

## BAB II

### TINJAUAN PUSTAKA DAN DASAR TEORI

#### 1.1. Tinjauan Pustaka

Penelitian-penelitian terdahulu yang terkait dengan penyakit ginjal kronis dan jenis penyakit lainnya telah banyak dilakukan, baik dari penelitian yang telah menerapkan *feature selection* maupun penelitian yang belum menerapkan *feature selection*. Beberapa contoh metode yang diterapkan untuk melakukan prediksi maupun analisa terhadap penyakit ginjal kronis adalah metode *Artificial Neural network* (ANN), metode SVM, metode *Fuzzy Expert System*, dan beberapa metode lain yang juga dapat dimanfaatkan untuk menganalisa jenis penyakit lainnya.

Dalam penelitian yang mengkomparasikan antara penggunaan metode ANN dan SVM untuk prediksi penyakit ginjal kronis, menyebutkan bahwa metode ANN memiliki kinerja yang lebih baik jika dibandingkan dengan SVM. Penelitian dilakukan dengan menggunakan *dataset Chronic Kidney Disease* (CKD) dari *UCI Machine Learning Repository*. Metode ANN yang digunakan bersamaan dengan *feature selection Correlation Coefficient* menghasilkan akurasi sebesar 99,75%, sedangkan metode SVM menghasilkan akurasi sebesar 97,75% (Almansour dkk., 2019).

Penelitian yang dilakukan dengan menerapkan beberapa jenis algoritma *data mining* sekaligus untuk dikomparasikan juga pernah dilakukan untuk memprediksi penyakit ginjal kronis. Khususnya untuk tingkatan (*stage*) penyakit ginjal kronis itu sendiri. Beberapa jenis algoritma *data mining* tersebut adalah *Probabilistic Neural networks* (PNN), *Multilayer Perceptron* (MLP), SVM dan *Radial Basis Function* (RBF). Dari penelitian tersebut diperoleh hasil bahwa metode PNN memberikan hasil akurasi yang tertinggi jika dibandingkan dengan metode lainnya yaitu sebesar 96,7% (Rady dan Anwar, 2019).

Penelitian yang dilakukan secara klinis mengenai penyakit ginjal kronis pernah dilakukan dengan menerapkan metode *Fuzzy (Fuzzy Expert System)*. Dari penelitian tersebut didapatkan hasil *output* berupa *statement* apakah pasien menderita

penyakit ginjal kronis atau tidak dari informasi yang dimasukkan ke dalam sistem. Sedangkan untuk akurasi dari sistem sendiri diperoleh sebesar 92,13%. Hasil tersebut diperoleh berdasarkan 16 parameter untuk melakukan prediksi dari penyakit ginjal kronis (Hamedan dkk., 2020).

Penelitian yang dilakukan menggunakan *Heterogeneous Modified Artificial Neural Network* (HMANN) dengan tujuan untuk melakukan deteksi awal, segmentasi dan diagnosa gagal ginjal kronis pada *Internet of Medical Things* (IoMT) menyebutkan bahwa metode HMANN dapat mencapai akurasi yang tinggi pada bagian segmentasi. Penelitian dilakukan berdasarkan dari data gambar *ultrasound* pada ginjal yang mana data tersebut dilambangkan sebagai langkah untuk *preprocessing*. Dari penelitian diperoleh hasil akurasi 0,98 dengan berdasarkan pada *training data* dan 0,93 dengan adanya *improve* pada saat *training* (Ma dkk., 2020).

Di bidang kesehatan lain, penelitian dengan menggunakan *feature selection* dan JST juga telah banyak dilakukan. Dalam penelitian dengan objek pendeteksian penyakit *Sleep Apnea* melalui sinyal EKG dengan membagi penelitian menjadi dua bagian. Bagian pertama menggunakan *Feature Extraction* (FE) yang kemudian melalui proses pengklasifikasian dengan beberapa metode *machine learning*, salah satunya adalah JST. Bagian kedua menggunakan *feature selection* jenis *Discriminant Relevance* (DR) kemudian masuk dalam proses klasifikasi kembali dengan salah satunya menggunakan JST. Dari penelitian tersebut diperoleh hasil akurasi terbaik sebesar 82,12% dengan jumlah fitur sebanyak 20 fitur dari total 84 fitur (Pinho dkk., 2019).

Penelitian lain dengan menggunakan *feature selection* dan JST juga digunakan untuk membantu ahli radiologi dalam menganalisa penyakit tumor otak dengan beberapa klasifikasi pengidentifikasian penyakitnya. Dalam penelitian tersebut menggunakan perpaduan antara *Genetic Algorithm* (GA) dan juga JST. Dari penelitian diperoleh hasil akurasi tertinggi sebesar 94,9% (Sachdeva dkk., 2016).

Penggunaan *feature selection* dan JST untuk pengoptimalan prediksi penyakit juga digunakan untuk memprediksi penyakit jantung. Dalam penelitian tersebut

menggunakan *feature selection* jenis *Differential Evolution* (DE) dan juga *hybrid Fuzzy AHP* dengan JST jenis *Feed-forward Neural network*. Dari penelitian diperoleh hasil akurasi sebesar 83%, yang mana dianggap sebagai hasil akurasi tertinggi dibandingkan dengan beberapa jenis model algoritma lain yang pernah digunakan untuk penelitian serupa (Vivekanandan dan Sriman Narayana Iyengar, 2017).

Penelitian untuk penyakit kanker payudara dengan menggunakan *hybrid feature selection* dan JST pernah dilakukan sebelumnya. Dalam penelitian tersebut menggunakan objek 457 buah fitur *2D Ultrasound images* dari diagnosa kanker payudara yang diderita pasien. Jenis *feature selection* yang digunakan diantaranya *Information gain* (IG), *Gain Ratio* (GR), *Relief F* (RLF), *Random Forest*, *Relief*, *Chi-Square* dan beberapa jenis lainnya. Dari penelitian diperoleh hasil akurasi sebesar 96,6% untuk penggunaan *Back-Propagation Artificial Neural network* (BPANN) dan 94,4% untuk penggunaan SVM (Singh dkk., 2017).

Penggunaan *feature selection* dan JST untuk penyakit lain adalah pada penelitian dengan menggunakan objek penelitian penyakit *Conjunctival Hyperemia*. Dalam penelitian tersebut bertujuan untuk mengetahui pengaruh penggunaan *feature selection* dan *class imbalance* untuk pemerinkatan otomatis. Jenis *feature selection* yang digunakan adalah *Correlation based Feature selection* (CFS). Dari penelitian tersebut diperoleh hasil perhitungan *Mean Square Error* (MSE) dibawah 0,1 (Sánchez Brea dkk., 2016).

Penelitian dengan menggunakan *hybrid feature selection* dan SVM juga pernah dilakukan dengan menggunakan objek penelitian beberapa jenis penyakit kanker. Jenis-jenis penyakit kanker yang digunakan dalam penelitian yaitu kanker paru, *Diffuse Large B-cell Lymphoma* (DLBCL), kanker paru, kanker usus besar, kanker prostat dan leukemia. Penelitian menggunakan basis *gene selection* sebagai metode penelitian. Dalam penelitian menyebutkan bahwa kombinasi antara IG dan SVM mampu menunjukkan nilai akurasi yang tinggi yaitu sebesar 90,32% pada kanker usus besar jika dibandingkan dengan algoritma serupa. Hasil dalam penelitian

diperoleh berdasarkan tiga *gene* diantaranya CSRP1, MYL9 dan GUCA2B (Gao dkk., 2017).

## 1.2. Dasar Teori

Dalam sub bab ini membahas mengenai dasar teori yang digunakan dalam penelitian. Adapun penjelasan dari masing-masing objek dan metode yang digunakan diantaranya:

### 1.2.1. Penyakit Ginjal Kronis

Penyakit ginjal kronis (*Chronic Kidney Disease*) adalah penyakit dengan adanya suatu kelainan yang dapat mempengaruhi fungsi ginjal yang mana pada tahap akhir dari penyakit ini dapat menyebabkan gagal ginjal (Almansour dkk., 2019). Penyakit ginjal kronis ini termasuk dalam kondisi yang berkaitan dengan peningkatan morbiditas dan mortalitas karena bisa memiliki keterkaitan dengan penyakit lain, termasuk penyakit kardiovaskular (Rady dan Anwar, 2019). Penyakit ginjal kronis termasuk penyakit yang umum, berbahaya namun juga bisa diobati (Arici, 2014).

Dalam pendefinisian dan pengklasifikasian penyakit ginjal kronis ini juga mengalami evolusi seiring dengan berjalannya waktu hingga saat ini. Untuk menetapkan kondisi pasien yang mengalami penyakit ginjal kronis sebagai penyakit dengan penurunan fungsi ginjal ini terdapat pedoman Internasional yang digunakan. Penyakit ginjal kronis diidentifikasi dengan tingkat *glomerulus* yang berjumlah kurang dari 60mL/menit per 1,75m<sup>2</sup>, atau juga adanya penanda rusaknya ginjal atau mengalami kedua kondisi tersebut setidaknya selama 3 bulan terlepas dari sumber yang menjadi penyebabnya (Webster dkk., 2017). Pengklasifikasian yang ada pada penyakit ginjal kronis ini dapat dilihat berdasarkan penyebab (*cause*; C), *Glomerular Filtration Rate* atau GFR (G; G1 hingga G5), dan *albuminuria* (A; A1 hingga A3) (Arici, 2014).

Penyakit ginjal kronis umumnya bisa dikatakan sebagai penyakit yang termasuk *silent condition* (Arici, 2014). Gejala yang termasuk khas akan muncul ketika penyakit ginjal kronis ini telah mencapai tingkatan (*stage*) tertentu. Ada tiga tes diagnostik yang dapat dilakukan untuk mendeteksi penyakit ginjal kronis yaitu tes

*dipstick urinalysis* untuk *albuminuria*, tes serum kreatinin untuk menghitung eGFR dan mengecek tekanan darah (Arici, 2014). Ketika seorang pasien mengalami penyakit ginjal kronis, gejala dan tanda-tanda yang dialami umumnya tidak spesifik. Tetapi ada beberapa tanda dan gejala yang bisa dicurigai sebagai gejala awal dari penyakit ginjal kronis, diantaranya (Arici, 2014):

1. Perubahan pada saat buang air kecil (*nocturia*, *polyuria*, dan frekuensinya)
2. Urin terdapat busa atau berbuih
3. Terdapat darah di dalam urin atau urin memiliki warna yang gelap
4. Tekanan darah tinggi
5. Bengkak (Edema)
6. Nyeri pinggang
7. Merasa lemas
8. Kulit pucat
9. Mual
10. Nafsu makan yang menurun
11. Munculnya rasa sakit (bisa disebabkan karena batu atau hal lainnya)

Selain gejala-gejala awal seperti yang telah disebutkan sebelumnya, penyakit ginjal kronis umum dialami pada (Arici, 2014):

1. Orang yang berusia lanjut (*elderly*)
2. Laki-laki
3. Orang dengan komplikasi sistemik, seperti anemia, hipertensi, gangguan mineral dan tulang.
4. Orang dengan obesitas
5. Orang yang memiliki keluarga dengan riwayat penyakit ginjal kronis

Penyakit ginjal kronis ini dapat memiliki potensi untuk menjadi penyakit yang progresif dengan adanya kemungkinan pasti hilangnya fungsi ginjal (Arici, 2014). Tingkat keparahan penyakit ginjal kronis dapat dipengaruhi dengan adanya penyakit lain seperti hipertensi, penyakit jantung, diabetes dan obesitas (Arici, 2014). Apabila penyakit ginjal kronis sudah semakin parah, maka bisa muncul masalah pada kekuatan

otot, gangguan tidur, gangguan pada paru, perubahan pada kulit, tingkat kesadaran dan lain sebagainya (Arici, 2014) .

### 1.2.2. Feature Selection

*Feature selection* atau yang juga biasa dikenal dengan *Variable Selection* merupakan suatu teknik *pre-processing* data yang sering digunakan dalam suatu penerapan *data mining* untuk mengurangi data dengan menghilangkan atribut atau fitur yang dianggap tidak penting dan berlebihan dari *dataset* yang digunakan (Jain dan Singh, 2018). Dalam penggunaannya, algoritma ini dianggap mampu mengoptimalkan suatu kinerja dari klasifikasi yang digunakan (Koutanaei dkk., 2015). Penerapan algoritma *feature selection* dalam suatu penelitian ini memiliki beberapa keuntungan, diantaranya (Koutanaei dkk., 2015):

1. Mengurangi *noise* yang ada di dalam *dataset*.
2. Mengurangi biaya komputasi agar dapat berhasil dalam menemukan model yang tepat.
3. Dapat membantu dalam pemahaman *final model* yang ada dalam algoritma klasifikasi.
4. Dapat membantu dalam memperbarui model.

Pendekatan dalam algoritma *feature selection* secara tradisional dan *modern* untuk *machine learning* terbagi menjadi empat kategori, diantaranya (Jain dan Singh, 2018):

1. *Filter method*.

Metode *filter* merupakan salah satu metode *feature selection* yang tertua. Dalam penggunaannya, variabel dipilih dengan menggunakan pendekatan *filter* yang mana penyaringan fitur ini dilakukan sebelum menerapkan algoritma *learning* jenis apapun. Metode ini akan melakukan *ranking* pada fitur yang digunakan berdasarkan kriteria evaluasi tertentu.

## 2. *Wrapper method*

Metode *wrapper* merupakan metode dalam *feature selection* yang melakukan pemilihan fitur terbaik atau yang dianggap paling “berguna” dengan mempertimbangkan algoritma *learning* yang digunakan dalam penelitian. Metode ini dikatakan mampu memberikan hasil yang lebih akurat jika dibandingkan dengan metode *filter*.

## 3. *Embedded method*

Metode *embedded* merupakan metode yang dalam pencarian fiturnya biasanya masih dipandu oleh proses *learning method*-nya. Metode ini mengukur “kegunaan” *subset* fitur yang ada dan sekaligus melakukan pemilihan fitur. Secara komputasi metode ini lebih baik dari metode *wrapper*.

## 4. *Hybrid method*

Metode *hybrid* merupakan metode yang menggabungkan satu atau lebih metode pendekatan *feature selection* untuk memperoleh manfaat dari kelebihan metode lainnya. Metode ini biasanya terdiri dari gabungan antara metode *filter* dan metode *wrapper*. Jenis metode ini biasanya mampu mencapai hasil akurasi yang lebih tinggi jika dibandingkan dengan jenis metode yang lainnya.

Sementara metode *feature selection* yang belakangan ini sering digunakan dalam penelitian adalah *feature selection* jenis *hybrid*. Metode *hybrid* ini menggabungkan satu metode atau lebih untuk mendapatkan hasil yang optimal. Hasil akurasi yang dicapai biasanya lebih tinggi dibanding dengan metode lainnya dengan mengurangi jumlah fitur dari *dataset* (Jain dan Singh, 2018).

### 1.2.3. *Information Gain*

Algoritma *Information gain* (IG) didefinisikan sebagai sebuah algoritma yang digunakan sebagai ukuran keefektifitasan suatu fitur dalam mengklasifikasikan data pada penelitian (Jadhav dkk., 2018). Konsep *information gain* ini diperoleh berdasarkan pada konsep yang dimiliki oleh *information entropy* (Koutanaei dkk.,

2015). Entropi (*entropy*) merupakan salah satu contoh dari beberapa cara yang bisa dilakukan untuk mengukur suatu keragaman (Jadhav dkk., 2018).

Persamaan yang dapat digunakan untuk menghitung entropi dapat dijabarkan sebagai berikut (Suyanto, 2018):

$$Entropy(S) = \sum_i^c -p_i \log_2 p_i \quad (2.1)$$

dengan:

$c$  = jumlah nilai pada atribut target (jumlah kelas).

$p_i$  = rasio antara jumlah sampel kelas  $i$  dengan jumlah semua sampel himpunan data.

Persamaan untuk menghitung *information gain* dari suatu atribut dalam penelitian dapat dijabarkan sebagai berikut (Suyanto, 2018):

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \quad (2.2)$$

dengan:

$A$  = atribut.

$v$  = suatu nilai yang mungkin untuk atribut  $A$ .

$Values(A)$  = himpunan nilai yang mungkin untuk atribut  $A$ .

$|S_v|$  = jumlah sampel untuk nilai  $v$ .

$|S|$  = jumlah seluruh sampel nilai data.

$Entropy(S_v)$  = *entropy* untuk sampel-sampel yang memiliki nilai  $v$

#### 1.2.4. Sequential Feature Selection

*Sequential Feature selection* adalah metode dalam *feature selection* yang mampu mengevaluasi fitur yang tidak dipilih dengan mengacu pada fitur yang terpilih dari data yang digunakan. Dengan menggunakan metode ini, fitur yang berlebihan dapat ditangani secara lebih efektif (Gu dkk., 2015). Ada beberapa jenis strategi penggunaan *Sequential Feature selection*, diantaranya (Gu dkk., 2015):

1. *Sequential Forward Feature selection* (SFS)



Metode SFS digunakan untuk memilih fitur yang digunakan berdasarkan strategi pencarian maju dan berurutan (*sequential forward search strategy*). Metode ini terbukti sebagai metode yang efektif dalam memilih fitur dengan tetap mempertimbangkan informasi timbal baliknya (*mutual information*). Kelemahan utama SFS adalah tidak dapat menghapus fitur yang dianggap menjadi usang setelah adanya penambahan fitur lainnya.

### 2. *Sequential backward feature selection* (SBS)

Metode SBS adalah metode yang memiliki cara kerja yang berlawanan dengan metode SFS. Kelemahan metode ini terletak pada tidak mempunya metode untuk mengevaluasi ulang kegunaan fitur setelah fitur tersebut dibuang.

### 3. *Plus-L Minus-R Selection* (LRS)

Metode LRS adalah metode yang dianggap sebagai integrasi antara metode SFS dan metode SBS. Dalam metode ini, L dan R dinotasikan dalam bilangan bulat. Metode LRS ini mengupayakan kompensasi kelemahan yang dimiliki oleh metode SFS dan metode SBS karena memiliki kemampuan untuk *backtracking*.

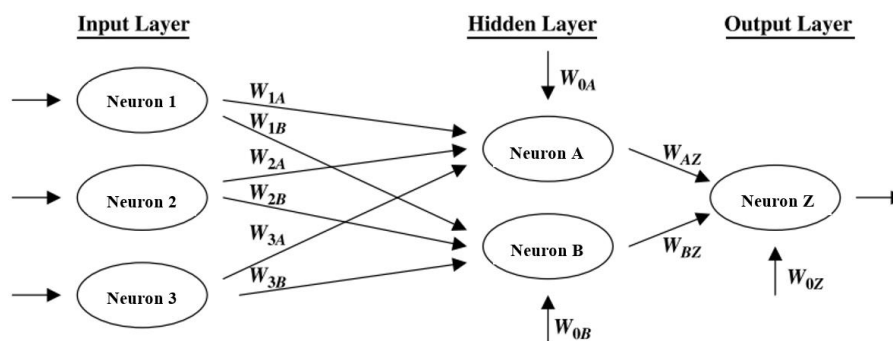
Dalam penelitian ini menggunakan SBS untuk memilih fitur yang dianggap terbaik diantara semua fitur yang digunakan sebagai *input*. Metode SBS dipilih karena metode ini termasuk dalam *greedy wrapper method*, yang mana memiliki kemampuan untuk memilih jalur pencarian dengan berdasarkan pada arah yang menunjukkan nilai terbaik (Kuhn dan Johnson, 2019). Untuk jenis validasi yang digunakan, dalam metode SBS menggunakan *cross-validation* (Bolón-Canedo dkk., 2015).

## 1.2.5. Jaringan Syaraf Tiruan

Jaringan Syaraf Tiruan (JST) atau yang juga dikenal dengan *Artificial Neural network* (ANN) merupakan metode yang merepresentasikan tingkat yang sangat mendasar untuk meniru jenis pembelajaran *non-linear* yang terjadi di dalam jaringan *neuron* pada otak (Larose, 2005). *Neuron* adalah ratusan atau bahkan miliaran sel saraf yang ada di dalam otak manusia (Almansour dkk., 2019). Meskipun dalam suatu *neuron* tertentu terlihat sederhana jika dalam struktur, tetapi *neuron* ini memiliki tugas-

tugas untuk mempelajari sesuatu yang kompleks seperti mengenali pola atau melakukan klasifikasi (Larose, 2005).

Jika membandingkan kinerja yang rumit seperti melakukan pengenalan dan prediksi, otak manusia bekerja lebih baik dibandingkan dengan komputer digital karena otak manusia memiliki kemampuan untuk mempelajari suatu hal (Almansour dkk., 2019). Metode JST ini diambil berdasarkan dengan bagaimana kerja otak dapat membentuk suatu koneksi yang akurat (Almansour dkk., 2019). Keuntungan penggunaan JST ini adalah jaringan dapat belajar untuk mengatasi suatu contoh yang tidak informatif dari dalam kumpulan data yang ada dengan membutuhkan waktu pelatihan yang lebih lama jika dibandingkan dengan pohon keputusan (Larose, 2005).



Gambar 2. 1 JST sederhana (Larose, 2005)

Metode JST terdiri dari setidaknya tiga lapisan (*layer*): *input layer*, *hidden layer* dan *output layer* (Larose, 2005). Dalam penggunaan metode JST, ada dua jenis algoritma utama yang biasa digunakan yaitu Algoritma *Feed-Forward* dan Algoritma *Backpropagation* (Olawoyin, 2016).

### 1.2.6. Algoritma *Backpropagation*

Algoritma *Backpropagation* adalah salah satu algoritma yang umum digunakan dalam penerapan metode JST (Olawoyin, 2016). *Backpropagation* merupakan algoritma pembelajaran yang terawasi yang mana juga memetakan *set input* menuju ke *output* yang benar. Algoritma ini terdiri dari *input layer*, *implicit (hidden) layer*, dan juga *output layer* (Ye dkk., 2019).

Dalam penggunaannya, algoritma *backpropagation* melibatkan tiga tahap, diantaranya (Fausett dan Fausett, 1994):

1. *Feedforward*

Selama tahapan *feedforward*, setiap *input* yang digunakan akan menerima sebuah sinyal *input* yang mana sinyal tersebut akan disebarkan ke masing-masing *hidden unit*. Setiap *hidden unit* ini akan menghitung aktivasi yang kemudian akan mengirimkan sinyal ke masing-masing *output unit*. Persamaan yang digunakan dalam tahap *feedforward* ini adalah sebagai berikut (Fausett dan Fausett, 1994):

$$z\_in_j = v_{0j} + \sum_i x_i v_{ij} \quad (2.3)$$

dengan:

$i$  = neuron ke- $i$  (misalnya  $i=1,2,3,4, \dots, n$ ) pada lapisan *input*.

$j$  = neuron ke- $j$  (misalnya  $j=1,2,3,4, \dots, p$ ) pada *hidden layer*.

$v_{0j}$  = bias pada lapisan *input*.

$x_i$  = bobot pada  $i$ .

$v_{ij}$  = nilai *input* pada  $i$  ke  $j$ .

Selanjutnya menerapkan fungsi aktivasi dengan menggunakan persamaan sebagai berikut:

$$z_j = f(z\_in_j) = \frac{1}{1 + \exp^{-z\_in_j}} \quad (2.4)$$

dengan:

$f(z\_in_j)$  = nilai aktivasi pada *hidden layer* ke *output*.

$exp$  = bilangan natural = 2,718281828.

Untuk menghitung pada proses *input* yang lain, dapat menggunakan persamaan sebagai berikut:

$$y\_in_k = w_{0k} + \sum_j z_j w_{jk} \quad (2.5)$$

dengan:

$w_{0k}$  = bias pada *hidden layer*.

$z_j$  = hasil nilai fungsi aktivasi dari *hidden layer*.

$j$  = *neuron* ke- $j$  (misalnya  $j=1,2,3,4, \dots, m$ ).

$w_{jk}$  = bobot pada  $j$  ke  $k$  pada *output*.

Selanjutnya kembali menerapkan fungsi aktivasi dengan menggunakan persamaan sebagai berikut:

$$y_k = f(y_{in_k}) = \frac{1}{1 + \exp^{-y_{in_k}}} \quad (2.6)$$

dengan:

$f(y_{in_k})$  = nilai aktivasi pada *hidden layer* ke *output*.

$\exp$  = bilangan natural = 2,718281828.

## 2. Backpropagation

Persamaan yang digunakan dalam tahap *backpropagation* ini adalah sebagai berikut (Fausett dan Fausett, 1994):

$$\delta_k = (t_k - y_k) f'(y_{in_k}) \quad (2.7)$$

dengan:

$\delta_k$  = informasi kesalahan.

$y_k$  = unit *output*.

$t_k$  = pola target.

Selanjutnya menghitung koreksi bobot dengan menggunakan persamaan sebagai berikut:

$$\Delta w_{jk} = \alpha \delta_k z_j \quad (2.8)$$

dengan:

$\Delta w_{jk}$  = besar perubahan untuk ditambahkan pada bobot lama yang diperbarui.

$\alpha$  = laju pembelajaran.

$\delta_k$  = nilai galat antara *input layer* dan *hidden layer*.

$z_j$  = nilai *input* dari *neuron*  $i$  ke *neuron*  $j$ .

Kemudian menghitung koreksi bias dengan menggunakan persamaan sebagai berikut:

$$\Delta w_{0k} = \alpha \delta_k \quad (2.9)$$

dengan:

$\Delta w_{0k}$  = besar perubahan untuk ditambahkan pada bobot lama yang diperbarui.

$\alpha$  = laju pembelajaran.

$\delta_k$  = nilai galat antara *input layer* dan *hidden layer*.

Selanjutnya menghitung *delta input* dengan menggunakan persamaan sebagai berikut:

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk} \quad (2.10)$$

dengan:

$\delta_k$  = informasi kesalahan.

$w_{jk}$  = bobot pada j ke k pada *output*.

Hasil yang diperoleh dari *delta input* dikalikan dengan turunan fungsi aktivasi yang telah dihitung sebelumnya dengan menggunakan persamaan sebagai berikut:

$$\delta_j = \delta_{in_j} f'(z_{in_j}) \quad (2.11)$$

dengan:

$\delta_j$  = informasi kesalahan.

### 3. Penyesuaian Bobot Sekolah Pascasarjana

Persamaan yang digunakan untuk menghitung penyesuaian bobot dalam algoritma *backpropagation* adalah sebagai berikut (Fausett dan Fausett, 1994):

$$w_{jk}(new) = w_{jk}(old) + \Delta w_{jk} \quad (2.12)$$

$$v_{ij}(new) = v_{ij}(old) + \Delta v_{ij} \quad (2.13)$$

dengan:

$w_{jk}(new)$  = bobot baru antara *input layer* dan *hidden layer* yang akan dicari.

$w_{jk}(old)$  = bobot lama yang diperbaharui.

$v_{ij}(new)$  = bobot baru antara *hidden layer* dan *output layer* yang akan dicari.

$v_{ij}(old)$  = bobot lama yang diperbaharui.

Pemilihan metode *Backpropagation* yang digunakan dalam penelitian berdasarkan pada kemampuan metode untuk meminimalkan kesalahan kuadrat total (*total squared error*) dari komputasi *output* dari *net* (Fausett dan Fausett, 1994).

### 1.2.7. Parameter Evaluasi

Klasifikasi yang digunakan dalam parameter evaluasi:

#### 1. *Confusion matrix*

*Confusion matrix* merupakan sebuah matriks yang digunakan dalam penelitian untuk merangkum performa klasifikasi dengan menggunakan beberapa data uji. Klasifikasi *confusion matrix* terlihat pada Tabel 2.1 (Shultz dan Fahlman, 2017):

Tabel 2. 1 *Confusion matrix*

<i>Actual Class</i>	<i>Assigned Class</i>	
	<i>Positive</i>	<i>Negative</i>
<i>Positive</i>	TP	FN
<i>Negative</i>	FP	TN

dengan:

TN = *True Negatives*.

FN = *False Negatives*.

TP = *True Positives*.

FP = *False Positives*.

#### 2. *Sensitivity (True Positive atau Recall)*

*Sensitivity* atau *Recall* digunakan untuk menghitung besarnya identifikasi yang positif benar. Persamaan yang digunakan untuk perhitungan *recall* (Shultz dan Fahlman, 2017):

$$Sensitivity = \frac{TP}{(TP+FN)} \quad (2.14)$$

dengan:

TP = *True Positives*.

FN = *False Negatives*.

### 3. *Specificity (True Negative Rate)*

*Specificity* atau *True Negative Rate* digunakan untuk menghitung besarnya identifikasi yang negatif benar. Persamaan yang digunakan untuk perhitungan *specificity* (Shultz dan Fahlman, 2017):

$$\text{Specificity} = \frac{TN}{(TN+FP)} \quad (2.15)$$

dengan:

TN = *True Negatives*.

FP = *False Positives*.

### 4. *Precision (Positive Predictive Value)*

*Precision* atau *Positive Predictive Value* digunakan untuk menghitung suatu contoh yang sesuai diantara contoh yang digunakan. Persamaan yang digunakan untuk perhitungan *precision* (Shultz dan Fahlman, 2017):

$$\text{Positive Predictive Value} = \frac{TP}{(TP+FP)} \quad (2.16)$$

dengan:

TP = *True Positives*.

FP = *False Positives*.

### 5. *Negative Predictive Value*

*Negative Predictive Value* digunakan untuk menghitung probabilitas sesuatu hasil yang negatif. Persamaan yang digunakan untuk perhitungan dari *negative predictive value* (Shultz dan Fahlman, 2017):

$$\text{Negative Predictive Value} = \frac{TN}{(TN+FN)} \quad (2.17)$$

dengan:

TN = *True Negatives*.

FN = *False Negatives*.

#### 6. Accuracy

*Accuracy* atau akurasi digunakan untuk menghitung ketepatan. Persamaan yang digunakan untuk menghitung akurasi (Koutanaei dkk., 2015):

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (2.18)$$

dengan:

TN = *True Negatives*.

FN = *False Negatives*.

TP = *True Positives*.

FP = *False Positives*.



Sekolah Pascasarjana