( $C_2$ ) sehingga  $C_2 = T$  dengan  $T = 2$ , sehingga bobot  $w_2$  diperbaiki dengan menggunakan persamaan 2.1.

$$W_2 \text{ (baru)} = (0, 0, 1, 1) + 0.1 * \{(0, 0, 1, 1) - (0, 0, 0, 1)\} = (0, 0, 0, 1.1)$$

- b. Vektor masukan 1 0 0 0

Jarak terdekat adalah  $D_1$  sehingga  $J = 1$ , sehingga  $C_1 = T$  dengan  $T = 1$ , sehingga bobot  $w_1$  (baru) = (1, 0.9, 0, 0)

- c. Vektor masukan 0 1 1 0

$$D_1 = 1.418$$

$$D_2 = 1.792$$

Jarak terdekat adalah  $D_1$  sehingga  $J = 1$ , tapi  $C_1 \neq T$  dengan  $T = 2$ , sehingga perlu diperiksa apakah jarak runner up ( $D_2$ ) masih masuk ke dalam window  $\epsilon = 0.3$  dengan menggunakan persamaan (3.6). Pernyataan kondisional  $((D_1) > (1-\epsilon)*D_2)$  AND  $((D_2) < ((1-\epsilon)*D_1))$  akan menghitung  $((1.418) > (0.7 * 1.792))$  AND  $(1.792 < (1.3*1.418))$  sehingga menghasilkan (True

AND True) yang hasil akhirnya adalah True sehingga bobot  $w_1$  dan  $w_2$  diperbaharui menggunakan persamaan (3.5) dan (3.6)

$$W_1 \text{ (baru)} = (1, 0.9, 0, 0) - 0.1 * \{(0, 1, 1, 0) - (1, 0.9, 0, 0)\}$$

$$= (1.1, 0.89, -0.1, 0)$$

$$W_2 \text{ (baru)} = (0, 0, 0, 1.1) + 0.1 * \{(0, 1, 1, 0) - (0, 0, 0, 1.1)\}$$

$$= (0, 0.1, 0.1, 0.99)$$

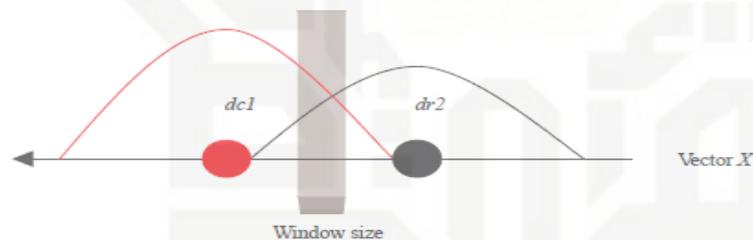
d. Kurangi nilai  $\alpha$

$$\alpha \text{ (baru)} = \alpha \text{ (lama)} - 0.1 \alpha \text{ (lama)} = 0.1 - 0.1 * 0.1 = 0.09$$

e. Tes kondisi berhenti

yaitu bila jumlah epoch sudah melebihi jumlah maksimum epoch yang diperbolehkan, atau bila nilai  $\alpha$  lebih kecil dari nilai maksimum error.

Jendela penyangga (*window*) dapat dilihat pada Gambar 2.6.



Gambar 2. 9 Jendela penyangga antara 2 kelas

### 2.3 Normalisasi

Pada perhitungan jarak *euclidean*, atribut berskala panjang dapat mempunyai pengaruh lebih besar daripada atribut berskala pendek. Oleh karena itu, untuk mencegah hal tersebut perlu dilakukan normalisasi terhadap nilai atribut. Normalisasi adalah proses transformasi nilai menjadi kisaran 0 dan 1 (Teknomo, 2006 dalam Elvia Budianita, 2013). Normalisasi ini merupakan proses penskalaan nilai atribut dari data sehingga bisa jatuh pada range tertentu. Normalisasi ini terbagi menjadi 5 jenis metode pencariannya, yakni : *min-max*, *z-score*, *decimal scaling*, *sigmoidal*, dan *softmax*. *Z-score* merupakan metode normalisasi yang

berdasarkan mean (nilai rata-rata) dan *standard deviation* (deviasi standar) dari data. Dibawah ini merupakan rumus dari *z-score* :

$$\text{newdata} = (\text{data} - \text{mean}) / \text{std}$$

Metode ini sangat berguna jika tidak mengetahui nilai actual minimum dan maksimum dari data.

Sedangkan metode normalisasi *Decimal scaling* ini, melakukan normalisasi dengan menggerakkan nilai decimal dari data ke arah yang diinginkan. Berikut rumus dari *decimal scaling* :

$$\text{newdata} = \text{data} / 10^i$$

Dimana *i* adalah nilai integer untuk menggerakkan nilai decimal ke arah yang diinginkan.

Metode selanjutnya adalah *Sigmoidal Normalization* yakni, melakukan normalisasi data secara nonlinier ke dalam range -1-1 dengan menggunakan fungsi sigmoid. Berikut rumus dari metode ini adalah :

$$\text{newdata} = (1 - e^{-x}) / (1 + e^{-x})$$

dimana:

$$x = (\text{data} - \text{mean}) / \text{std}$$

$$e = \text{nilai eksponensial (2.718281828)}$$

Metode ini sangat berguna pada saat data-data yang ada melibatkan data-data outlier.

Berikutnya metode *Softmax*, metode ini merupakan pengembangan transformasi secara linier. Output rangenya adalah 0-1. Berikut adalah rumus dari metode ini :

$$\text{newdata} = 1 / (1 + e^{-(\text{transfdata})})$$

dimana:

$$\text{transfdata} = (\text{data} - \text{mean}) / (x * (\text{std} / (2 * 3.14)))$$

$$x = \text{respon linier di deviasi standar}$$

**Hak Cipta Dilindungi Undang-Undang**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

metode yang terakhir adalah Min-Max, merupakan metode normalisasi dengan melakukan transformasi linier terhadap data asli. Formula untuk normalisasi atribut X adalah:

$$X^* = \frac{X - \min(X)}{\max(X) - \min(X)} \dots\dots\dots(2.7)$$

dengan,  
 X\* adalah nilai setelah dinormalisasi,  
 X adalah nilai sebelum dinormalisasi,  
 min(X) adalah nilai minimum dari fitur, dan  
 max(X) adalah nilai maksimum dari suatu fitur

Keuntungan dari metode ini adalah keseimbangan nilai perbandingan antar data saat sebelum dan sesudah proses normalisasi. Tidak ada data bias yang dihasilkan oleh metode ini. Kekurangannya adalah jika ada data baru, metode ini akan memungkinkan terjebak “out of bound” error. Namun dari ke-5 metode tersebut, maka digunakan metode normalisasi Min-Max, karena selain dalam data diketahui nilai minimum dan maksimum nya, min-max ini sudah banyak digunakan para peneliti lainnya untuk melakukan normalisasi, karena min-max tergolong mudah dan hasil yang dapat adalah tidak bias sehingga mempermudah pengerjaan penormalisasian data dan lebih efisien.

Teknomo (2006) dalam Elvia Budianita (2013) juga menjelaskan tentang penilaian dan rank adalah variabel ordinal yang dapat diubah menjadi variabel kuantitatif melalui normalisasi. Suharto (2008) dalam Elvia Budianita (2013) menjelaskan bahwa data ordinal merupakan data yang memiliki peringkat atau urutan. Angka yang diberikan mengandung tingkatan dan digunakan untuk mengurutkan objek dari yang paling rendah sampai yang paling tinggi, atau sebaliknya. Misalnya, jawaban pertanyaan tentang kecenderungan masyarakat untuk menghadiri rapat umum pemilihan kepala daerah, mulai dari tidak pernah absen menghadiri dengan kode 5, kadang-kadang saja menghadiri dengan kode 4, kurang menghadiri dengan kode 3, tidak pernah menghadiri dengan kode 2 hingga tidak ingin menghadiri sama sekali dengan kode 1.

Teknomo (2006) dalam Elvia Budianita (2013) selanjutnya menjelaskan, setelah jajaran dinormalisasi, jarak dapat dihitung sebagai variabel kuantitatif. Jarak antara dua benda yang diwakili oleh variabel ordinal dapat ditentukan dengan mengubah skala ordinal ke skala rasio dengan melakukan langkah-langkah berikut:

1. Mengkonversi nilai ordinal ke rank ( $r = 1$  hingga  $R$ ).
2. Normalisasi peringkat ke nilai standar dari nol sampai satu  $[0,1]$  dengan persamaan:

$$X = \frac{r-1}{R-1} \quad (2.8)$$

3. Jarak dapat dihitung dengan memperlakukan nilai ordinal sebagai variabel kuantitatif (diantaranya dapat menggunakan persamaan jarak *euclidean*, *city blok*, *chebyshev*, *minkowski*, *canberra*, sudut pemisahan, dan koefisien korelasi).

Contoh:

Kuesioner untuk meminta tingkat kepuasan dalam hal keselamatan, kenyamanan, kemudahan, dan kedekatan untuk dua lokasi taman A dan B. Setiap taman memiliki 5 nilai tingkat kepuasan yaitu: -2 = sangat puas, -1 = tidak puas, 0 = ketidakpedulian, 1 = puas, 2 = sangat puas. Misalkan jawaban dari responden dapat dilihat pada Tabel 2.2.

**Tabel 2. 2 Contoh jawaban responden**

Taman	Keselamatan	Kenyamanan	Kemudahan	Kedekatan
A	-2	1	0	2
B	0	1	-1	1

Maka untuk mengukur jarak perbedaan taman A dan B adalah sebagai berikut:

1. Mengubah skala ordinal ke skala rasio. Indeks asli ( $i = -2$  sampai  $2$ ) diatur dan diubah menjadi rank ( $r = 1$  sampai  $5$ ).
2. Pangkat tertinggi adalah  $R = 5$ . Kemudian rank tersebut dinormalisasi ke dalam nilai  $[0,1]$ . Hasil normalisasi dapat dilihat pada Gambar 2.6.



**Gambar 2. 10 Hasil normalisasi jawaban responden (Teknomo, 2006 dalam Elvia Budianita, 2013)**

Berdasarkan hasil normalisasi tersebut, maka masing-masing taman A dan B memiliki nilai baru yakni taman A = 0, 3/4, 1/2, 1 dan taman B = 1/2, 3/4, 1/4, 3/4 sehingga jarak perbedaan antara taman A dan B dapat dihitung dengan persamaan *euclidean* sebagai berikut:

$$d_{AB} = \sqrt{(1/2 - 0)^2 + (3/4 - 3/4)^2 + (1/4 - 1/2)^2 + (3/4 - 1)^2}$$

$$= \sqrt{(0.25 + 0 + 0.0625 + 0.0625)} = 0.612$$

## 2.4 Akademik

Akademisi, atau Akademik adalah istilah umum bagi komunitas mahasiswa dan cendekiawan terlibat dalam sebuah pendidikan tinggi dan penelitian yang dilaksanakan. Akademi dalam pendidikan bahasa Indonesia merupakan salah satu bentuk perguruan tinggi selain politeknik, sekolah tinggi, institut, dan universitas. Akademik adalah perguruan tinggi yang menyelenggarakan pendidikan vokasi dalam satu cabang atau sebagian cabang ilmu pengetahuan, teknologi atau seni. Akademik merupakan sebuah tempat yang tersusun dengan segala sistematis perguruan tinggi yang digunakan dalam melaksanakan semua aktifitas belajar mengajar. Data Akademik adalah data yang terdapat dalam sebuah perguruan tinggi yang dapat dipertanggung jawabkan untuk dipakai dalam menjalankan aktifitas di perguruan tinggi tersebut

## 2.5 Indeks Prestasi Dan Jalur Penerimaan

Indeks Prestasi (IP) adalah penilaian keberhasilan studi semester yang dilakukan pada tiap akhir semester. Sedangkan jalur penerimaan adalah jalur seleksi penerimaan mahasiswa untuk memasuki perguruan tinggi, ada beberapa jalur penerimaan mahasiswa untuk memasuki perguruan tinggi yaitu seperti PBUD,

SNMPTN, UMJM. Lama studi dari mahasiswa sangat penting bagi mahasiswa itu sendiri dan juga jurusan untuk menentukan tindakan dalam menjaga kinerja pendidikan agar tetap baik, dan untuk menjadi lebih baik lagi. Ada 3 katagori lama masa studi yaitu katagori cepat, sedang, dan lambat, untuk katagori cepat = 4 tahun, sedang = 5 tahun, lambat =  $\geq 6$  tahun (Wiranata. 2016).

## 2.6 Penelitian Terkait

Tabel 2. 3 Penelitian Terkait

No	Peneliti	THN	Judul Penelitian	Metode	Kesimpulan
1	Aflim Ananda Putra	2015	Penerapan Metode <i>Learning Vector Quantization 2</i> (LVQ2) untuk Menentukan Penyakit Cabai	<i>Learning Vector Quantization 2</i> (LVQ2)	Hasil Pengujian yang berdasarkan 66 data latih dan 11 data uji. Dengan total uji yang dilakukan sebanyak 16 kali pengujian dengan mengombinasi 4 nilai leaning rate dan 4 window yang berbeda mendapatkan akurasi sebesar 97.22% dalam mengenali pola jenis penyakit cabai.
2	Wida Pitriani	2015	Mengklasifikasikan Tingkat Pre-Eklamsi Pada Ibu Hamil Dengan Menerapkan Jaringan Syaraf Tiruan Variasi <i>Learning Vector Quantization 2</i> (LVQ2)	<i>Learning Vector Quantization 2</i> (LVQ2)	Dalam Penelitian ini hasil pengujian parameter menyatakan bahwa diagnosa tingkat pre-eklamsi pada ibu hamil dengan metode variasi LVQ2 sesuai dengan yang diharapkan (target). Bila dipersentasekan hasil akurasi pengujiannya data uji adalah 90% dan rata-rata akurasinya mencapai 97.75%.
3	Wan Roni Aryawibowo	2015	Penerapan <i>Learning Vector Quantization 2</i> (LVQ2) untuk Mendeteksi Penyakit Infeksi Saluran	<i>Learning Vector Quantization 2</i> (LVQ2)	

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

No	Peneliti	THN	Judul Penelitian	Metode	Kesimpulan
			Pernapasan Akut (ISPA)		
4	Elvia Budianita,	2013	Penerapan Variasi Jaringan Syaraf Tiruan <i>Learning Vector Quantization 2</i> untuk Klasifikasi Status Gizi Anak	<i>Learning Vector Quantization 2</i> (LVQ2)	kesimpulan di penelitian tersebut menyatakan bahwa algoritma LVQ2 lebih baik diterapkan untuk klasifikasi status gizi anak dibandingkan dengan algoritma LVQ1. Nilai parameter yang di gunakan pada LVQ2 meliputi inputan learning rate ( $\alpha$ ) = 0.05, nilai minimal learning rate (Min a) = 0.02, nilai pengurangan $\alpha$ adalah 0,1 dan nilai window ( $\epsilon$ ) = 0.2, merupakan nilai parameter yang sudah cukup efektif dan efisien dalam melakukan klasifikasi status gizi anak sekolah dasar karena telah sesuai dengan target yang ingin dicapai secara menyeluruh (nilai akurasi mencapai 100%). Hasil pengujian jumlah data latih antara LVQ1 dengan LVQ2 menunjukkan bahwa algoritma LVQ2 lebih baik dibandingkan dengan LVQ1 yakni nilai rata-rata akurasi dengan LVQ2 adalah 95.2% sedangkan nilai rata-rata akurasi dengan LVQ1 adalah 88%.
5	Nugroho	2011	mengenai implementasi jaringan syaraf tiruan LVQ2	<i>Learning Vector</i>	Hasil pengujian penelitian ini adalah penggunaan window pada jaringan syaraf

Hak cipta milik UIN Suska Riau  
State Islamic University of Sultan Syarif Kasim Riau

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

No	Peneliti	THN	Judul Penelitian	Metode	Kesimpulan
			dalam melakukan deteksi wajah.	<i>Quantization</i> 2 (LVQ2)	tiruan LVQ2 memberikan pengaruh positif yakni dapat meningkatkan identification rate sebesar 4.16% pada saat menggunakan window berukuran 0.1 bila dibandingkan tanpa menggunakan window (LVQ1) dimana tingkat identification rate-nya hanya mencapai 79.14% sedangkan LVQ2 sebesar 83.33%.
	Aldi Syahputra	2015	Mengenai perediksi lama masa study menggunakan metode naive bayes	metode naive bayes	Kesimpulan yang di dapat menggunakan metode naive bayes classifier ini memberikan hasil klasifikasi mahasiswa dalam menyelesaikan studi nya kedalam kategori cepat, sedang dan lambat dengan tingkat akurasi mencapai 59.16 %

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.