

## BAB II. LANDASAN TEORI

### 2.1. Studi Literatur

Pada tahun 2018, Dania Eridani, Muhammad Aditya Miftahul Rifki, dan R. Rizal Isnanto melakukan penelitian menggunakan metode *forward chaining* dalam mendiagnosa jenis gangguan kecemasan. Metode *forward chaining* pada penelitian tersebut digunakan untuk melacak jenis gangguan kecemasan berdasarkan *rule* yang telah ditentukan pada input gejala gangguan kecemasan. Penelitian tersebut menghasilkan akurasi 100% dalam mendiagnosa jenis gangguan kecemasan (Eridani, Rifki, & Isnanto, 2018). Penelitian ini menggunakan data 7 jenis gangguan kecemasan. Pasien akan menjawab 23 pertanyaan yang berhubungan dengan gejala gangguan kecemasan. Pilihan jawaban pasien terdiri dari ya atau tidak, sehingga data gejala yang diinputkan bersifat biner.

Reski Mai Candra dan Bambang Mirwanto melakukan penelitian menggunakan Teorema Bayes untuk mendiagnosa jenis gangguan kecemasan. Penelitian tersebut menghasilkan akurasi sebesar 100% (Candra & Mirwanto, 2018). Penelitian ini menggunakan data 5 jenis gangguan kecemasan. Pasien penderita gangguan kecemasan akan menjawab 6 pertanyaan yang berhubungan dengan gejala-gejala gangguan kecemasan yang dialami untuk mendiagnosa jenis gangguan kecemasan yang diderita oleh pasien. Untuk menjawab keenam pertanyaan tersebut, terdapat lebih dari 2 pilihan jawaban yang disediakan. Jawaban dari pasien kemudian diubah ke skala angka untuk dijadikan inputan dalam mendiagnosa jenis gangguan kecemasan.

Teguh Suprayitno melakukan penelitian tentang diagnosa jenis gangguan kecemasan menggunakan metode jaringan saraf tiruan (JST) *Learning Vector Quantization 2.1* (LVQ2.1). Penggunaan metode LVQ2.1 memberikan hasil akurasi sebesar 100% pada *learning rate* sebesar 0.15 dan nilai *window* sebesar 0.1 (Suprayitno, 2018). Penelitian ini menggunakan 5 kelas gangguan kecemasan dengan input 30 gejala gangguan kecemasan. Pasien menjawab 30 pertanyaan yang berhubungan dengan gejala gangguan kecemasan yang dialami pasien. Pilihan jawaban pasien terdiri dari ya atau tidak. Data pasien gangguan kecemasan yang

digunakan pada penelitian ini sejumlah 200 data. Masing-masing kelas gangguan kecemasan terdiri dari 40 data pasien.

Penelitian yang dilakukan oleh Supriadi pada 2019 menghasilkan kesimpulan bahwa metode *Learning Vector Quantization* 3 dapat diterapkan untuk klasifikasi jenis gangguan kecemasan. Penelitian tersebut menghasilkan akurasi terbaik sebesar 100% pada *learning rate* sebesar 0.15 dan 0.2 serta parameter *window* sebesar 0.1 dan 0.3 (Supriadi, 2019). Penelitian ini menggunakan 6 kelas gangguan kecemasan. Pasien menjawab 32 pertanyaan yang berhubungan dengan gejala gangguan kecemasan yang dialami. Pilihan jawaban yang disediakan terdiri dari ya atau tidak, sehingga data yang diinputkan bersifat biner. Jumlah data pasien gangguan kecemasan yang digunakan pada penelitian ini sejumlah 240 data. Tiap kelas gangguan kecemasan terdiri dari 40 data.

Pada tahun 2019, Zubaidah Al Ubaidah Sakti melakukan penelitian tentang identifikasi jenis gangguan kecemasan menggunakan metode *Modified K-Nearest Neighbor* (MKNN). Metode MKNN menghasilkan akurasi tertinggi sebesar 95% dengan nilai *K optimum* sebesar 2 pada *dataset* tidak seimbang dalam mengidentifikasi jenis gangguan kecemasan yang diderita pasien (Sakti, 2019). Jumlah data yang digunakan pada penelitian ini terdiri dari 120 data. Adapun *output* dari penelitian ini terdiri dari 6 kelas gangguan kecemasan. Identifikasi jenis gangguan kecemasan dilakukan berdasarkan hasil kuesioner *Hamilton Anxiety Rating Scale* (HARS) yang terdiri dari 14 pertanyaan (Sakti, 2019). Pasien akan menjawab tiap pertanyaan dalam bentuk skala angka yang dimulai dari 0 (tidak ada gejala) sampai 4 (gejala sangat berat). Jawaban dari tiap pertanyaan tersebut kemudian akan dijadikan input dalam menentukan jenis gangguan kecemasan yang diderita pasien.

Irin Saputra melakukan penelitian tentang diagnosa jenis gangguan kecemasan menggunakan metode *backpropagation momentum* dengan kombinasi biner. Penelitian tersebut menghasilkan sebuah aplikasi yang dapat digunakan untuk mendiagnosa jenis gangguan kecemasan yang diderita oleh pasien. Pada aplikasi tersebut, terdapat sejumlah 30 pertanyaan yang berhubungan dengan gejala gangguan kecemasan yang dialami pasien. Pasien akan menjawab pertanyaan-pertanyaan yang terdapat pada aplikasi tersebut. Pilihan jawaban yang disediakan pada tiap pertanyaan terdiri dari ya atau tidak. Jawaban dari pasien selanjutnya akan

dijadikan input untuk menentukan jenis gangguan kecemasan yang diderita pasien. *Output* yang dihasilkan terdiri dari 5 kelas gangguan kecemasan. Penelitian ini menggunakan 200 data pasien pengidap gangguan kecemasan. Akurasi yang dihasilkan dari penelitian ini adalah 100% pada nilai *learning rate* sebesar 0.2 dan *momentum* sebesar 0.5 dan 0.8 (Saputra, 2019).

Pada tahun 2018, Ade Pujiyanto, Kusriani, dan Andi Sunyoto melakukan penelitian tentang sistem pendukung keputusan prediksi penerima beasiswa di Universitas Amikom Yogyakarta menggunakan metode *backpropagation* dengan variasi *momentum*. Penelitian tersebut menghasilkan akurasi sebesar 90% dengan nilai *error* terkecil sebesar 0.000101 pada nilai *epoch* ke-239 serta konfigurasi *learning rate* sebesar 0.2 dan nilai *momentum* sebesar 0.2 (Pujiyanto, Kusriani, & Sunyoto, 2018). Variabel yang digunakan sebagai inputan pada penelitian ini yaitu Indeks Prestasi Kumulatif (IPK), jumlah sistem kredit semester (SKS), status, dan pendapatan orang tua. *Dataset* yang digunakan berupa *dataset* penerima beasiswa yang diperoleh dari tahun 2012 sampai tahun 2017 sejumlah 3000 data.

Penelitian yang dilakukan oleh Budy Satria pada tahun 2018 menggunakan metode *backpropagation* untuk memprediksi volume penggunaan air di PT. PDAM Kota Duri. Penelitian ini menghasilkan akurasi sebesar 99.999% (Satria, 2018). Berdasarkan hasil penelitian tersebut, dapat disimpulkan bahwa metode *backpropagation* dapat digunakan untuk memprediksi konsumsi air di PT. PDAM Kota Duri pada tahun berikutnya. Data yang digunakan pada penelitian ini diambil dari data total konsumsi air per bulan oleh pelanggan PDAM pada tahun 2016 dan 2017. Variabel yang digunakan pada penelitian ini antara lain adalah sosial umum, sosial khusus, rumah tangga 1, rumah tangga 2, rumah tangga 3, niaga 1, niaga 2, niaga 3, dan jumlah penggunaan air.

Penelitian yang dilakukan oleh Hasdi Putra dan Nabilah Ulfa Walmi pada tahun 2020 menggunakan metode *backpropagation* dengan variasi *momentum* menghasilkan akurasi terbaik sebesar 88.14% dalam prediksi produksi padi (Putra & Walmi, 2020). Nilai akurasi terbaik diperoleh pada konfigurasi *epoch* sebesar 200, *momentum* sebesar 0.5, dan *learning rate* sebesar 0.5. Data yang digunakan pada penelitian ini merupakan data produksi padi selama 5 tahun dari tahun 2013 sampai 2017 yang berasal dari 19 kota/kabupaten di Sumatera Barat. Variabel yang dijadikan inputan dalam penelitian ini adalah data target luas tanam padi, data target

luas panen padi, data target produktivitas padi, dan data target produksi padi. *Output* dari penelitian ini adalah hasil prediksi produksi padi tahun 2018.

Teguh Wibawa dan Imam Suharjo melakukan penelitian pada tahun 2021 tentang identifikasi gejala penyakit hipertensi berdasarkan citra iris mata menggunakan jaringan saraf tiruan dan metode *backpropagation* dengan variasi *momentum*. Nilai akurasi terbaik yang dihasilkan pada penelitian ini adalah 87.5% pada konfigurasi *learning rate* sebesar 0.1 dan *epoch* ke-100 (Wibawa & Suharjo, 2021). Data yang digunakan dalam penelitian ini berupa data 196 citra mata. Pada proses pengambilan data, data citra mata hipertensi dan citra mata normal telah dipisahkan untuk selanjutnya melalui proses *pre-processing*. Proses *pre-processing* dilakukan untuk mengekstraksi fitur yang ada pada citra mata. Hasil ekstraksi fitur dari citra mata tersebut antara lain *angular second moment*, *entropy*, *correlation*, *contrast*, dan *inverse different moment*. Selanjutnya hasil ekstraksi fitur akan diproses pada jaringan saraf tiruan untuk mengidentifikasi gejala penyakit hipertensi yang diderita.

## 2.2. Dasar Teori

### 2.2.1. Sistem Pakar

Pada dasarnya, sistem pakar dirancang untuk menyelesaikan suatu permasalahan dengan meniru kinerja para pakar atau ahli. Sistem pakar adalah program komputer yang mempunyai pengetahuan yang berasal dari manusia yang berpengetahuan luas dalam *domain* tertentu, di mana pengetahuan di sini adalah pengetahuan manusia yang sangat minim penyebarannya, mahal, serta susah didapat (Siswanto, 2010). Pengetahuan manusia akan diproses di dalam komputer. Hasil pemrosesan pengetahuan manusia tersebut selanjutnya dapat dimanfaatkan dalam pemecahan suatu masalah.

Sistem pakar dapat menyelesaikan masalah dalam *domain* yang terbatas berdasarkan pengetahuan yang diinputkan pada sistem pakar tersebut. Pengetahuan pada sistem pakar diperoleh dari manusia atau pakar yang memiliki keahlian di bidang tertentu, sehingga sistem pakar hanya dapat menyelesaikan masalah yang dapat diselesaikan oleh manusia. Dengan demikian, tingkat kehandalan sistem pakar tergantung pada pengetahuan pakar yang dimasukkan ke dalamnya.

Akan tetapi, sistem pakar terkadang memiliki kinerja yang lebih baik daripada pakar. Dengan adanya sistem pakar, orang awam pun dapat menyelesaikan masalah rumit yang umumnya hanya dapat diselesaikan oleh para pakar. Sistem pakar juga dapat dijadikan sebagai asisten untuk membantu aktivitas para pakar dalam penyelesaian suatu masalah.

### 2.2.2. Gangguan Kecemasan

Gangguan kecemasan dapat diartikan sebagai keadaan mental yang ditandai dengan rasa khawatir yang berlebihan dan tidak dapat dihindari oleh seseorang. Gangguan kecemasan juga merupakan gangguan fungsi yang disebabkan oleh perasaan takut dan khawatir yang tidak dapat dibenarkan, yang biasanya disertai gejala fisiologis, seperti jantung berdebar, napas pendek atau nyeri dada. Perasaan tersebut umumnya ditimbulkan dari kecemasan berlebihan yang tidak dapat dikontrol, sehingga dapat menimbulkan perasaan tidak nyaman bagi penderitanya.

Gangguan kecemasan dapat ditandai dengan ketegangan motorik berupa hiperaktivitas, tubuh yang bergetar, duduk yang tidak tenang, dan tubuh yang tidak santai. Berbeda dengan kecemasan yang biasa yang dialami, gangguan kecemasan tidak bisa dikendalikan oleh penderitanya. Gangguan kecemasan merupakan salah satu jenis gangguan psikologi yang paling banyak ditemukan di dunia. Di Indonesia, gangguan mental ini diderita oleh sekitar 9%-12% populasi umum (Dewi & Fauziah, 2018). Selain itu, gangguan kecemasan juga merupakan gangguan diagnosis klinis yang paling umum dialami oleh remaja. Berdasarkan hasil Riset Kesehatan Dasar tahun 2013, sekitar 6% remaja di Indonesia atau 14 juta penduduk Indonesia mengalami gangguan mental emosional yang ditunjukkan dengan gejala-gejala kecemasan dan depresi (Dewi & Fauziah, 2018). Gangguan kecemasan terbagi menjadi 6 kategori, di antaranya adalah sebagai berikut (Sakti, 2019):

#### 1. *General Anxiety Disorder*

*General anxiety disorder* atau gangguan kecemasan menyeluruh adalah gangguan kecemasan yang menetap dan tidak dapat terkontrol (Davison & Neale, 2001). Gangguan ini dapat ditandai dengan kekhawatiran berlebihan yang bersifat pervasif dan disertai oleh berbagai simtom somatik, sehingga dapat menyebabkan gangguan signifikan dan menimbulkan stres pada

penderita. Seseorang yang menderita jenis gangguan kecemasan ini biasanya sulit berkonsentrasi dan tidak bisa merasa santai.

## 2. *Panic Disorder*

*Panic disorder* atau gangguan panik adalah serangan panik secara mendadak yang umumnya ditandai dengan denyut jantung yang semakin cepat, pusing, dan gemeteran (Davison & Neale, 2001). Gangguan panik dapat terjadi secara intens dan tidak menentu sehingga memicu reaksi fisik walau tanpa penyebab yang jelas. Penderita gangguan panik umumnya akan mengalami gejala jantung berdebar atau nyeri dada saat serangan panik tersebut muncul.

## 3. *Social Anxiety Disorder*

*Social anxiety disorder* atau gangguan kecemasan sosial adalah perasaan cemas atau takut yang berlebihan pada situasi sosial (Abbasi, Ariapooran, & Karimi, 2019). Jenis gangguan kecemasan ini merupakan penyakit kejiwaan umum yang ditandai dengan ketakutan yang berlebihan dan/atau menghindari situasi sosial di mana seseorang merasa terlalu diperhatikan oleh orang lain dan merasa takut akan penilaian negatif dari orang lain.

## 4. *Specific Phobia*

*Specific phobia* atau fobia spesifik merupakan kecemasan berlebihan, tanpa alasan yang jelas dan menetap pada diri individu terhadap hadirnya suatu objek atau situasi yang spesifik, misalnya takut terhadap air, takut darah dan takut binatang dan lain-lain (Purwaningtyas, 2020). Fobia spesifik yang paling umum dialami contohnya adalah fobia terhadap ular sebagai hewan yang dapat berpotensi mengancam, fobia terhadap kegelapan, fobia dengan ketinggian, fobia dengan suara kilatan petir, dan lain-lain.

## 5. *Obsessive Compulsive Disorder*

*Obsessive compulsive disorder* atau gangguan obsesif kompulsif adalah gangguan pemikiran obsesif yang tidak dapat dikontrol dan perilaku atau tindakan mental yang berulang (Davison & Neale, 2001). Gangguan ini menyebabkan penderitanya merasa harus melakukan suatu tindakan secara berulang-ulang karena obsesi berlebihan yang umumnya didasari oleh perasaan cemas dan takut.

## 6. *Post Traumatic Stress Disorder*

*Post traumatic stress disorder* atau gangguan stres pasca trauma adalah gangguan jiwa yang menyebabkan penderitanya mengalami kecemasan berlebihan, stres akut, maupun trauma berkepanjangan akibat pengalaman traumatis yang dialami di masa lalu (Davison & Neale, 2001). Gangguan ini merupakan efek dari trauma psikologis. Beberapa contoh pengalaman traumatis yang dapat memicu gangguan ini antara lain perang, kecelakaan, bencana alam, dan pelecehan seksual.

### 2.2.3. Kuesioner *Hamilton Anxiety Rating Scale* (HARS)

Kuesioner *Hamilton Anxiety Rating Scale* (HARS) adalah suatu kuesioner yang dapat digunakan untuk mengukur tingkat kecemasan yang diderita oleh seseorang. Kuesioner HARS pertama kali dikembangkan pada tahun 1959 oleh Max Hamilton. Kuesioner ini mengukur tingkat kecemasan yang diderita berdasarkan gejala yang dialami. Pada kuesioner HARS, terdapat sejumlah pertanyaan yang mewakili tiap gejala kecemasan untuk mendiagnosa tingkat kecemasan yang diderita. Pasien penderita gangguan kecemasan dapat menjawab tiap pertanyaan pada kuesioner tersebut menggunakan skala *likert* yang terdiri dari 5 tingkatan nilai. Tingkatan nilai pada skala *likert* tersebut dimulai dari 0 hingga 4. Nilai 0 berarti tidak ada gejala, nilai 1 berarti gejala ringan, nilai 2 berarti gejala sedang, nilai 3 berarti gejala berat, dan nilai 4 berarti gejala sangat berat. Skor jawaban yang diberikan oleh pasien pada tiap pertanyaan kuesioner HARS selanjutnya akan dijumlahkan untuk mengetahui tingkat kecemasan yang diderita. Bila total skor pasien kurang dari sama dengan 6, maka pasien dinyatakan tidak mengidap kecemasan. Bila total skor berada di kisaran 7-14, maka pasien menderita kecemasan ringan. Bila *range* total skor berada di kisaran 15-27, maka pasien menderita kecemasan sedang. Bila total skor lebih dari 27, maka pasien menderita kecemasan berat (Siswanto M. , 2015)

Secara umum, kuesioner HARS digunakan untuk mengukur sejauh mana tingkat keparahan dari kecemasan yang diderita oleh seseorang. Namun kuesioner HARS juga dapat digunakan untuk mendiagnosa jenis gangguan kecemasan yang diderita pasien penderita gangguan kecemasan meski dalam penerapannya masih cukup jarang dilakukan. Penentuan jenis gangguan kecemasan menggunakan

kuesioner HARS memerlukan keterlibatan pakar di bidang psikologi. Adapun jumlah pertanyaan yang terdapat pada kuesioner HARS terdiri dari 14 pertanyaan yang mewakili 14 kelompok gejala kecemasan. 14 kelompok gejala tersebut antara lain adalah (Sakti, 2019):

1. Perasaan cemas: cemas, firasat buruk, takut akan pikiran sendiri dan mudah tersinggung.
2. Ketegangan: merasa tegang, lesu, tidak dapat beristirahat dengan tenang, mudah terkejut, mudah menangis, gemetar dan gelisah.
3. Ketakutan: pada kegelapan, pada orang asing, takut ditinggal sendiri, takut pada binatang besar, takut keramaian lalu lintas, dan takut pada kerumunan orang banyak.
4. Gangguan tidur: sukar tidur, terbangun pada malam hari, tidur tidak nyenyak, bangun dengan lesu, banyak mimpi, mimpi buruk, dan mimpi yang menakutkan.
5. Gangguan kecerdasan: sukar berkonsentrasi, daya ingat menurun, dan daya ingat buruk.
6. Perasaan depresi (murung): hilangnya minat, berkurangnya kesenangan pada hobi, merasa sedih, terbangun saat dini hari, dan perasaan berubah-ubah sepanjang hari.
7. Gejala somatik/fisik (otot): sakit dan nyeri di otot, kaku, kedutan otot, gigi gemerutuk, dan suara tidak stabil.
8. Gejala somatik/fisik (sensorik): telinga berdenging (tinnitus), penglihatan kabur, muka merah atau pucat, merasa lemas, dan perasaan ditusuk-tusuk.
9. Gejala kardiovaskuler (jantung dan pembuluh darah): denyut jantung cepat (takikardi), berdebar-debar, nyeri di dada, denyut nadi mengeras, rasa lesu/lemas seperti mau pingsan, dan detak jantung menghilang/berhenti sekejap.
10. Gejala respiratori (pernafasan): rasa tertekan atau sempit di dada, rasa tercekik, dan sering menarik nafas pendek/sesak.
11. Gejala gastrointestinal (pencernaan): sulit menelan, perut melilit, gangguan pencernaan, nyeri sebelum dan sesudah makan, perasaan terbakar di perut,

rasa penuh/kembung, mual, muntah, buang air besar dengan konsistensinya lembek, konstipasi, dan kehilangan berat badan.

12. Gejala urogenital (perkemihan dan kelamin): sering buang air kecil, tidak dapat menahan buang air kecil, tidak dapat menstruasi, darah menstruasi berlebihan, darah menstruasi sangat sedikit, masa menstruasi berkepanjangan, masa menstruasi sangat pendek, menstruasi beberapa kali dalam sebulan, ejakulasi dini, ereksi melemah, ereksi hilang, dan impotensi.
13. Gejala autoimun: mulut kering, muka merah, mudah berkering, kepala terasa berat, kepala terasa sakit, dan bulu-bulu kuduk berdiri.
14. Tingkah laku/sikap: gelisah, tidak tenang, jari gemetar, dahi berkerut, wajah tegang, nafas pendek dan cepat serta wajah merah.

#### 2.2.4. Metode *Backpropagation*

*Backpropagation* adalah salah satu algoritma yang digunakan oleh *perceptron* dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan *neuron* pada *hidden layer* atau lapisan tersembunyi. Algoritma ini melatih jaringan untuk mengenali pola masukan yang digunakan selama pelatihan sehingga jaringan dapat memberikan respon yang benar terhadap pola masukan yang serupa tapi tidak sama dengan pola yang dipakai selama pelatihan. Proses pengenalan pola dilakukan melalui pembelajaran terbimbing (*supervised learning*) pada jaringan *multilayer* untuk meminimalkan *error* terhadap jaringan yang menghasilkan keluaran. Pelatihan *backpropagation* meliputi 3 fase, yaitu perambatan maju, perambatan balik, dan perubahan bobot.

##### 1. Perambatan Maju (*Feedforward Propagation*)

Selama perambatan maju, sinyal masukan akan dipropagasikan ke lapisan tersembunyi menggunakan fungsi aktivasi. Setelah itu keluaran dari tiap lapisan tersembunyi akan dirambatkan ke lapisan *output* hingga menghasilkan keluaran jaringan.

##### 2. Perambatan Balik (*Backpropagation*)

Pada perambatan balik, keluaran jaringan akan dibandingkan dengan target yang harus dicapai. Perhitungan kesalahan yaitu menggunakan rumus selisih target dengan hasil keluaran. Bila hasil perhitungan kesalahan lebih kecil dari batas toleransi yang ditentukan, maka perulangan akan dihentikan. Namun bila

kesalahan lebih besar dari batas toleransi, maka kesalahan akan didistribusikan pada unit keluaran dengan dihitung berdasarkan faktor *error*. Hal tersebut dilakukan untuk memodifikasi jaringan dengan mengubah bobot.

### 3. Perubahan Bobot

Pada tahap ini, semua bobot akan dimodifikasi secara bersamaan. Fase perubahan bobot ini dilakukan untuk menurunkan tingkat *error*. Perubahan bobot dapat dilakukan dengan 2 cara, yaitu tanpa menggunakan *momentum* atau dengan *momentum*. *Momentum* merupakan perubahan bobot yang didasarkan pada arah *gradient* pola terakhir dan pola sebelumnya.

Ketiga tahap tersebut akan diulang secara terus menerus hingga terpenuhinya kondisi penghentian. Jumlah iterasi atau *epoch* umumnya digunakan sebagai kondisi penghentian. Bila jumlah iterasi sudah melebihi jumlah maksimum iterasi yang ditetapkan, atau jika kesalahan atau *error* yang terjadi sudah lebih kecil dari batas toleransi atau *target error* yang ditentukan, maka iterasi akan dihentikan.

#### 2.2.4.1. Fungsi Aktivasi

Seperti algoritma jaringan saraf tiruan pada umumnya, metode *backpropagation* memiliki fungsi aktivasi yang digunakan untuk menentukan nilai keluaran berdasarkan nilai total masukan pada *neuron*. Fungsi aktivasi merupakan aturan untuk memetakan penjumlahan input elemen pemrosesan terhadap keluarannya melalui pilihan yang tepat atau alat pengenalan non-linieritas ke dalam desain jaringan. Beberapa fungsi aktivasi yang sering digunakan adalah fungsi *threshold*, fungsi *sigmoid*, dan fungsi identitas/*linear*.

##### 1. Fungsi *Threshold* (Batas Ambang)

Fungsi *threshold* terbagi menjadi 2 jenis, yaitu fungsi *threshold* biner dan fungsi *threshold bipolar*. Berikut ini adalah persamaan fungsi *threshold* biner pada *backpropagation*, dengan  $\alpha$  sebagai nilai *threshold*.

$$f(x) = \begin{cases} 0 & \text{jika } x \leq \alpha \\ 1 & \text{jika } x \geq \alpha \end{cases} \quad (2.1)$$

Adapun fungsi *threshold* yang tidak bernilai 0 atau 1 tetapi bernilai -1 atau 1 disebut *threshold bipolar*. Berikut ini adalah persamaan untuk fungsi *threshold* bipolar.

$$f(x) = \begin{cases} 1 & \text{jika } x \leq \alpha \\ -1 & \text{jika } x \geq \alpha \end{cases} \quad (2.2)$$

## 2. Fungsi Sigmoid

Fungsi *sigmoid* terbagi menjadi fungsi *sigmoid* biner dan fungsi *sigmoid bipolar*. Fungsi aktivasi ini sering dipakai karena mudah diturunkan. Berikut ini adalah persamaan untuk fungsi *sigmoid* biner. Fungsi ini memiliki nilai dengan rentang 0 sampai 1.

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

Berikut ini adalah persamaan untuk fungsi *sigmoid bipolar*.

$$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (2.4)$$

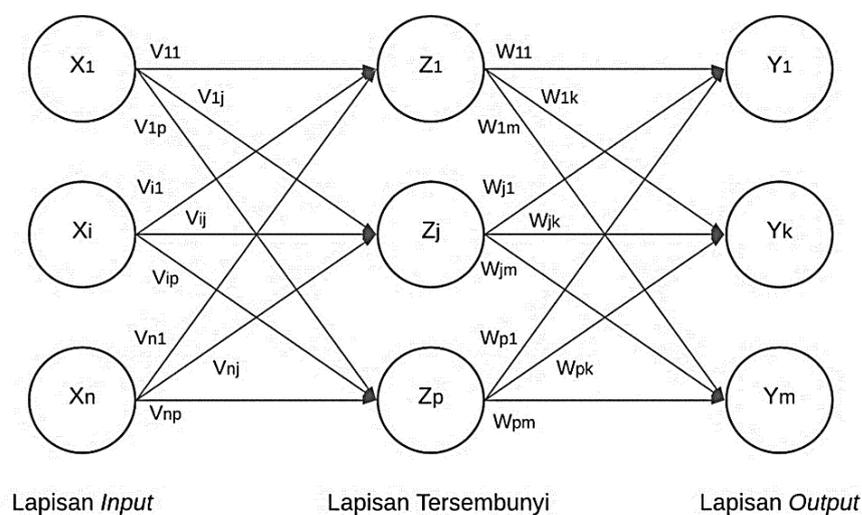
## 3. Fungsi Identitas/Linear

Fungsi *linear* digunakan jika keluaran jaringan diharapkan berupa sembarang bilangan riil.

$$f(x) = x \quad (2.5)$$

### 2.2.4.2. Arsitektur Jaringan

Jaringan *backpropagation* memiliki sebuah arsitektur yang menentukan pola hubungan antar *neuron*. Arsitektur jaringan *backpropagation* berupa hubungan antar unit pada lapisan input, lapisan tersembunyi, dan lapisan *output* (Puspitaningrum, 2006). Unit pada lapisan input terhubung dengan unit pada lapisan tersembunyi. Setiap unit di lapisan tersembunyi terhubung dengan unit yang ada di lapisan *output*. Arsitektur jaringan saraf tiruan *backpropagation* dapat dilihat pada gambar 2.1 berikut.



Gambar 2.1 Arsitektur Jaringan Saraf Tiruan *Backpropagation*

Penjelasan tentang masing-masing lapisan pada arsitektur jaringan saraf tiruan *backpropagation* dijabarkan pada uraian berikut:

1. Lapisan Input

Lapisan input terdiri dari satu set *neuron* yang berisi *neuron-neuron* atau unit-unit input. Unit-unit input bertugas menerima pola input dari luar. Sehingga unit-unit input berguna sebagai penghubung jaringan ke sumber data. Pada gambar 2.1, lapisan input memiliki 3 *neuron*, yaitu  $X_1$ ,  $X_i$ , dan  $X_n$ . Tiap *neuron* mengirimkan nilai bobot yang dilambangkan dengan  $v$  kepada lapisan tersembunyi untuk proses pembelajaran menggunakan jaringan saraf tiruan *backpropagation*.

2. Lapisan Tersembunyi

Lapisan tersembunyi terdiri dari unit-unit tersembunyi. Unit-unit tersembunyi ini memiliki nilai *output* yang tidak dapat diamati secara langsung. Pada gambar 2.1, lapisan tersembunyi memiliki 3 *neuron*, yaitu  $Z_1$ ,  $Z_j$ , dan  $Z_p$ . Tiap *neuron* mengirimkan nilai bobot yang dilambangkan dengan  $w$  kepada lapisan *output*.

3. Lapisan *Output*

Lapisan *output* terdiri dari unit-unit *output*. Unit-unit *output* ini merupakan hasil pembelajaran menggunakan jaringan saraf tiruan. Pada gambar 2.1, lapisan *output* memiliki 3 *neuron*, yaitu  $Y_1$ ,  $Y_k$ , dan  $Y_m$ . Ketiga *output* ini merupakan solusi untuk memecahkan suatu permasalahan dari hasil dari proses pembelajaran.

#### 2.2.4.3. Algoritma Pelatihan *Backpropagation*

Algoritma pelatihan *backpropagation* antara lain sebagai berikut:

1. Langkah 0: Tetapkan nilai maksimum *epoch*, *target error*, *learning rate*, *momentum*, serta arsitektur *backpropagation* yang digunakan. Selanjutnya inisialisasi semua bobot dengan bilangan acak. Proses perhitungan *backpropagation* akan dilakukan selama nilai *epoch* lebih kecil dari nilai maksimum *epoch* dan nilai *error* lebih besar dari *target error*.

2. Langkah 1: Apabila kondisi berhenti belum terpenuhi, maka lakukan langkah berikut:

1. Untuk setiap pasangan pelatihan, lakukan:

Tahap Perambatan Maju (*Feedforward Propagation*)

1. Tiap unit masukan ( $X_i$ ,  $i = 1, 2, 3, \dots, n$ ) menerima sinyal  $x_i$  dan meneruskan sinyal ke semua unit pada lapisan tersembunyi.

2. Setiap unit tersembunyi ( $Z_j$ ,  $j = 1, 2, 3, \dots, p$ ) menjumlahkan bobot sinyal input dengan persamaan berikut:

$$z\_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (2.6)$$

Keterangan:

$z\_in_j$  : Total sinyal masukan pada lintasan j

$v_{0j}$  : Nilai bobot bias

$x_i$  : Nilai input unit i

$v_{ij}$  : Bobot antara unit i dengan lapisan unit j

Gunakan fungsi aktivasi untuk menghitung sinyal dari *hidden* unit, kemudian kirim sinyal tersebut ke semua unit *output*.

Berikut ini adalah persamaan untuk menghitung sinyal dari *hidden* unit:

$$z_j = f(z\_in_j) \quad (2.7)$$

3. Tiap unit *output* ( $Y_k$ ,  $k = 1, 2, 3, \dots, m$ ) menjumlahkan bobot sinyal input:

$$y\_in_k = w_{0k} + \sum_{i=1}^p z_i w_{jk} \quad (2.8)$$

Keterangan:

$y\_in_k$  : Total sinyal masukan pada lintasan j

$w_{0k}$  : Nilai bias pada lapisan tersembunyi

$z_i$  : Nilai input dari unit i

$w_{jk}$  : Bobot unit k dengan lapisan unit j

Gunakan fungsi aktivasi untuk menghitung sinyal *output*-nya.

Berikut ini adalah rumus untuk menghitung sinyal dari *output* unit.

$$y_k = f(y\_in_k) \quad (2.9)$$

2. Untuk tiap-tiap pasangan pelatihan, lakukan:

Tahap Perambatan Balik (*Backpropagation*)

1. Tiap unit *output* ( $Y_k$ ,  $k = 1, 2, 3, \dots, m$ ) menerima pola target sesuai pola input pelatihan, kemudian hitung *error* dengan persamaan berikut, di mana  $t$  adalah target *output* dan  $f'$  adalah turunan dari fungsi aktivasi.

$$\delta_k = (t_k - y_k) f'(y_{in_k}) = (t_k - y_k) y_k(1 - y_k) \quad (2.10)$$

Hitung koreksi bobot yang akan digunakan dengan rumus berikut di mana  $\alpha$  adalah *learning rate*.

$$\Delta w_{jk} = \alpha \delta_k z_j \quad (2.11)$$

Hitung juga koreksi bias yang digunakan untuk memperbaiki nilai  $w_{0k}$  dengan rumus berikut, di mana nilai  $\delta_k$  adalah faktor kesalahan pada unit keluaran  $k$ .

$$\Delta w_{0k} = \alpha \delta_k \quad (2.12)$$

2. Hitung faktor kesalahan pada unit tersembunyi berdasarkan kesalahan di tiap unit tersembunyi ( $Z_j$ ,  $j = 1, 2, 3, \dots, p$ ) dengan rumus berikut.

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk} \quad (2.13)$$

Untuk menghitung informasi *error*, kalikan nilai  $\delta_{in_j}$  dengan turunan dari fungsinya.

$$\delta_j = \delta_{in_j} f'(z_{in_j}) = \delta_{in_j} z_j (1 - z_j) \quad (2.14)$$

Hitung koreksi bobot yang digunakan untuk memperbaiki nilai  $v_{ji}$  dengan rumus berikut.

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (2.15)$$

Hitung koreksi bias untuk memperbaiki nilai  $v_{0j}$  dengan rumus berikut.

$$\Delta v_{0j} = \alpha \delta_j \quad (2.16)$$

3. Perubahan bobot dan bias

1. Tiap unit *output* ( $Y_k$ ,  $k = 1, 2, 3, \dots, m$ ) memperbarui nilai bias dan bobotnya ( $j = 0, 1, 2, \dots, p$ ) dengan persamaan berikut, di mana  $\mu$  adalah *momentum* yang digunakan untuk memperbaiki nilai bobot dan bias:

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

2. Tiap unit tersembunyi ( $Z_j, j = 1, 2, 3, \dots, p$ ) memperbaiki bias dan bobot ( $i = 0, 1, 2, \dots, p$ ) dengan persamaan berikut:

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \mu \Delta v_{ij} \quad (2.18)$$

4. Uji apakah kondisi berhenti terpenuhi.

#### 2.2.4.4. Algoritma Pengujian *Backpropagation*

Proses pengujian *backpropagation* dilakukan menggunakan bobot dan bias yang didapatkan dari tahap perubahan bobot pada proses pelatihan *backpropagation*. Algoritma pengujian *backpropagation* adalah sebagai berikut:

1. Langkah 1: Hitung semua keluaran di unit tersembunyi  $z_j$  ( $j = 1, 2, \dots, p$ ) dengan rumus berikut.

$$z\_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (2.19)$$

$$z_j = f(z\_in_j) = \frac{1}{1+e^{-z\_in_j}} \quad (2.20)$$

2. Langkah 2: Hitung semua keluaran jaringan di unit  $y_k$  ( $k = 1, 2, \dots, m$ ) dengan rumus berikut.

$$y\_in_k = w_{0k} + \sum_{j=1}^p z_j w_{jk} \quad (2.21)$$

$$y_k = f(y\_in_k) = \frac{1}{1+e^{-y\_in_k}} \quad (2.22)$$

#### 2.2.5. *Black Box Testing*

*Black box testing* adalah pendekatan dalam pengujian perangkat lunak yang dilakukan sesuai kebutuhan standar program secara fungsional dan berfokus pada respon terhadap input yang dipilih tanpa memperhatikan keseluruhan kode program. Melalui pendekatan *black box testing*, pengujian perangkat lunak dapat menemukan fitur/fungsi yang kurang atau kesalahan pada struktur data dan *database*. *Black box testing* juga dapat digunakan untuk mengetahui *behavior* perangkat lunak dan membantu mengidentifikasi ketidaksesuaian tampilan perangkat lunak. Pada *black box testing*, pengujian tidak perlu menguasai bahasa pemrograman untuk melakukan pengujian perangkat lunak. *Black box testing* dapat dilakukan tanpa menggunakan perangkat lunak untuk pengujian atau *software testing*, akan tetapi *black box testing* juga bisa dilakukan dengan *software testing* seperti LoadRunner, QTP, Selenium, dan Jmeter. Dalam melakukan *black box testing*, hal-hal yang harus diperhatikan oleh pengujian antara lain adalah:

1. Penguji harus memastikan terdapat dokumen spesifikasi perangkat lunak.
2. Penguji menyiapkan skenario *case* uji untuk kondisi valid dan tidak valid.
3. Penguji menentukan hasil dari kondisi valid dan tidak valid.
4. Penguji membandingkan hasil uji dengan kondisi yang diharapkan.
5. Penguji dapat mengusulkan perbaikan pada perangkat lunak bila ada dan melakukan pengujian ulang.

#### 2.2.6. Confusion Matrix

*Confusion matrix* adalah salah satu jenis pengujian yang digunakan untuk mengukur akurasi kinerja model klasifikasi *machine learning*, baik klasifikasi 2 kelas (*binary classification*) atau klasifikasi multi kelas (*multi classification*). Pada perhitungan akurasi menggunakan *confusion matrix*, terdapat 4 istilah untuk merepresentasikan hasil proses klasifikasi. Keempat istilah tersebut adalah *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). TP merupakan data yang benar dan terdeteksi benar. TN adalah data yang salah dan terdeteksi salah. FP adalah data yang terdeteksi benar namun seharusnya terdeteksi salah. FN adalah data yang terdeteksi salah, namun seharusnya terdeteksi benar. Tabel berikut merupakan contoh ilustrasi dari *confusion matrix*.

Tabel 2.1 *Confusion Matrix*

		Nilai Aktual	
		<i>True</i>	<i>False</i>
Nilai Prediksi	<i>True</i>	TP	FP
	<i>False</i>	FN	TN

*Confusion matrix* termasuk pengujian analitik yang menampilkan dan membandingkan nilai sebenarnya dengan nilai hasil prediksi. Perhitungan akurasi sistem menggunakan *confusion matrix* dapat dihitung dengan rumus berikut.

$$Akurasi = \frac{TP + TN}{TP + FN + FP + TN} \times 100\% \quad (2.23)$$

Selain digunakan untuk mengukur akurasi, *confusion matrix* umumnya juga digunakan dalam menghitung *precision*, *recall*, dan *F-Measure* dari hasil klasifikasi jaringan syaraf tiruan. *Precision* adalah perbandingan prediksi *True Positive*

dengan keseluruhan data positif yang diprediksi. Berikut adalah persamaan untuk menghitung *precision*.

$$Precision = \frac{TP}{TP + FP} \times 100\% \quad (2.24)$$

*Recall* adalah perbandingan total data positif yang diklasifikasikan bernilai benar atau *true* dengan total data positif. Perhitungan *recall* dilakukan berdasarkan perbandingan prediksi *True Positive* dengan seluruh data yang termasuk kategori *True Positive* dan *False Negative*. Berikut adalah persamaan untuk menghitung nilai *recall*.

$$Recall = \frac{TP}{TP + FN} \times 100\% \quad (2.25)$$

*F-Measure* atau yang juga dapat disebut *F1-Score* adalah rata-rata harmonik dari *recall* dan *precision*. Rataan harmonik ini menitikberatkan pada nilai terkecil yang menghasilkan angka tepat di antara nilai *recall* dan *precision* yang dirata-ratakan. Berikut adalah persamaan untuk menghitung *F-Measure*.

$$F - Measure = \frac{2 \times Recall \times Precision}{Recall + Precision} \quad (2.26)$$