

BAB II

TINJAUAN PUSTAKA

A. Penelitian Terdahulu

1. Pada kasus prediksi curah hujan di Kota Medan dengan metode *Backpropagation* neural network dengan inialisasi bobot *Nguyen Widrow* menghasilkan kesimpulan pada proses training JST, semakin kecil nilai target *error* maka nilai iterasinya akan semakin besar dan keakurasiannya juga semakin tinggi. Proses training dengan inialisasi bobot *Nguyen Widrow* tidak lebih baik dari bobot *Random*, hal ini dilihat dari proses training dengan bobot *Nguyen Widrow* lebih lama dibandingkan proses training dengan bobot *Random* dengan tingkat keakurasian terbesar adalah 43.1 %, dengan target *error* 0.007 (Andrian dan Wayahdi, 2014).
2. Implementasi metode *Backpropagation* untuk memprediksi tingkat kelulusan uji kompetensi siswa kelas 12 Teknik Komputer Jaringan (TKJ), memberi kesimpulan bahwa metode *Backpropagation* dapat digunakan untuk memprediksi tingkat kelulusan uji kompetensi siswa kelas 12 TKJ dengan sangat baik dengan tingkat akurasi yang tinggi yaitu 95% pada pola 5-6-1 (Syofneri, Defit dan Sumijan, 2019).
3. Penelitian yang dilakukan oleh Mustafidah dan Suwarsito (2015b) memberikan kesimpulan bahwa algoritma *Levenberg Marquardt* merupakan algoritma yang memiliki performa terbaik yang dapat

menghasilkan rata-rata *error* terkecil dengan tingkat pengujian $\alpha = 5\%$ menggunakan 10 *neuron* pada lapisan tersembunyi.

4. Penelitian yang dilakukan oleh Mustafidah, Budiastanto, dan Suwarsito (2019) memberi kesimpulan bahwa algoritma *Levenberg Marquardt* merupakan algoritma pelatihan yang menghasilkan *error* terkecil sebesar 0.001001 dengan lapisan masukan sebanyak 5, 10, dan 15 *neuron* dengan 1 *neuron* pada lapisan *output*.
5. Pengaruh algoritma inisialisasi *Nguyen Widrow* terhadap algoritma *Backpropagation* dalam prediksi indeks harga konsumen (IHK) memberi kesimpulan bahwa algoritma inisialisasi *Nguyen Widrow* dengan nilai MSE 0,0001145 memiliki performa hasil prediksi lebih tinggi dari inisialisasi *Random* yang memiliki MSE 0,0001430 (Mahfuzh, Widiyanto dan Chamidah, 2020).
6. Implementasi metode *Backpropagation* dengan inisialisasi bobot *Nguyen Widrow* untuk peramalan harga saham memberikan kesimpulan bahwa aplikasi peramalan dengan bobot *Nguyen Widrow* pada metode *Backpropagation* menunjukkan akurasi yang sangat baik berdasarkan nilai MAPE yang berada dibawah 10% yaitu sebesar 1,84% (Kurniawan, Wibawanto dan Widodo, 2019).

B. Landasan Teori

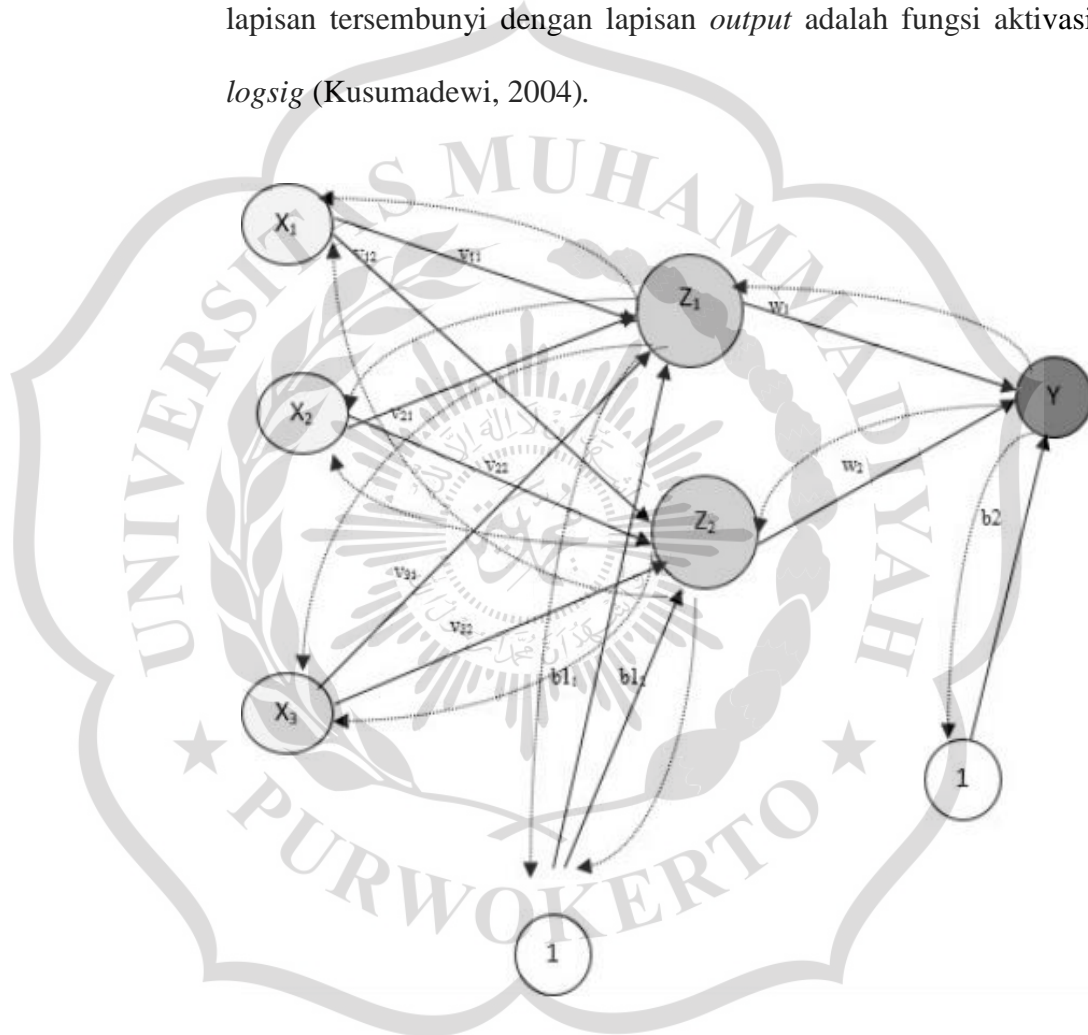
1. Jaringan Syaraf Tiruan

- a. Algoritma Pembelajaran *Backpropagation*

Menurut Kusumadewi (2004) *Backpropagation* merupakan algoritma pembelajaran yang terawasi dan biasanya digunakan oleh perceptron dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan *neuron-neuron* yang ada pada lapisan tersembunyinya. Algoritma *Backpropagation* menggunakan *error output* untuk mengubah nilai bobot-bobotnya dalam arah mundur (*backward*). Untuk mendapatkan *error* ini, tahap perambatan maju (*forward propagation*) harus dikerjakan terlebih dahulu. Pada saat perambatan maju, *neuron-neuron* diaktifkan dengan fungsi aktivasi yang dapat didiferensiasikan seperti sigmoid.

Arsitektur jaringan *Backpropagation* seperti terlihat pada Gambar 2.1, Jaringan terdiri atas 3 unit (*neuron*) pada lapisan *input*, yaitu X_1 , X_2 dan X_3 ; 1 lapisan *output*, yaitu y . Bobot yang menghubungkan X_1 , X_2 dan X_3 dengan *neuron* pertama pada lapisan tersembunyi adalah V_{11} , V_{21} , dan V_{31} (V_{ij} : bobot yang menghubungkan *neuron input* ke- i ke *neuron* ke- j pada lapisan tersembunyi). Perlu diingat bahwa, untuk pemakaian *toolbox nnet* pada Matlab, bobot memiliki pengertian yang sebaliknya (V_{ij} : bobot yang menghubungkan *neuron* ke- j pada suatu lapisan *neuron* ke- i pada lapisan sesudahnya). Misal: V_{12} adalah bobot yang menghubungkan *neuron* ke-2 pada lapisan *input*, ke *neuron* ke-1 pada lapisan tersembunyi. Kembali ke Gambar 2.1, b_1 dan b_1 adalah bobot bias yang menuju ke *neuron* pertama dan kedua pada lapisan

tersembunyi. Bobot yang menghubungkan dan dengan *neuron* pada lapisan *output*, adalah w_1 dan w_2 . Bobot bias b_2 menghubungkan lapisan tersembunyi dengan lapisan *output*. Fungsi aktivasi yang digunakan, antara lapisan *input* dan lapisan tersembunyi, dan antara lapisan tersembunyi dengan lapisan *output* adalah fungsi aktivasi *logsig* (Kusumadewi, 2004).



Gambar 2.1 Arsitektur *Backpropagation*

b. Algoritma *Levenberg Marquardt* (trainlm)

Algoritma *Levenberg Marquardt*, yang dikembangkan oleh Kenneth Levenberg dan Donald Marquardt, memberikan solusi

numerik untuk masalah meminimalkan fungsi *non-linear*. Algoritma ini cepat dan memiliki konvergensi yang stabil. Algoritma *Levenberg Marquardt* menggunakan perbaikan metode *Newton* yang diketahui sebagai metode yang sangat cepat dan akurat untuk mendapatkan *error* minimum (Yu dan Wilamowski, 2011) dan (Kusumadewi, 2004).

Algoritma *Levenberg Marquard* merupakan algoritma pelatihan jaringan yang memperbarui nilai bobot dan bias. Algoritma ini dirancang secara khusus untuk meminimalisasi jumlah kuadrat *error* dan mampu memperbarui bobot dengan cara memperkecil nilai *error* dari selisih 18 bobot baru dengan bobot lama. Pada saat yang bersamaan algoritma ini akan mempertahankan konstanta pembelajaran agar tidak menjadi besar (Bishop, 1995).

c. MSE

Menurut Kusumadewi dan Hartati (2006) perhitungan *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*). Selisih yang dihasilkan antara keduanya biasanya ditentukan dengan cara dihitung menggunakan suatu persamaan. *Error* dihitung menggunakan MSE (*Mean Squared Error*). MSE merupakan fungsi

kinerja jaringan yang mengukur kinerja berdasarkan rata-rata dari kuadrat *error*nya. Berikut persamaan 2.1:

$$MSE = \frac{\sum p \sum j (T_{jp} - X_{jp})^2}{n_p n_o} \dots\dots\dots (2.1)$$

Keterangan:

T_{jp} = nilai keluaran jaringan syaraf

X_{yp} = nilai target yang di inginkan 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 MSE besar.

d. *Hidden Layer*

Jaringan syaraf memiliki dua lapisan yaitu lapisan *input* dan keluaran *output*. Diantara dua lapisan ini terdapat lapisan lain yang disebut *layer* tersembunyi. Lapisan tersembunyi ini disisipkan diantara lapisan *input* dan *output*. Lapisan yang tersembunyi ini mengambil struktur yang lebih kompleks. Tujuan dari lapisan tersembunyi adalah memungkinkan jaringan syaraf menghasilkan *output* yang diharapkan dari *input* yang diberikan. Lapisan

tersembunyi seperti sebuah “kotak hitam” ini didefinisikan dalam istilah jumlah *neuron* untuk lapisan tersembunyi dan lapisan *output*. Jaringan syaraf menghasilkan *output* yang benar yang dilakukan oleh lapisan tersembunyi. Salah satu tantangannya adalah untuk menghindari pembuatan struktur tersembunyi yang kompleks atau terlalu sederhana. Struktur tersembunyi yang terlalu rumit akan memakan waktu terlalu lama untuk melatih. Apabila strukturnya terlalu sederhana maka tidak akan banyak ditemukan masalahnya. Titik awal yang baik adalah lapisan tersembunyi tunggal dengan jumlah *neuron* yang sama dengan dua kali lapisan *input*. Berdasarkan kinerja jaringan ini, jumlah lapisan tersembunyi dari jumlah *neuron* akan meningkat atau menurun (Heaton, 2008).

Banyaknya *neuron* dalam lapisan tersembunyi akan sangat berpengaruh terhadap kinerja jaringan terutama dalam *error* yang dihasilkan yang berdampak pada tingkat keakuratan keluaran jaringan. Penentuan jumlah *neuron* yang digunakan dalam lapisan tersembunyi mengikuti aturan HEATON:

- 1) Jumlah *neuron* dalam lapisan tersembunyi baiknya di antara ukuran lapisan masukan dan lapisan keluaran.
- 2) Jumlah *neuron* dalam lapisan tersembunyi sebaiknya $\frac{2}{3}$ dari ukuran lapisan masukan ditambah ukuran lapisan keluaran.

3) Jumlah *neuron* dalam lapisan tersembunyi sebaiknya lebih kecil atau kurang dari dua kali ukuran *neuron* dalam lapisan *input* (Heaton, 2015).

e. Parameter Jaringan

Menurut Kusumadewi (2004) dalam pelatihan *Backpropagation* dapat menggunakan perintah `train` ada beberapa parameter yang harus diset pada pelatihan ini:

1) Maksimum Epoch

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

Instruksi `net.trainParam.eponchs=MaxEpoch`

2) Kinerja Tujuan

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

Instruksi `net.trainParam.goal=TargetError`

3) *Learning Rate*

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

Rate di set terlalu kecil, maka algoritma akan konvergen dalam jangka waktu yang sangat lama.

Instruksi **net.trainParam.lr=LearningRate**

f. Fungsi Aktivasi

Fungsi aktivasi digunakan jaringan syaraf tiruan yang dilatih dengan metode *Backpropagation* adalah fungsi sigmoid bipolar dan fungsi linear (identitas). Fungsi sigmoid bipolar hampir sama dengan fungsi sigmoid biner hanya saja *output* dari fungsi ini memiliki *range* antara 1 sampai -1. Fungsi linear memiliki nilai *output* yang sama dengan nilai *inputnya* (Kusumadewi, 2004).

g. Model *Neuron*

Menurut Suyanto (2007) *neuron* adalah unit pemrosesan informasi yang merupakan dasar dari operasi jaringan syaraf tiruan. Terdapat tiga elemen dari model *neuron*, yaitu:

- 1) Sekumpulan sinapsis atau jalur hubungan, dimana masing-masing sinapsis memiliki bobot atau kekuatan hubungan.
- 2) Suatu *adder* untuk menjumlahkan sinyal-sinyal *input* yang diberi bobot oleh sinapsis *neuron* yang sesuai. Operasi-operasi yang digambarkan mengikuti aturan *linear combiner*
- 3) Suatu fungsi aktivasi untuk membatasi amplitudo *output* dari setiap *neuron*.

h. Pelatihan Jaringan Syaraf Tiruan

Terdapat dua jenis pelatihan dalam jaringan syaraf tiruan menurut Suyanto (2007).

1) Pembelajaran Terawasi (*Supervised Learning*).

Dalam pelatihan ini jaringan dilatih dengan cara diberikan data yang disebut pelatihan data. Pelatihan data terdiri sekumpulan sampel *input-output* yang diharapkan dan disebut *associative memory*. Setelah jaringan dilatih, *associative memory* dapat mengingat suatu pola. Parameter jaringan berubah-ubah berdasarkan vektor latih dan sinyal kesalahan. Proses perubahan dilakukan secara berulang agar jaringan syaraf terpusat bisa memiliki kemampuan memetakan sekumpulan sampel *input-output* dengan akurasi yang tinggi.

2) Pembelajaran Tidak Terawasi (*Unsupervised Learning*).

Dalam proses pelatihan ini, jaringan dilatih hanya dengan diberi data *input* yang memiliki kesamaan sifat tanpa disertai *output*.

i. Pembobotan *Random*

Pemilihan bobot awal sangat mempengaruhi jaringan syaraf tiruan dalam mencapai minimum global atau mungkin hanya lokal saja terhadap nilai *error*. Apabila pemilihan bobot awal terlalu besar maka *input* 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* ketiap lapisan tersembunyi atau lapisan *output* akan sangat kecil, hal ini

yang akan menyebabkan proses pelatihan akan berjalan sangat lambat, biasanya bobot awal diinisialisasi secara *random* dengan nilai -0,5 sampai 0,5 (atau -1 sampai 0,1 atau interval yang lainnya) (Kusumadewi, 2004).

j. Pembobotan *Nguyen Widrow*

Nguyen Widrow adalah sebuah algoritma yang digunakan untuk inialisasi bobot pada jaringan syaraf tiruan untuk mengurangi waktu pelatihan. *Nguyen Widrow* akan menginisialisasi bobot-bobot dari 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. *Nguyen Widrow* dapat diimplementasikan sebagai berikut (Siang, 2009) dan (Kusumadewi, 2004):

1) Tetapkan :

n = jumlah *neuron* (unit) pada lapisan input

p = jumlah *neuron* (unit) pada lapisan tersembunyi

β = faktor penskalaan ($= 0.7(p)^{1/n} = 0.7 \sqrt[n]{p}$)

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

3) Inialisasi bobot-bobot dari lapisan *input* ke lapisan tersembunyi:

V_{ij} (*lama*) = bilangan *random* antara -0.5 sampai 0.5 (atau antara $-y$ sampai y).

- 4) Hitung $\|V_{ij}(lama)\|$
- 5) Inisialisai ulang bobot-bobot (bobot baru) seperti yang terlihat pada persamaan 2.2:

$$V_{ij} = \frac{\beta \cdot V_{ij}}{\|V_{ij}\|} \dots\dots\dots (2.2)$$

- 6) Set bias yang digunakan sebagai inisialisasi:
 b_{1j} = bilangan *random* antara $-\beta$ sampai β .

2. Matlab

Matlab merupakan perangkat lunak yang digunakan untuk pemrograman, analisis, serta komputasi teknis dan matematis berbasis matriks. Matlab adalah singkatan dari *matriks laboratory* karena mampu menyelesaikan masalah perhitungan dalam bentuk matriks. Matlab versi pertama dirilis pada tahun 1970 pada Cleve Moler. Pada awalnya, Matlab didesain untuk menyelesaikan masalah-masalah persamaan aljabar linear. Seiring berjalannya waktu, program ini terus mengalami perkembangan dari segi fungsi dan performa komputasi.

Bahasa pemrograman yang dikembangkan oleh *MathWorks Inc.* menggabungkan proses pemrograman, komputasi, dan visualisasi melalui lingkungan kerja yang mudah digunakan. Matlab juga memiliki keunggulan umum lainnya, seperti analisis dan eksplorasi data, pengembangan algoritma, pemodelan dan simulasi, visualisasi plot dalam bentuk 2D dan 3D, hingga pengembangan aplikasi antara muka grafis (Tjolleng, 2017).

3. SPSS

Menyatakan bahwa SPSS singkatan dari *Statistical Package For Social Science* yaitu merupakan 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. SPSS berfungsi untuk membantu memproses data-data statistik secara cepat dan tepat, serta menghasilkan berbagai *output* yang dikehendaki oleh para pengambil keputusan. Statistik dapat diartikan sebagai suatu kegiatan yang bertujuan untuk mengumpulkan data, meringkas atau menyajikan data kemudian menganalisis data dengan menggunakan metode tertentu, dan menginterpretasikan hasil dari analisa tersebut (Taniredja dan Mustafidah, 2011).

4. Uji T-Berpasangan

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 (Nuryadi *dkk.*, 2017) :

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

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

H_α berarti selisih sebenarnya dari kedua rata-rata tidak sama dengan 0.

Berikut persamaan 2.3, persamaan 2.4, dan persamaan 2.5:

$$t_{hitung} = \frac{D}{\frac{SD}{\sqrt{n}}} \quad \dots\dots\dots (2.3)$$

Dengan:

$$SD = \sqrt{var} \quad \dots\dots\dots (2.4)$$

$$var(SD^2) = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 \quad \dots\dots\dots (2.5)$$

Keterangan:

t = nilai t hitung

D = rata-rata selisih pengukuran 1 dan 2

SD = standar deviasi selisih pengukuran 1 dan 2

n = jumlah sampel

Apabila :

$t_{hitung} > t_{tab} \rightarrow$ berbeda secara signifikan (H_0 ditolak).

$t_{hitung} < t_{tab} \rightarrow$ tidak berbeda secara signifikan (H_0 diterima).