

BAB II

TINJAUAN PUSTAKA

A. Penelitian Terdahulu

1. Mustafidah, *et al.* (2019) telah melakukan penelitian untuk mengetahui kinerja algoritma pelatihan *Levenberg-Marquardt* ditinjau dari epoch yang diperlukan oleh jaringan menggunakan beberapa variasi banyaknya neuron dalam lapisan tersembunyi. Berdasarkan penelitian tersebut menghasilkan informasi bahwa banyaknya neuron yang paling optimal dalam lapisan tersembunyi yang ditinjau dari epoch yang diperlukan untuk mencapai kondisi *error* paling kecil pada neuron masukan 5 adalah 9 neuron dengan rata-rata epoch 10.80 pada laju pembelajaran = 0.01. Pada 10 neuron dalam lapisan input, kondisi optimal pada penggunaan 19 neuron dalam lapisan tersembunyi dengan rata-rata epoch 21.52 pada laju pembelajaran = 0.7. Sementara itu, pada 15 neuron dalam lapisan input, diperoleh rata-rata epoch sebesar 7.38 pada laju pembelajaran = 0.7 yang dicapai pada 29 neuron dalam lapisan tersembunyi. Pengujian ini menggunakan taraf $\alpha = 5\%$.
2. Mustafidah dan Suwarsito (2020) telah melakukan penelitian dengan tujuan pengujian performansi algoritma *Levenberg-Marquardt* ditinjau dari variasi banyaknya jumlah neuron pada *hidden layer* dan *learning rate*. Penelitian ini menghasilkan bahwa performansi terbaik pada algoritma *Levenberg-Marquardt* saat menggunakan *hidden layer* berjumlah 9 neuron

dan lr 0,5. MSE yang ditunjukkan pada kinerja ini sebesar $0,00014 \pm 0,00018$ dari kesalahan target 0,001.

3. Mahfuzh, *et al.* (2020) telah melakukan penelitian untuk mengetahui performa hasil prediksi dan pelatihan dari algoritma *backpropagation* dengan inisialisasi bobot awal menggunakan algoritma Nguyen Widrow dan bilangan random dalam memprediksi Indeks Harga Konsumen (IHK). Penelitian tersebut menghasilkan bahwa Inisialisasi bobot awal dengan Nguyen Widrow dapat mempercepat proses pelatihan dengan jumlah iterasi 3625, sedangkan penggunaan bobot Random menghasilkan iterasi berjumlah 9086. Berdasarkan perbandingan grafik MSE diketahui bahwa algoritma Nguyen Widrow memiliki nilai MSE 0,0001145, sedangkan inisialisasi Random memiliki MSE 0,0001430. Maka algoritma Nguyen Widrow memiliki performa hasil prediksi lebih tinggi dari inisialisasi Random.
4. Christyaditama, *et al.* (2020) telah melakukan penelitian tentang optimasi jaringan syaraf tiruan untuk meningkatkan akurasi pemilihan kompetensi siswa SMK menggunakan *Nguyen Widrow*. Parameter dalam proses pelatihan dan pengujian yaitu algoritma pelatihan *traingdx*, *epoch*: 1.000, kecepatan pembelajaran: 0.1, dan *error*: 0.001. Penelitian ini menghasilkan kesimpulan bahwa kinerja terbaik model pelatihan jaringan syaraf tiruan dengan *Nguyen Widrow* adalah MSE *error* rata-rata terkecil sebesar 0,002 dan akurasi rata-rata tertinggi 96,38%. Pelatihan model jaringan syaraf tiruan data latih dengan *Nguyen Widrow* memiliki MSE

error terkecil sebesar 0,000996 dan akurasi tertinggi 97,49% pada arsitektur JST 9-9-1. Sehingga dapat disimpulkan bahwa Jaringan Syaraf Tiruan dengan bobot awal *Nguyen Widrow* dapat mengatasi masalah JST sehingga memberikan peningkatan akurasi prediksi pemilihan kompetensi siswa lebih baik daripada Jaringan Syaraf Tiruan tanpa *Nguyen Widrow*.

5. Bagaskoro (2019) telah melakukan penelitian menggunakan algoritma jaringan syaraf tiruan *backpropagation* dan *Nguyen Widrow* untuk studi kasus dataset gagal jantung. Penelitian tersebut menghasilkan informasi bahwa penggunaan *backpropagation* tanpa *Nguyen Widrow* lebih unggul dibandingkan dengan *Nguyen*. Proses dengan *Nguyen Widrow* menghasilkan waktu 3 menit 22 detik, nilai akurasinya sebesar 96,974773 dan mendapat nilai MSE sebesar 0.040307. Sedangkan penggunaan *backpropagation* tanpa *nguyen* menghasilkan waktu 3 menit 17 detik, nilai akurasinya sebesar 97,71637333, dan nilai MSE sebesar 0.040307.
6. Aisyah *et al.*, (2018) telah melakukan penelitian untuk mengetahui optimasi algoritma pelatihan *backpropagation* menggunakan *Nguyen Widrow* dan tanpa *Nguyen Widrow* untuk diagnosis angina ludwig. Hasil penelitian tersebut menjelaskan bahwa rata-rata iterasi menggunakan *Nguyen Widrow* lebih cepat yaitu sekitar 0,0624 detik, dan tanpa *Nguyen Widrow* 0,1019 detik. Sedangkan untuk kebutuhan pengenalan pola ditemukan bahwa tanpa *Nguyen Widrow* akurasi mencapai 90%, dan menggunakan *Nguyen Widrow* hanya 70%.

7. Rosmaliati *et al.* (2018) telah melakukan penelitian untuk memprediksi umur transformator dengan menggunakan jaringan syaraf tiruan Nguyen Widrow. Penelitian tersebut menyimpulkan bahwa *Backpropagation Neural Network* (BPNN) konvensional maupun BPNN Nguyen Widrow menunjukkan nilai UMK yang signifikan sesuai proporsi data latih dan data uji. Berdasarkan jenis wavelet yang digunakan dalam hasil MSE, BPNN Nguyen Widrow menunjukkan kinerja yang lebih baik dibandingkan dengan BPNN konvensional.
8. Budianita, *et al.*, (2018) telah melakukan penelitian dengan menerapkan algoritma pembobotan awal Nguyen Widrow dan *backpropagation* untuk diagnosa dan klasifikasi Diabetes Mellitus. Hasil penelitian ini didapatkan informasi bahwa akurasi terbaik sebesar 93,33% pada penggunaan inisialisasi bobot awal Nguyen Widrow. Untuk pengujian bobot acak memberikan akurasi terbaik dengan 66.67%.
9. Heryati, *et al.* (2018) telah melakukan penelitian dibidang jaringan syaraf tiruan untuk memprediksi pencapaian prestasi mahasiswa dengan algoritma *backpropagation*. Penelitian ini menggunakan arsitektur jaringan dengan 3 layer, yaitu *input layer* dengan 5 neuron, *hidden layer* dengan 2 sampai 10 neuron, serta *output layer* dengan 1 neuron. Hasil penelitian tersebut menghasilkan informasi bahwa arsitektur jaringan yang terbaik adalah pada model 5 *input*, 7 neuron pada *hidden layer* dan 1 *output* dengan nilai MSE = 0.01363.

10. Muliantara dan ER (2014) telah melakukan penelitian klasifikasi tanaman menggunakan pendekatan *backpropagation* dan Nguyen Widrow. Penelitian ini bertujuan mengembangkan metode *backpropagation* menggunakan metode Nguyen Widrow, dengan harapan akurasi semakin meningkat. Dari hasil penelitian didapatkan akurasi rata-rata diatas 83% menggunakan Nguyen Widrow, sedangkan tanpa menggunakan Nguyen Widrow menghasilkan rata-rata hanya diatas 54%.
11. Andrian dan Wayahdi (2014) telah melakukan penelitian untuk analisa inialisasi algoritma nguyen widrow pada proses prediksi curah hujan di kota Medan dengan model neuron 8 input layer, 6 hiden layer, dan I output layer. Penelitian tersebut menghasilkan bahwa waktu yang digunakan proses training dengan bobot nguyen widrow lebih lama dibanding dengan menggunakan bobot Random.

B. Landasan Teori

1. *Backpropagation*

a. Pengertian

Berdasarkan yang dikatakan Sutojo, *et al.* (2011) jaringan *backpropagation* termasuk dalam pelatihan terbimbing atau dikenal dengan istilah *supervised* jaringan dipandu oleh sejumlah pasangan data (masukan dan target) yang berfungsi sebagai pembimbing untuk melatih jaringan hingga diperoleh bobot yang terbaik. Pelatihan dilakukan dengan memberikan pasangan pola-pola masukan dan keluaran. Sebaliknya, dalam *unsupervised learning* tidak ada

pembimbing yang digunakan untuk memandu proses pelatihan. Jaringan hanya diberi input, tetapi tidak mendapatkan target yang diinginkan sehingga modifikasi bobot pada jaringan dilakukan menurut parameter tertentu. Pola-pola masukan yang tersedia diklasifikasikan ke dalam kelompok-kelompok yang berbeda.

b. Arsitektur *Backpropagation*

Menurut Kusumadewi (2004) arsitektur jaringan *backpropagation* seperti terlihat pada Gambar 2.1, jaringan terdiri atas 3 unit (*neuron*) pada lapisan *input*, yaitu X1, X2, dan X3, 1 lapisan *output*, yaitu Y. Bobot yang menghubungkan X1, X2 dan X3 dengan neuron pertama pada lapisan tersembunyi, adalah V11, V12, V13 (VIJ: bobot yang menghubungkan neuron *input* ke-i ke neuron ke-j pada lapisan tersembunyi). Perlu diingat bahwa, untuk pemakaian *toolbox* nnet pada Matlab, bobot VIJ memiliki pengertian yang sebaliknya VIJ : bobot yang menghubungkan neuron ke-j pada suatu lapisan neuron ke-i pada lapisan sesudahnya). Misal: V12 adalah bobot yang menghubungkan neuron ke-2 pada lapisan *input*, ke neuron ke-1 pada lapisan tersembunyi. Kembali ke Gambar 2.1, b11 dan b12 adalah bobot bias yang menuju ke neuron pertama dan kedua pada lapisan tersembunyi. Bobot yang menghubungkan Z1 dan Z2 dengan neuron pada lapisan *output*, adalah W1 dan W2. Bobot bias b2 menghubungkan lapisan tersembunyi dengan lapisan *output*. Model arsitektur *backpropagation* ditunjukkan pada gambar 2.1.



Gambar 2.1 Model arsitektur *backpropagation*

2. Algoritma *Levenberg-Marquardt* (*trainlm*)

Algoritma *Levenberg-Marquardt* dikembangkan oleh Kenneth dan Donald Marquardt, memberikan solusi numerik untuk masalah meminimalkan fungsi nonlinier. Dalam bidang jaringan syaraf tiruan, algoritma ini cocok untuk pelatihan kecil dan menengah (Arhami dan Desiani, 2005).

Algoritma *Levenberg-Marquardt* merupakan pengembangan dari algoritma *backpropagation*. Algoritma ini dibangun untuk mengatasi beberapa kekurangan yang ada pada algoritma *backpropagation* dengan memanfaatkan teknik optimasi numerik standar. Tujuan dari *Levenberg-Marquardt* adalah meminimalkan total *error* (Arhami dan Desiani, 2005).

3. Parameter Jaringan

Menurut Kusumadewi (2004) dalam pelatihan *backpropagation* dapat menggunakan perintah *train* ada beberapa parameter pelatihan yang dapat digunakan diantaranya yaitu:

a. Maksimum *Epoch*

Maksimum *epoch* adalah jumlah *epoch* maksimum yang boleh dilakukan selama proses pelatihan. Iterasi akan dihentikan apabila nilai *epoch* melebihi maksimum *epoch*.

Intruksi: **net.trainParam.epochs = MaxEpoch**

Nilai default untuk maksimum *epoch* adalah 10.

b. Kinerja tujuan atau target *error*

Kinerja tujuan adalah target nilai fungsi kinerja. Iterasi akan dihentikan apabila nilai fungsi kinerja kurang dari atau sama dengan kinerja tujuan.

Intruksi: **net.trainParam.goal = TargetError**

Nilai default untuk kinerja tujuan adalah 0.

c. *Learning rate*

Learning rate adalah laju pembelajaran. Semakin besar nilai *learning rate* akan berimplikasi pada semakin besarnya langkah pembelajaran. Jika *learning rate* diset terlalu besar, maka algoritma akan menjadi tidak stabil. Sebaliknya, jika *learning rate* diset terlalu

kecil, maka algoritma akan konvergen dalam jangka waktu yang sangat lama.

Intruksi: **$\text{net.trainParam.lr} = \text{LearningRate}$**

Nilai default untuk *learning rate* adalah 0.01

4. Neuron Tersembunyi (*Hidden Layer*)

Menurut Heaton (2008) menentukan jumlah neuron di lapisan tersembunyi adalah bagian yang penting dalam menentukan arsitektur *neural network*, walaupun dalam lapisan ini tidak secara langsung berinteraksi dengan lingkungan eksternal akan tetapi, lapisan ini memiliki pengaruh besar pada hasil akhir. Tujuan dari lapisan tersembunyi adalah memungkinkan jaringan syaraf menghasilkan *output* yang diharapkan dari *input* yang diberikan. Jumlah neuron dalam lapisan tersembunyi dapat didasarkan pada:

- a. Jumlah neuron tersembunyi lebih besar dari jumlah neuron input atau jumlah neuron output.
- b. Jumlah neuron tersembunyi $\frac{2}{3}$ ukuran lapisan input ditambah neuron lapisan keluaran.
- c. Jumlah neuron yang tersembunyi kurang dari dua kali neuron dari lapisan input.

5. Inisialisasi bobot awal Secara Random

Pemilihan bobot awal sangat mempengaruhi jaringan syaraf dalam mencapai minimum global (atau mungkin hanya lokal saja) terhadap nilai *error*, serta cepat atau tidaknya proses pelatihan menuju kekonvergenan.

Apabila nilai bobot awal terlalu besar, maka input ke setiap lapisan tersembunyi atau lapisan output akan jatuh pada daerah dimana turunan fungsi sigmoidnya akan sangat kecil. Sebaliknya, apabila nilai bobot awal terlalu kecil, maka input ke setiap lapisan tersembunyi atau lapisan output akan sangat kecil, yang akan menyebabkan proses pelatihan akan berjalan sangat lambat. Biasanya bobot awal diinisialisasi secara random dengan nilai antara -0,5 sampai 0,5 (atau -1 sampai 1, atau interval yang lainnya) (Kusumadewi, 2004).

6. Inisialisasi Bobot Awal dengan Metode Nguyen Widrow

Menurut Kusumadewi (2004) menyatakan metode Nguyen Widrow akan menginisialisasi bobot-bobot lapisan dengan nilai antara -0,5 sampai 0,5. Sedangkan bobot-bobot dari lapisan input ke lapisan tersembunyi dirancang sedemikian rupa sehingga dapat meningkatkan kemampuan lapisan tersembunyi dalam melakukan proses pembelajaran. Metode Nguyen-Widrow secara sederhana dapat diimplementasikan dengan prosedur sebagai berikut:

Tetapkan: n = jumlah neuron (unit) pada lapisan input

p = jumlah neuron (unit) pada lapisan tersembunyi

β = faktor penskalaan = $(0.7 (p)^{1/n})$

Kerjakan untuk setiap unit pada lapisan tersembunyi ($j=1,2,\dots,p$):

- a. Inisialisasi bobot-bobot dari lapisan input ke lapisan tersembunyi.

V_{ij} = bilangan random antara -0,5 sampai 0,5 (atau antara $-\gamma$ sampai γ).

b. Hitung: $\|V_j\|$

c. Inisialisasi ulang bobot-bobot:

$$V_{ij} = \frac{\beta V_{ij}}{\|V_j\|} \dots \dots \dots (2. 1)$$

d. Set bias:

b_{1j} = bilangan random antara $-\beta$ sampai β .

Analisis Nguyen Widrow didasarkan atas fungsi aktivasi tangen hiperbolik.

7. MSE (*Mean Square Error*)

Menurut Kusumadewi dan Hartati (2006) menyatakan perhitungan kesalahan atau *error* merupakan pengukuran bagaimana jaringan dapat belajar dengan baik sehingga jika dibandingkan dengan pola yang baru akan dengan mudah dikenali. Kesalahan pada keluaran jaringan merupakan selisih antara keluaran sebenarnya (*current output*) dan keluaran yang diinginkan (*desired output*). Selisih yang dihasilkan antara keduanya biasanya ditentukan dengan cara dihitung menggunakan suatu persamaan. Dalam penelitian ini, *error* (kesalahan) dihitung menggunakan MSE (*Mean Squared Error*). MSE merupakan fungsi kinerja jaringan yang mengukur kinerja berdasarkan rata-rata dari kuadrat *error*nya. Persamaan yang digunakan adalah seperti pada persamaan berikut.

$$MSE = \frac{\sum p \sum j (T_{jp} - X_{jp})^2}{n_p n_o} \dots \dots \dots (2. 2)$$

T_{jp} = nilai keluaran jaringan syaraf

X_{jp} = nilai target yang diinginkan untuk setiap keluaran

n_p = jumlah seluruh pola

n_o = jumlah keluaran

MSE dapat memberikan gambaran terhadap seberapa konsisten model yang dibangun. Dengan meminimalkan nilai MSE, berarti meminimalkan varian model. Model yang memiliki varian kecil mampu memberikan hasil yang relatif lebih konsisten untuk seluruh data *input* dibandingkan dengan model dengan varian besar (MSE besar).

8. Matlab

Menurut Tjoellong (2017) Matlab adalah suatu perangkat lunak yang digunakan untuk pemrograman, komputasi teknis dan matematis, serta analisis berbasis matriks. MATLAB merupakan kepanjangan dari *Matrix Laboratory* karena mampu penyelesaian masalah dalam bentuk matriks. Pada tahun 1970 Matlab versi pertama dirilis oleh Cleve Moler. Pada awalnya Matlab didesain untuk menyelesaikan masalah persamaan aljabar linear, namun seiring perkembangan zaman, program Matlab terus mengalami perkembangan baik dari segi fungsi ataupun performa komputasi.

Matlab adalah bahasa pemrograman level tinggi (ingat dalam dunia pemrograman semakin tinggi level bahasa semakin mudah cara menggunakannya) yang dikhususkan untuk komputasi teknis. Bahasa ini mengintegrasikan kemampuan komputasi, visualisasi, dan pemrograman dalam sebuah lingkungan yang tunggal dan mudah digunakan. Matlab memberikan sistem interaktif yang menggunakan konsep *array*/matrik sebagai standar variabel elemennya tanpa membutuhkan pen-deklarasi-an *array* seperti pada bahasa lainnya (Away, 2010).

9. SPSS

Taniredja dan Mustafidah (2011) menyatakan bahwa SPSS (*Statistical Package for Social Science*) yaitu paket statistika untuk ilmu-ilmu sosial, akan tetapi SPSS banyak juga digunakan untuk bidang-bidang lain yang memang membutuhkan statistika. Sejak dikeluarkannya SPSS dengan versi *under DOS* sampai sekarang dengan versi *under Windows*, sudah dikembangkan SPSS sampai generasi atau *release 17* yang paling baru dengan penambahan fasilitas yang makin lengkap seperti grafik pengendali untuk *quality control* dan penambahan fasilitas untuk *link S-Plus* yaitu *Package Statistika* terbaru yang sangat cocok untuk tujuan ilmiah dan pengembangannya, tidak hanya pengolahan data semata.

Menurut Basuki (2015), SPSS adalah sebuah program aplikasi yang memiliki kemampuan analisis statistik cukup tinggi serta sistem manajemen data pada lingkungan grafis dengan menggunakan menu-

menu deskriptif dan kotak-kotak dialog yang sederhana sehingga mudah untuk dipahami cara pengoperasiannya. Pada awalnya SPSS dibuat untuk keperluan pengolahan data statistik untuk ilmu-ilmu sosial, sehingga kepanjangan SPSS itu sendiri adalah *Statistical Package for the Social Sciences*. Sekarang kemampuan SPSS diperluas untuk melayani berbagai jenis pengguna (*user*), seperti untuk proses produksi di pabrik, riset ilmu sains dan lainnya. Dengan demikian, sekarang kepanjangan dari SPSS adalah *Statistical Product and Service Solutions*. SPSS banyak digunakan dalam berbagai riset pemasaran, pengendalian dan perbaikan mutu (*quality improvement*), serta riset-riset sains.

10. Uji-t Berpasangan (*Paired sample t-test*).

Menurut Nuryadi *et al* (2017), Uji-t Berpasangan (*paired t-test*) adalah salah satu metode pengujian hipotesis dimana data yang digunakan tidak bebas (berpasangan). Ciri-ciri yang paling sering ditemui pada kasus yang berpasangan adalah satu individu (objek penelitian) dikenai 2 buah perlakuan yang berbeda. Walaupun menggunakan individu yang sama, peneliti tetap memperoleh 2 macam data sampel, yaitu data dari perlakuan pertama dan data dari perlakuan kedua. Hipotesis dari kasus ini dapat ditulis:

$$H_0 = \mu_1 - \mu_2 = 0 \text{ atau } \mu_1 = \mu_2$$

$$H_a = \mu_1 - \mu_2 \neq 0 \text{ atau } \mu_1 \neq \mu_2$$

H_a berarti bahwa selisih sebenarnya dari kedua rata-rata tidak sama dengan nol.

Rumus T-Test Sampel Berpasangan adalah sebagai berikut:

$$t_{hit} = \frac{\bar{D}}{\frac{SD}{\sqrt{n}}} \dots \dots \dots (2. 3)$$

$$SD = \sqrt{var} \dots \dots \dots (2. 4)$$

$$var(s^2) = \frac{1}{1-n} (x_i - \bar{x})^2 \dots \dots \dots (2. 5)$$

t = nilai t hitung

\bar{D} = rata-rata selisih pengukuran 1 dan 2

SD = standar deviasi selisih pengukuran 1 dan 2

n = jumlah sample.

Intrepretasi penggunaan uji t-test:

a. Untuk menginterpretasikan uji t-test terlebih dahulu harus ditentukan:

- Nilai signifikasi α
- Df (*Derajat of freedom*) = N-k, khusus untuk *paired sample t-test*
df = N - 1.

b. Bandingkan nilai t_{hit} dengan $t_{tab=a;n-1}$

c. Apabila:

$t_{hit} > t_{tab} \rightarrow$ berbeda secara signifikasi (H_0 ditolak)

$t_{hit} > t_{tab} \rightarrow$ tidak berbeda secara signifikasi (H_0 diterima).