

## BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini akan dijelaskan kajian pustaka berdasarkan penelitian sebelumnya dan juga dasar teori yang memiliki hubungan dengan penelitian ini, yaitu kasus penyakit, demam typhoid-paratipoid, peramalan, *backpropagation*, algoritma genetika.

### 2.1 Kajian Pustaka

Kajian ini dilakukan dengan mengkaji penelitian yang sudah dilakukan sebelumnya yang ditunjukkan dengan Tabel 2.1. Penelitian yang dilakukan oleh Havaluddin & Alfred (2015) meneliti tentang peramalan menggunakan data *time series*, digunakan metode jaringan syaraf tiruan *backpropagation* yang dioptimasi dengan algoritma genetika. Hasil yang diperoleh pada penelitian ini menunjukkan bahwa gabungan dari GA-BPNN (*Genetic Algorithm Backpropagation Neural Network*) memiliki nilai MSE (*Mean Square Error*) yang lebih kecil dibandingkan dengan hanya menggunakan metode *backpropagation*. Operator dari algoritma genetika dapat digunakan dengan tepat untuk mengoptimasi BPNN, sehingga dapat membantu masalah konvergensi dini.

Pada penelitian yang membahas tentang peramalan beban puncak algeria menggunakan model *time series* berdasarkan jaringan syaraf tiruan *backpropagation* oleh Abdellah & Djamel (2013) yang meneliti tentang peramalan beban tenaga listrik tertinggi dengan data *time series* menggunakan metode jaringan syaraf tiruan *backpropagation*. Penelitian ini menggunakan dataset sebanyak 23 bulan dari maret 2010 sampai februari 2012. Pada hasil trainingnya didapatkan hasil output yang mendekati target akan tetapi kesimpulannya menyatakan bahwa perlu tambahan dataset agar proses peramalan memiliki akurasi yang lebih baik.

Pada penelitian yang dilakukan oleh Yizhen et al. (2015) tentang peramalan saham di Shanghai berdasarkan metode algoritma genetika dan *backpropagation*. Hasil dari penelitian ini menunjukkan bahwa optimasi bobot serta bias dari metode *backpropagation* menggunakan algoritma genetika dapat meningkatkan akurasi dari peramalan saham.

Penelitian selanjutnya dilakukan oleh Wang et al. (2011) yang meneliti tentang peramalan beban listrik menggunakan metode *backpropagation* yang dikombinasikan dengan algoritma genetika. Hasil penelitian ini menunjukkan bahwa dengan pengoptimasian menggunakan algoritma genetika dapat mengatasi masalah pemilihan acak bobot awal dari struktur jaringan sehingga dapat menyelesaikan masalah yang menyebabkan terjebak lokal optimum dengan mudah.

Penelitian lainnya yang dilakukan oleh Chen et al. (2014) yang meneliti tentang peramalan pergerakan longsor yang membandingkan dua metode jaringan syaraf tiruan yaitu GA-RBFN (*Genetic Algorithm Radial Basis Function*

Network) dan GA-BPNN. Penelitian ini menunjukkan bahwa rata-rata *error* GA-BPNN lebih kecil daripada GA-RBFN yaitu sebesar 0.25262 dan 0.3259.

**Tabel 2.1 Kajian Pustaka**

No.	Judul	Objek	Metode	Hasil
1.	<i>A Genetic-Based Backpropagation Neural Network for Forecasting in Time-Series Data</i> (Havilludin dan Alfred, 2015)	Data Time Series (Network Traffic)	<i>Backpropagation Neural Network dan Genetic Algorithm</i>	Operator dari algoritma genetika dapat digunakan dengan tepat untuk mengoptimasi BPNN, sehingga dapat membantu masalah konvergensi dini
2.	<i>Forecasting the Algerian Load Peak Profile Using Time Series Model Based On Backpropagation Neural Networks</i> (Abdellah dan Djamel, 2013)	Beban tenaga listrik tertinggi di Algeria	<i>Backpropagation Neural Network</i>	Hasil output mendekati target akan tetapi perlu tambahan dataset agar hasil yang didapat lebih baik
3.	<i>The Forecasting of Shanghai Index trend Based on Genetic Algorithm and Back Propagation Artificial Neural Network Algorithm</i> (Yizen et al, 2011)	Trending index pasar saham di Shanghai	<i>Backpropagation Neural Network dan Genetic Algorithm</i>	optimasi bobot serta bias dari metode <i>backpropagation</i> menggunakan algoritma genetika dapat meningkatkan akurasi dari peramalan saham
4.	<i>Optimizing of BP Neural Network Based on Genetic Algorithm in Power Load Forecasting</i> (Wang et al, 2011)	Beban tenaga listrik	<i>Backpropagation Neural Network dan Genetic Algorithm</i>	Dengan pengoptimasian menggunakan algoritma genetika dapat mengatasi masalah pemilihan acak bobot awal dari struktur jaringan sehingga

				dapat menyelesaikan masalah yang menyebabkan terjebak lokal optimum dengan mudah
5.	<i>Genetic Algorithm based Neural Network for the Displacement of Landslide Forecasting</i> (Chen et al, 2014)	Pergerakan longsor	<i>Backpropagation Neural Network</i> dan <i>Genetic Algorithm</i>	Dengan membandingkan GA-RBFN dan GA-BPNN bahwa GA-BPNN memiliki hasil lebih baik daripada GA-RBFN

## 2.2 Kasus Penyakit

Menurut Riset Kesehatan Daerah tahun 2007 (Riskesmas) jumlah kasus penyakit adalah kelompok penyakit di Indonesia yang sudah dikategorikan berdasarkan kasus penyakit yang ada. Kelompok penyakit yang menyebabkan kematian antara lain:

1. Penyakit yang menular.
2. Penyakit yang tidak menular.
3. Gangguan perinatal atau maternal.
4. Disabilitas dan cedera.

### 2.2.1 Klasifikasi Kasus Penyakit

Standar pengelompokan penyakit di skala internasional dikeluarkan oleh *World Health Organization* (WHO). Pengelompokan penyakit ini disebut sebagai ICD atau *International Statistical Classification of Disease and Related Health Problem* yang sudah mengalami direvisi sebanyak sepuluh kali (WHO, 2016). Berdasarkan standar internasional di Indonesia, kategori penyakit ditunjukkan dengan Tabel 2.2:

**Tabel 2.2 Klasifikasi Penyakit pada ICD10**

I	<a href="#">A00-B99</a>	Golongan Penyakit Infeksi serta parasit
II	<a href="#">C00-D48</a>	Golongan Penyakit Neoplasma
III	<a href="#">D50-D89</a>	Gangguan sistem imun yang termasuk dalam penyakit pada darah dan organ pembentuk darah
IV	<a href="#">E00-E90</a>	Endokrin, gangguan metabolik serta nutrisi
V	<a href="#">F00-F99</a>	Gangguan jiwa serta perilaku

VI	<a href="#">G00-G99</a>	Penyakit gangguan sistem syaraf
VII	<a href="#">H00-H59</a>	Penyakit pada adnexa dan mata
VIII	<a href="#">H60-H95</a>	Penyakit pada telinga dan mastoid
IX	<a href="#">I00-I99</a>	Penyakit yang terjadi di sistem sirkulasi
X	<a href="#">J00-J99</a>	Penyakit yang terjadi di sistem pernapasan
XI	<a href="#">K00-K93</a>	Penyakit yang terjadi di sistem pencernaan
XII	<a href="#">L00-L99</a>	Penyakit yang terjadi di kulit dan jaringan subcutaneous
XIII	<a href="#">M00-M99</a>	Penyakit yang terjadi di sistem musculoskeletal
XIV	<a href="#">N00-N99</a>	Penyakit yang terjadi di sistem saluran kemih dan genital
XV	<a href="#">O00-O99</a>	Kelahiran serta kehamilan
XVI	<a href="#">P00-P96</a>	Keadaan yang berasal dari periode perinatal
XVII	<a href="#">Q00-Q99</a>	Malformasi kongenital, kelainan chromosom serta deformasi
XVIII	<a href="#">R00-R99</a>	Tanda, gejala, kelainan lab dan kelainan klinik yg tidak terdapat pada klasifikasi lain
XIX	<a href="#">S00-T98</a>	Cedera, keracunan dan beberapa penyebab yang berasal dari luar
XX	<a href="#">V01-Y98</a>	Penyebab kematian eksternal dan morbiditas
XXI	<a href="#">Z00-Z99</a>	Faktor faktor yang memengaruhi status kesehatan juga hubungannya dengan jasa kesehatan
XXII	<a href="#">U00-U99</a>	Kode dengan kegunaan khusus

Sumber: WHO, 2016

Pada tanggal 19 Februari 1996, di Indonesia dimulai penggunaan ICD yang didasarkan oleh keputusan Direktur Jendral Pelayanan Medik (Dirjen Yanmed) no: HK 00 . 05. 1. 4. 0074 mengenai penggunaan klasifikasi internasional tentang penyakit revisi ke sepuluh (ICD-10).

### 2.2.2 Penyakit Demam *Typoid-paratyphoid*

Salah satu penyakit yang dikelompokkan pada ICD10 adalah penyakit demam *typoid-paratyphoid*. Menurut WHO (2016) pada bagian *chapter 1* dari klasifikasi penyakit di ICD10 demam *typoid-paratyphoid* merupakan salah satu penyakit yang termasuk kedalam kelompok penyakit infeksi dan parasit. Penyakit demam typhoid ini tergolong memiliki potensi bahaya yang dapat mengakibatkan kematian. Penyebab dari penyakit ini adalah bakteri *Salmonella enterica* serotipe Typhi. Sementara penyebab dari penyakit demam adalah bakteri yang berbeda yaitu *Salmonella enterica* serotipe Paratyphi A, B (tartrate negatif), atau C (Newton et al., 2015).

## **2.3 Peramalan**

Peramalan (*forecasting*) merupakan perkiraan atau dugaan akan suatu peristiwa pada masa mendatang (Supranto, 2000). Menurut Nachrowi (2004) peramalan merupakan suatu teknik yang digunakan untuk memprediksi atau memperkirakan nilai yang akan terjadi di masa depan melalui data yang relevan pada masa lalu dan masa sekarang. Peramalan dibutuhkan untuk memprediksi kuantitas, waktu, lokasi yang dibutuhkan pada masa depan.

Menurut Sinaga (2007) peramalan dibutuhkan oleh perusahaan atau instansi untuk dapat mengetahui serta untuk menentukan kebutuhan akan diperlukan atau kapan suatu peristiwa akan terjadi di masa yang akan datang, sehingga persiapan akan suatu tindakan atau kebijakan dapat dilakukan. Menurut Hendri (2010) contoh manfaat dari peramalan untuk perusahaan adalah sebagai berikut:

1. Sebagai penentu untuk membantu penyusunan anggaran perusahaan.
2. Sebagai pedoman untuk membantu pengendalian persediaan serta pengendalian produksi perusahaan.
3. Sebagai upaya peningkatan pelayanan konsumen berdasarkan hasil evaluasi.
4. Sebagai dasar pertimbangan yang digunakan pada perencanaan jangka panjang contohnya perluasan perusahaan.

### **2.3.1 Jenis-jenis Peramalan**

Terdapat dua jenis peramalan, yaitu berdasarkan jangka waktu serta berdasarkan metode.

#### **2.3.1.1 Berdasarkan Jangka Waktu**

1. Peramalan jangka pendek, peramalan yang diladakan dalam jangka waktu kurang dari satu tahun. Contohnya digunakan pada rencana pembelian dan penjadwalan kerja.
2. Peramalan jangka menengah, peramalan yang diladakan dalam jangka waktu kurang dari tiga tahun. Contohnya perencanaan anggaran produksi perusahaan.
3. Peramalan jangka panjang, peramalan yang diladakan dalam jangka waktu lebih dari tiga tahun. Contohnya digunakan dalam perencanaan produksi produk baru.

#### **2.3.1.2 Berdasarkan Metode**

1. Peramalan kualitatif, peramalan yang menggunakan data non-numerik dimana metode ini bersifat subjektif. Hasil peramalan ini bergantung kepada opini atau pendapat para ahli serta pembuat atau penyusunnya (Makridakis, 1998).
2. Peramalan kuantitatif, peramalan yang menggunakan data numerik dimana metode ini bersifat objektif. Hasil peramalan ini bergantung pada data masa lalu serta masa sekarang (Makridakus, 1998).

Terdapat dua metode kuantitatif yaitu:

- a) *Time series*, metode yang meramalkan data berbentuk deret waktu.
- b) Causal, metode yang meramalkan data terkait dengan hubungan sebab akibat.

### 2.3.2 Tahapan Peramalan

Menurut Daly (2010) proses peramalan memiliki tahap sebagai berikut:

1. Mendefinisikan tujuan peramalan.
2. Mencari dan memilih teori peramalan yang berhubungan.
3. Mencari dan mengumpulkan data.
4. Mencari dan memilih metode yang akan digunakan untuk proses peramalan.
5. Menganalisis data.
6. Mendefinisikan estimasi model sementara.
7. Mengevaluasi dan merevisi model.
8. Menampilkan hasil peramalan.

### 2.3.3 Ukuran Akurasi Hasil Peramalan

Hasil peramalan yang didapat pada saat melakukan peramalan tidaklah selalu sesuai dengan keinginan atau kenyataan sehingga terjadi penyimpangan atau kesalahan. Kesalahan ini bukan hanya disebabkan oleh unsur *error*, tetapi ketidakmampuan model peramalan untuk mengenali unsur lain yang ada pada deret data. Faktor tidak diduga dapat menjadi penyebab kesalahan ini dikarenakan tidak ada metode peramalan yang bisa menghasilkan hasil peramalan yang akurat ataupun disebabkan oleh ketidakmampuan metode peramalan untuk mengenali komponen-komponen yang ada dalam data (Bowerman et al., 1987 dalam Sungkawa, 2011). MSE merupakan salah satu metode yang digunakan untuk menghitung akurasi hasil peramalan dengan cara menghitung perbedaan suatu data aktual yang akan diestimasi dengan sebuah estimator (Deborah & Prathap, 2014). Semakin kecil nilai MSE, maka tingkat akurasi peramalan semakin baik. Perhitungan MSE ditunjukkan pada Persamaan 2.1:

1. *Mean Squared Error* (MSE)

$$MSE = \frac{1}{n} \sum_{i=1}^n (y'_i - y_i)^2 \quad 2.1$$

dimana

$y'_t$  = data hasil prediksi periode  $t$

$y_t$  = data aktual periode  $t$

$n$  = jumlah data

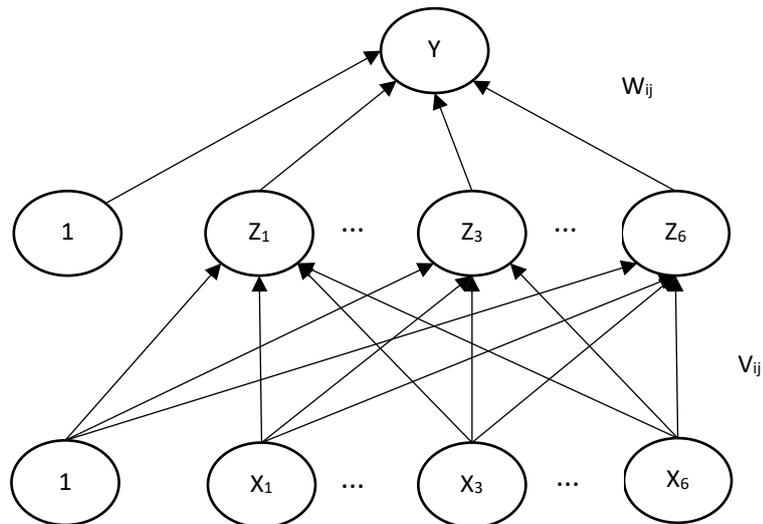
## 2.4 Backpropagation

*Backpropagation* merupakan salah satu metode yang ada dalam jaringan syaraf tiruan. Penggunaan serta penerapan metode *backpropagation* ini tergolong

dalam algoritma pelatihan yang bersifat *supervised*. Proses pelatihan metode *backpropagation* didasarkan dari hubungan sederhana yaitu, jika hasil yang dikeluarkan salah maka penimbang (*weight*) dikoreksi agar galatnya dapat diperkecil serta selanjutnya diharapkan mendekati hasil yang benar (Kosasi, 2014).

### 2.4.1 Arsitektur Jaringan Syaraf Tiruan *Backpropagation*

Terdapat beberapa layer, yaitu layer *input*, *hidden layer*, dan layer *output* dalam arsitektur jaringan syaraf tiruan *backpropagation*. Proses dari metode *backpropagation* adalah setelah menerima *input* atau masukan pada layer *input*, *input* ini akan diproses propagasi melewati setiap layer di atasnya hingga menghasilkan keluaran atau *output* dari jaringan. Nilai *error* didapatkan dari perbandingan *output* jaringan dengan target *output*. Kemudian dengan menggunakan bobot awal, jaringan akan melewatkan turunan dari nilai *error* tersebut menuju *hidden layer*. Setelah itu akan dilakukan perhitungan jumlah bobot dari *error* hasil propagasi sebelumnya oleh setiap *neuron* yang ada pada *hidden layer*. Sesudah menemukan besarnya nilai *error*, maka dilakukan perubahan nilai bobot untuk mengurangi nilai *error* oleh *neuron-neuron* tersebut. Proses ini akan terus dilakukan hingga nilai *error* yang didapatkan oleh jaringan mendekati nilai nol (Giantara, 2013). Arsitektur jaringan saraf tiruan *backpropagation* ditunjukkan dengan Gambar 2.1:



Gambar 2.1 Arsitektur Jaringan Syaraf Tiruan *Backpropagation*

Keterangan:

- X = layer *input*
- Z = *hidden layer*
- Y = layer *output*
- W,V = bobot
- 1 = bias

## 2.4.2 Fungsi Aktivasi

Ada beberapa karakteristik penting dalam fungsi aktivasi *backpropagation* yaitu: *continious, differentiable, monotpnically, non-decreaseing*, dan *easy to compute* untuk efisiensi perhitungan (Fausett, 1994). Terdapat dua fungsi aktivasi pada *backpropagation*, yaitu:

### 1. Binary Sigmoid

Fungsi *binary* sigmoid digunakan untuk jaringan yang membutuhkan nilai keluaran yang mempunyai jarak nilai dari 0 sampai 1 yang didefinisikan dengan Persamaan 2.2:

$$y = f(x) = \frac{1}{1+exp^{-x}} \quad 2.2$$

dimana:

$y, f(x)$  = hasil sinyal *output* dari  $x$   
 $exp^{-x}$  = eksponensial dari nilai  $-x$

### 2. Bipolar Sigmoid

Fungsi *bipolar* sigmoid digunakan untuk jaringan yang membutuhkan nilai keluaran yang mempunyai jarak nilai dari 1 sampai -1 yang didefinisikan pada Persamaan 2.3:

$$y = f(x) \frac{2}{1+exp^{-x}} - 1 \quad 2.3$$

dimana:

$y, f(x)$  = hasil sinyal *output* dari  $x$   
 $exp^{-x}$  = eksponensial dari nilai  $-x$

## 2.4.3 Tahapan Pelatihan Algoritma *Backpropagation*

Secara umum langkah-langkah metode *backpropagation* adalah inialisasi bobot awal, proses *feedforward*, menghitung *backpropagation error*, memperbarui bobot dan bias (Haviluddin & Alfred, 2015). Berikut tahapan proses pada *backpropagation* menurut Fausett (1994):

### 1. Inialisasi bobot awal serta bias awal

Bobot serta bias akan di inisialiasasi secara *random* dengan interval 0 -1 atau 1.

### 2. Proses *feedforward*

- a) Setiap unit *input* ( $X_i, i=1, \dots, n$ ) menampung sinyal *input*  $x_i$  serta menyebarkan sinyal ini keseluruhan unit yang berada di *hidden layer*. *Input* yang digunakan merupakan data *training*.
- b) Setiap *hidden* unit ( $Z_j, j=1, \dots, p$ ) melakukan penjumlahan pada sinyal-sinyal *input* yang dikalikan dengan bobot  $v_{ij}$ , dengan Persamaan 2.4:

$$z_{inj} = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad 2.4$$

dimana

$z_{inj}$  = sinyal masuk pada *hidden layer*

$x_i$  = sinyal *input*

$v_{0j}$  = bias pada *hidden layer*

$v_{ij}$  = bobot pada *hidden layer*

- c) Menghitung fungsi aktivasi  $Z_j$  menggunakan *binary* menggunakan Persamaan 2.5, kemudian mengirimkan sinyal *output* ke semua unit pada unit *output*.

$$Z_j = f(z_{inj}) \quad 2.5$$

- d) Setiap unit *output* ( $y_k, k = 1, \dots, m$ ) melakukan penjumlahan sinyal-sinyal *input* yang sudah berbobot, termasuk biasanya menggunakan Persamaan 2.6:

$$y_{ink} = w_{0k} + \sum_{j=1}^n z_j w_{ij} \quad 2.6$$

dimana

$y_{ink}$  = sinyal masuk pada *output*

$w_{0k}$  = bobot serta bias ke *output layer*

$z_j$  = fungsi aktivasi lapisan tersembunyi

$w_{ij}$  = bobot lapisan tersembunyi

- e) Menghitung fungsi aktivasi menggunakan *binary sigmoid* yang menghitung sinyal *output* dari *hidden unit* menggunakan persamaan 2.7:

$$Y_k = f(y_{ink}) \quad 2.7$$

### 3. Menghitung *backpropagation error*

- a) Setiap unit *output* ( $Y_k, k=1, \dots, m$ ) menampung suatu pola target yang sesuai dengan pola *input* data *training* untuk menghitung kesalahan yang dihasilkan jaringan dengan Persamaan 2.8:

$$\delta_k = (t_k - y_k) y_k (1 - y_k) \quad 2.8$$

Dimana

$\delta_k$  = faktor koreksi *output layer*

$t_k$  = data *training*

$y_k$  = *output* pelatihan

- b) Perhitungan perubahan bobot  $W_{jk}$  yang akan merubah bobot  $W_{jk}$  menggunakan Persamaan 2.9:

$$\Delta W_{jk} = \alpha \delta_k z_j \quad 2.9$$

dimana

$\Delta W_{jk}$  = delta perubahan bobot *output layer*

$\alpha$  = *learning rate*

$\delta_k$  = faktor koreksi *output layer*

$z_j$  = fungsi aktivasi pada *hidden layer*

- c) Perhitungan perubahan bias  $W_{0k}$  yang akan merubah bias  $W_{0k}$  menggunakan Persamaan 2.10:

$$\Delta W_{0k} = \alpha \delta_k \quad 2.10$$

dimana

$\Delta W_{0k}$  = delta perubahan bias *output layer*

$\alpha$  = *learning rate*

$\delta_k$  = faktor koreksi *output layer*

- d) Perhitungan faktor koreksi\_in *hidden unit*.

$$\delta_{inj} = \sum_{k=1}^m \delta_k w_{ij} \quad 2.11$$

dimana

$\delta_{inj}$  = faktor koreksi\_in *hidden unit*

$\delta_i$  = faktor koreksi *output layer*

$w_{ij}$  = bobot hidden layer ke lapisan keluaran

- e) Menghitung faktor koreksi *hidden unit*.

$$\delta_j = \delta_{inj} z_j (1 - z_j) \quad 2.12$$

dimana

$\delta_j$  = faktor koreksi *hidden unit*

$\delta_{inj}$  = faktor koreksi\_in *hidden unit*

$z_j$  = fungsi aktifasi *hidden layer*

f) Menghitung koreksi bobot *hidden laer*.

$$\Delta v_{jk} = \alpha \delta_j x_j \quad 2.13$$

dimana

$\Delta v_{jk}$  = koreksi bobot *hidden layer*

$\alpha$  = *learning rate*

$\delta_j$  = faktor koreksi *hidden unit*

$x_j$  = sinyal *input*

g) Menghitung koreksi bias *hidden layer*.

$$\Delta v_{0j} = \alpha \delta_j \quad 2.14$$

dimana

$\Delta v_{0j}$  = koreksi bias *hidden layer*

$\alpha$  = *learning rate*

$\delta_j$  = faktor koreksi *hidden unit*

4. Menghitung perubahan bobot serta bias

a) Menghitung bobot baru dari *input layer* ke *hidden layer*.

$$v_{ij}(\mathbf{baru}) = v_{ij}(\mathbf{lama}) + \Delta v_{ij} \quad 2.15$$

dimana

$v_{ij}(\mathbf{baru})$  = bobot baru *input layer* ke *hidden layer*

$v_{ij}(\mathbf{lama})$  = bobot lama *input layer* ke *hidden layer*

$\Delta v_{ij}$  = koreksi bobot *hidden layer*

b) Menghitung bias baru dari *input layer* ke *hidden layer*.

$$v_{0j}(\mathbf{baru}) = v_{0j}(\mathbf{lama}) + \Delta v_{0j} \quad 2.16$$

dimana

$v_{0j}(\mathbf{baru})$  = bias baru *input layer* ke *hidden layer*

$v_{0j}(\mathbf{lama})$  = bias lama *input layer* ke *hidden layer*

$\Delta v_{0j}$  = koreksi bias *hidden layer*

c) Menghitung bobot baru *hidden layer* ke *output layer*.

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad 2.17$$

dimana

$w_{jk}(\text{baru})$  = bobot baru *hidden layer* ke *output layer*

$w_{jk}(\text{lama})$  = bobot lama *hidden layer* ke *output layer*

$\Delta w_{jk}$  = koreksi bobot *output layer*

d) Menghitung bias baru dari *hidden layer* ke *output layer*.

$$w_{0k}(\text{baru}) = w_{0k}(\text{lama}) + \Delta w_{0k} \quad 2.18$$

dimana

$w_{0k}(\text{baru})$  = bias baru *hidden layer* ke *output layer*

$w_{0k}(\text{lama})$  = bias lama *hidden layer* ke *output layer*

$\Delta w_{0k}$  = koreksi bias *output layer*

## 5. Menghitung nilai MSE

Setelah melakukan semua tahapan diatas dan mendapatkan hasil peramalan, maka akan dilakukan proses perhitungan kesalahan dari algoritma *backpropagation* dengan menggunakan perhitungan MSE pada Persamaan 2.1.

## 2.5 Algoritma Genetika

Algoritma genetika merupakan pencarian heuristic yang meniru proses dari seleksi alam yang pertama kali ditemukan oleh John Holland (1992). Algoritma genetika dibuat berdasarkan proses evolusi manusia yang memastikan kelangsungan hidup dari suatu generasi. Algoritma genetika dirancang dan tergabung dalam model yang dibuat untuk memilih fitur terbaik dalam meningkatkan kinerja model peramalan (Haidar & Verma, 2016).

### 2.5.1 Tahapan Algoritma Genetika

Secara sederhana tahapan dalam algoritma genetika adalah inialisasi populasi awal kemudian anggota dari populasi tersebut akan melanjutkan proses evolusi. Proses evolusi ini terdiri dari reproduksi, evaluasi, dan seleksi. Setiap algoritma genetika mempunyai kriteria berhenti yang berbeda-beda yaitu, setelah generasi ke-n, tidak ada peningkatan hasil setelah beberapa waktu, dan hasil yang diinginkan telah tercapai (Haidar & Verma, 2016).

#### 1. Inialisasi

Proses inialisasi merupakan proses pembangkitan individu yang dilakukan secara acak sesuai dengan jumlah populasi yang ditentukan (*popSize*). Masing-masing individu memiliki *chromosome* yang berbeda-beda sesuai dengan solusi permasalahan yang akan

dilakukan(Mahmudy, 2015). Berikut adalah contoh populasi yang terdiri dari lima individu yang ditunjukkan dengan Tabel 2.3.

**Tabel 2.3 Contoh Populasi**

P	Chromosome						fitness
	x1	x2	x3	x4	x5	x6	
P1	0.4	-0.2	0.5	-0.2	-0.6	0.1	0.009796
P2	0.7	-0.5	0.4	-0.3	0.6	-0.5	0.011808
P3	0.2	-0.8	-0.2	-0.9	-0.9	-0.3	0.006726
P4	0.6	0.7	-0.1	-0.4	0.5	-0.3	0.00599
P5	0.1	0.5	0.6	-0.8	0.8	0.1	0.013481

Dimana

P1-P5 : Induk atau *parent*.

x1-x6 : *chromosome*.

*fitness* : Nilai kebugaran dari setiap individu.

## 2. Reproduksi

Proses reproduksi merupakan proses yang bertujuan untuk menghasilkan keturunan atau individu baru(*offspring*) dari sebuah populasi. Dalam proses reproduksi terdapat dua macam cara yaitu *crossover* dan mutasi. Sebelum melakukan proses reproduksi terlebih dahulu dilakukan penentuan *crossover rate(cr)* dan *mutation rate(mr)* yang digunakan untuk menentukan rasio keturunan yang dihasilkan dari proses *crossover* dan mutasi sesuai dengan *popSize* dengan cara mengkalikan *cr* dan *mr* terhadap *popSize*(Mahmudy, 2015).

Salah satu metode *crossover* yang sering digunakan adalah *one-cut point crossover* dengan cara menukar gen sesuai dengan titik potong yang ditentukan. Misal ditentukan  $cr=0,5$  dan  $popSize=5$  maka ada  $0,4 \times 5=2$  *offspring* yang dihasilkan. Misalkan P1 dan P3 terpilih menjadi induk dan dengan menggunakan metode *one-cut point* dengan titik potong pada posisi ke empat, maka akan dihasilkan *offspring* C1 dan C2 ditunjukkan dengan Gambar 2.2.

P1	0.4	0.4	0.4	0.4	0.4	0.4
P3	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2
C1	0.4	0.4	0.4	0.4	-0.2	-0.2
C2	-0.2	-0.2	-0.2	-0.2	0.4	0.4

**Gambar 2.2 Proses one-cut point crossover**

Dalam proses mutasi dipilih satu induk secara acak untuk dilakukan proses mutasi. Terdapat beberapa metode dalam proses mutasi salah satunya adalah *random mutation* yaitu menambah atau mengurangi nilai

dari satu gen terpilih dengan bilangan acak yang kecil dengan menggunakan Persamaan 2.19(Mahmudy, 2015).

$$x'_i = x_i + r(max_i - min_i) \quad 2.19$$

dimana

$x_i$  = gen yang terpilih untuk mutasi

$r$  = nilai random -0,1 sampai 0,1

$max_i$  = nilai maksimum dari individu yang terpilih untuk mutasi

$min_i$  = nilai minimum dari individu yang terpilih untuk mutasi

Misalkan  $mr=0,1$  dan  $popSize=5$  maka ada  $0,1 \times 5=1$ (dibulatkan ke atas). Jika P2 dengan gen x2 terpilih sebagai induk dan nilai  $r=0,0584$  maka dihasilkan C3 ditunjukkan dengan Gambar 2.3.

P2	0.7	-0.5	0.4	-0.3	0.6	-0.5
C3	0.7	-0,42	0.4	-0.3	0.6	-0.5

**Gambar 2.3 Proses random mutation**

$$x'_i = -0,5 + 0,0584(0,7 - (-0,5))$$

$$x'_i = -0,42$$

### 3. Evaluasi

Proses evaluasi berguna untuk menghitung nilai kebugaran(*fitness*) dari masing-masing individu termasuk keturunan dari hasil reproduksi. Dari hasil reproduksi, diperoleh populasi baru ditunjukkan dengan Tabel 2.4.

**Tabel 2.4 Evaluasi**

P	Chromosome						<i>fitness</i>
	x1	x2	x3	x4	x5	x6	
P1	0.4	-0.2	0.5	-0.2	-0.6	0.1	0.009796
P2	0.7	-0.5	0.4	-0.3	0.6	-0.5	0.011808
P3	0.2	-0.8	-0.2	-0.9	-0.9	-0.3	0.006726
P4	0.6	0.7	-0.1	-0.4	0.5	-0.3	0.00599
P5	0.1	0.5	0.6	-0.8	0.8	0.1	0.013481
C1	0.4	0.4	0.4	0.4	-0.2	-0.2	0.06794
C2	-0.2	-0.2	-0.2	-0.2	0.4	0.4	0.05473
C3	0.7	-0,42	0.4	-0.3	0.6	-0.5	0.06052

### 4. Seleksi

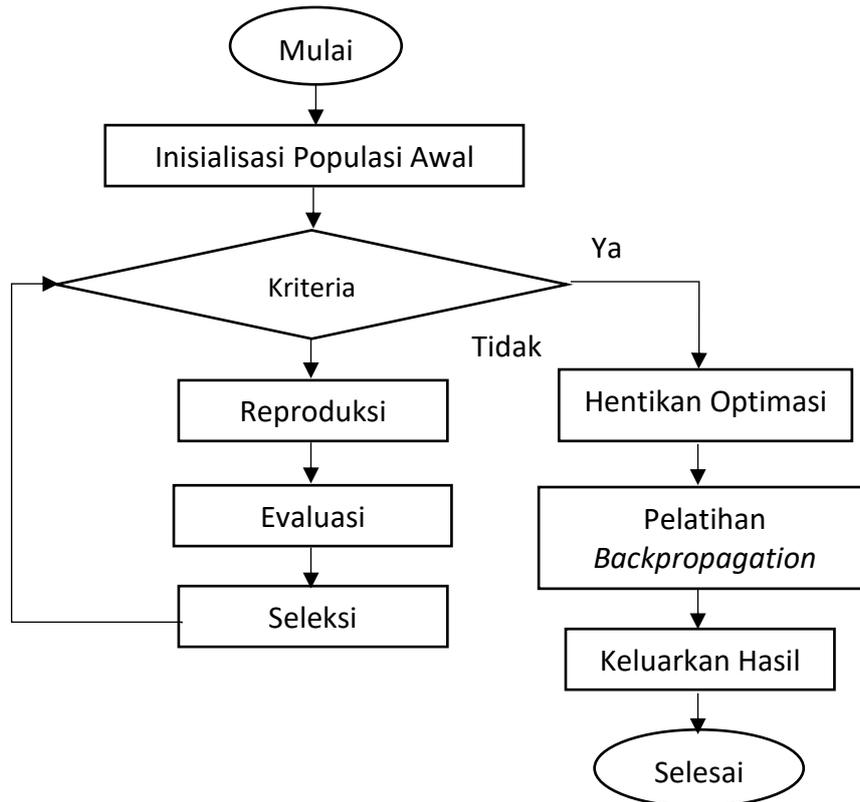
Seleksi bertujuan untuk menyaring individu yang berasal dari populasi hasil evaluasi yang bertahan ke generasi seterusnya. Salah satu metode seleksi yang sering digunakan adalah *elitism selection*. Metode ini berjalan dengan cara mengambil individu terbaik berdasarkan nilai *fitness* paling baik sesuai dengan jumlah *popSize* untuk diloloskan ke generasi

selanjutnya(Mahmudy, 2015). Berdasarkan Tabel 2.4, maka didapatkan individu hasil seleksi ditunjukkan dengan Tabel 2.5.

P(t+1)	Asal	Chromosome						fitness
		x1	x2	x3	x4	x5	x6	
P1	P1	0.4	-0.2	0.5	-0.2	-0.6	0.1	0.009796
P2	C1	0.4	0.4	0.4	0.4	-0.2	-0.2	0.06794
P3	C3	0.7	-0,42	0.4	-0.3	0.6	-0.5	0.06052
P4	C2	-0.2	-0.2	-0.2	-0.2	0.4	0.4	0.05473
P5	P5	0.1	0.5	0.6	-0.8	0.8	0.1	0.013481

## 2.6 Backpropagation-Algorithm Genetika

Kelemahan metode jaringan syaraf tiruan *backpropagation* adalah jaringan syaraf tiruan sering terjebak pada lokal minimum dikarenakan konvergensi dini (Nawi et al., 2013). Menurut beberapa penelitian yang telah diuraikan pada kajian pustaka, algoritma genetika dapat digunakan untuk proses optimasi metode *backpropagation* untuk mendapatkan parameter yang optimal. Langkah-langkah penggabungan algoritma ini dijelaskan pada Gambar 2.4:



Gambar 2.4 Diagram Alir Algoritma *Backpropagation-Algorithm Genetika*