

BAB II

TINJAUAN PUSTAKA

II.1 Pesantren

Pesantren sesungguhnya merupakan lembaga pendidikan tertua di Indonesia, yang secara nyata telah melahirkan banyak ulama'. Tidak sedikit tokoh Islam lahir dari lembaga pesantren. Bahkan Prof.Dr.Mukti Ali pernah mengatakan bahwa tidak pernah ada ulama yang lahir dari lembaga selain pesantren. Istilah "pesantren" berasal dari kata pe-"santri"-an, dimana kata "santri" berarti murid dalam bahasa Jawa. Istilah "pondok" berasal dari bahasa Arab "*funduuq*" ("فندق") yang berarti penginapan. Khusus di Aceh, pesantren disebut juga dengan nama "*dayah*"(Dkk, 1996).

II.2 Jaringan Syaraf Tiruan

Salah satu metode untuk merepresentasikan buatan dari otak manusia yang selalu mencoba mensimulasikan suatu proses pembelajaran adalah pengertian dari jaringan syaraf tiruan, dalam memberikan sebuah gambaran mengenai jaringan syaraf tiruan yang dirancang dan diterapkan menggunakan program komputer yang mampu menyesuaikan proses perhitungan selama proses pembelajaran yang mirip dengan kinerja otak manusia maka digunakanlah kata buatan. Puspitaningrum dalam bukunya Pengantar Jaringan Syaraf Tiruan mendegenisikan *Artificial Neural Network* sebagai alat yang digunakan untuk memecahkan masalah di bidang-bidang yang melibatkan pengelompokan dan *pattern recognition* (pengenalan pola) yang berada pada cabang ilmu *Artificial*

Intelligence atau kecerdasan buatan(Puspitaningrum, n.d.). Menurut Zilouchian, jaringan syaraf tiruan merupakan sebuah sistem pemrosesan informasi yang memiliki karakteristik menyerupai jaringan syaraf biologis manusia(Zilouchian, 2001). Dari kedua definisi sebelumnya dapat disimpulkan bahwa Jaringan syaraf tiruan merupakan suatu alat untuk memecahkan masalah yang memiliki cara kerja yang mirip dengan jaringan syaraf biologis manusia.

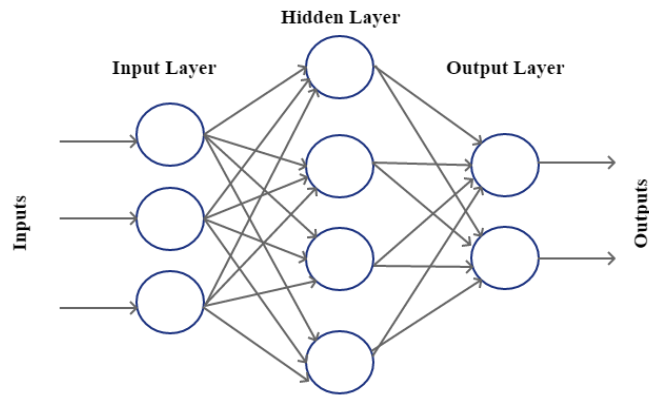
II.3 Feedforward Neural Network

Feedforward Neural Network adalah jaringan syaraf tiruan yang koneksi antar node-nya tidak membentuk sebuah siklus. Kebalikan dari *Feedforward Neural Network* adalah *Recurrent Neural Network*, di mana pada RNN terdapat siklus pada jalur node-nya.

Algoritma *Feedforward Neural Network* merupakan bentuk *neural network* yang paling sederhana karena informasi hanya diproses dalam satu arah. Data dapat melewati beberapa hidden node, namun selalu bergerak dalam satu arah dan tidak pernah mundur ke belakang.

Dengan kata lain, jaringan saraf *Feedforward* hanya memungkinkan sinyal untuk melakukan perjalanan melalui satu jalur saja, yakni dari *input* ke *output*. Tidak ada koneksi umpan balik dari *output* ke dirinya sendiri (*loop*).

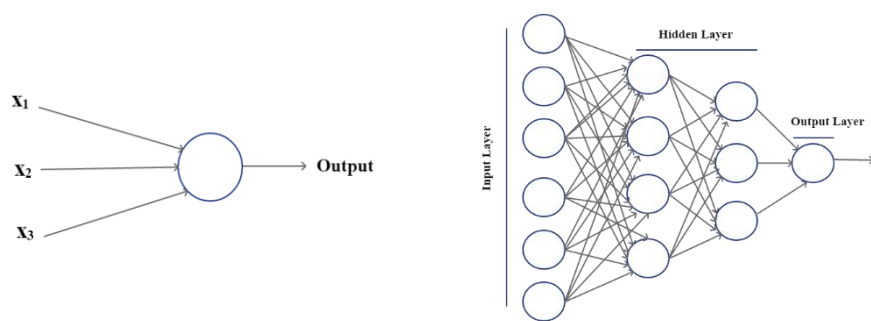
Hal inilah yang menjadi alasan mengapa *neural network* jenis ini disebut sebagai *Feedforward* atau dalam bahasa indonesia diartikan umpan maju (DeepAI, n.d.).



Gambar II. 1 Algoritma *Feedforward Neural Network*

Algoritma *Feedforward Neural Network* sering juga disebut sebagai *multilayer perceptron*. Alasannya karena arsitekturnya menggabungkan beberapa lapisan dari *perceptron*.

Pada gambar di bawah, sebelah kiri adalah arsitektur lapisan *perceptron* dan sebelah kanan adalah *feedforward neural network*. Dapat dilihat bahwa feedforward membentuk lapisannya dengan menggabungkan beberapa *perceptron*.



Gambar II. 2 Arsitektur lapisan perceptron (kiri) dan *feedforward neural network* (kanan)

Arsitektur Feedforward Neural Network terdiri atas input layer, hidden layer, dan output layer.

Lapisan paling kiri disebut input layer, dan neuron di dalam lapisan ini disebut input neuron. Lapisan paling kanan adalah output layer yang berisi output neuron. Lapisan tengah disebut hidden layer, karena neuron pada lapisan ini bukan merupakan input maupun output.

- *Input layer*: Lapisan ini terdiri dari *neuron* yang menerima input dan meneruskannya ke lapisan lain. Jumlah *neuron* pada lapisan *input* harus sama dengan atribut atau fitur dalam *dataset*.
- *Output layer*: Merupakan lapisan keluaran yang berisi hasil fitur yang telah diprediksi. Keluaran yang dihasilkan bergantung pada jenis model yang kita buat.
- *Hidden layer*: Merupakan lapisan yang berada di antara lapisan *input* dan *output*. Lapisan ini berisi beberapa *neuron* yang menerapkan transformasi ke *neuron input* sebelum melewatinya. Saat jaringan dilakukan *training*, nilai bobot diperbarui agar lebih prediktif.
- *Bobot neuron*: Bobot mengacu pada kekuatan atau amplitudo koneksi antara dua *neuron*. Nilai bobot ini sering diinisialisasi ke nilai acak kecil, seperti nilai dalam rentang 0 hingga 1.

Berikut adalah langkah-langkah cara kerja algoritma *Feedforward Neural Network*(Edgell, 2021).

Langkah 1: Serangkaian *input* memasuki jaringan melalui *input layer* dan dikalikan dengan bobot tertentu.

Langkah 2: Setiap nilai hasil perkalian kemudian dijumlahkan. Jika nilai penjumlahan lebih besar dari batas yang ditentukan (biasanya 0), *output* ditetapkan sebagai 1. Jika nilai kurang dari ambang batas, hasilnya akan menjadi -1.

Langkah 3: *Output* dari jaringan saraf kemudian dapat dibandingkan nilai prediksinya dengan bantuan aturan delta. Hal ini berguna bagi jaringan untuk mengoptimalkan bobotnya melalui proses *training* untuk mendapatkan nilai *output* dengan akurasi yang lebih baik. Proses *training* dan pembelajaran ini menghasilkan penurunan gradien.

Langkah 4: Pemutakhiran bobot dilakukan melalui *backpropagation*. Pada langkah ini, setiap *hidden layer* dimodifikasi agar tetap selaras dengan nilai *output* yang dihasilkan oleh lapisan terakhir.

II.4 Backpropagation

Pada bidang peramalan (*forecasting*) untuk pengaplikasiannya banyak menggunakan metode *backpropagation*. Dalam pengaplikasiannya *backpropagation* mendapatkan keseimbangan kemampuan jaringan mengenali pola selama pelatihan serta kemampuan jaringan dalam memberi tanggapan yang benar terhadap pola masukan yang serupa namun tidak sama dengan pola yang dipakai selama pelatihan (Siang, 2005). Menurut Kusumadewi (Kusumadewi, 2003), dalam hal pelatihan metode *backpropagation* memiliki kesamaan dengan metode jaringan syaraf tiruan yang lain. Pelatihan dilakukan dalam proses perhitungan bobot yang pada akhir pelatihan akan didapatkan bobot-bobot yang baik pada jaringan umpan maju (*feedforward*). Untuk meminimumkan kasalahan (*error*), semua dilakukan selama proses pelatihan. Berdasarkan nilai rata-rata

kesalahan kuadrat (MSE) akan dihitung menjadi kesalahan. Sebagai dasar perhitungan juga digunakan fungsi aktivasi dan rata-rata kuadrat kesalahan. Dalam rangka meminimumkan kinerja dan bagaimana mengatur bobot-bobot tersebut, sebagian besar pelatihan jaringan *feedforward* (umpan maju) menggunakan *gradient* dari fungsi aktivasi. *Backpropagation* merupakan suatu teknik yang digunakan untuk menentukan *gradient*. *Gradient negative* merupakan dasar algoritma pelatihan standar *backpropagation* yang meggerakannya kearah sana. Memperbaiki bobot-bobot jaringan dengan arah yang membuat fungsi aktivasi menjadi turun dengan cepat merupakan prinsip dasar dari algoritma *backpropagation*.

Dalam proses pelatihannya, *backpropagation* secara umum melakukan tahapan-tahapan sebagai berikut (Saputra, 2021):

1. Inisialisasi semua bobot dengan bilangan acak kecil.
2. Jika kondisi penghentian belum terpenuhi, lakukan langkah 3-9
3. Untuk setiap pasang data pelatihan, lakukan langkah 4-9
4. Tiap unit masukan menerima sinyal dan meneruskannya ke unit tersembunyi di atasnya

5. Hitung semua keluaran unit tersembunyi z_j ($j = 1, 2, \dots, p$)

$$z_net_j = v_{j0} + \sum_{i=1}^n (x_i v_{ij}) \dots\dots\dots (1)$$

$$z_j = f(z_net_j) = \frac{1}{1+e^{-z_net_j}} \dots\dots\dots (2)$$

6. Hitung semua keluaran jaringan di unit y_k ($k = 1, 2, \dots, m$)

$$y_net_k = w_{k0} + \sum_{j=1}^n (z_j w_{kj}) \dots\dots\dots (3)$$

$$y_k = f(y_net_k) = \frac{1}{1+e^{-y_net_k}} \dots\dots\dots (4)$$

7. Hitung faktor δ unit keluaran berdasarkan kesalahan di setiap unit keluaran y_k
($k = 1, 2, \dots, m$)

$$\delta_k = (t_k - y_k)f(y_{net_k}) = (t_k - y_k)y_k(1 - y_k) \dots\dots\dots (5)$$

δ_k merupakan unit kesalahan yang akan dipakai dalam perubahan bobot layer dibawahnya. Hitung suku perubahan bobot w_{kj} dengan laju percepatan α .

$$\Delta w_{kj} = \alpha \delta_k z_j; k = (1, 2, \dots, n) \dots\dots\dots (6)$$

8. Hitung faktor δ unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi z_j ($j = 1, 2, \dots, p$)

$$\delta_{net_j} = v_{j0} + \sum_{k=1}^n (\delta_k w_{kj}) \dots\dots\dots (7)$$

Faktor δ unit tersembunyi:

$$\delta_j = \delta_{net_j} f(z_{net_j}) = \delta_{net_j} z_j (1 - z_j) \dots\dots\dots (8)$$

Hitung suku perubahan bobot v_{ij} (yang akan dipakai nanti untuk merubah bobot v_{ij})

$$\Delta v_{kj} = \alpha \delta_i x_i; j = (1, 2, \dots, n) \dots\dots\dots (9)$$

9. Hitung semua perubahan bobot. Perubahan bobot garis yang menuju ke unit keluaran w_{kj} (baru)

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) + v_{ji} \dots\dots\dots (10)$$

Dengan ($k = 1, 2, \dots, m$; $j = 0, 1, \dots, p$). Perubahan bobot garis yang menuju ke unit tersembunyi:

$$v_{ji}(\text{baru}) = v_{ji}(\text{lama}) + \Delta v_{ji} \dots\dots\dots (11)$$

II.5 Learning Vector Quantization

Nurkhozin, dkk dalam penelitiannya menyebutkan *learning vector quantization* (LVQ) merupakan jaringan yang terdiri atas unit masukan dan unit keluaran dengan arsitektur lapis-tunggal umpan maju (*Single Layer Feedforward*)(Nurkhozin et al., 2011). Menurut Kususmadewi mendefenisikan LVQ adalah metode yang melakukan pembelajaran pada lapisan kompetitif yang terawasi.

Dalam mengkalifikasikan vector-vektor input suatu lapisan kompetitif akan dengan cepat belajar, Jarak antara vector-vektor masukan akan mempengaruhi kelas-kelas yang didapatkan dari lapisan kompetitif. Secara cepat lapisan kompetitif akan meletakkan kedua vektor kedalam kelas yang sama apabila kedua buah vektor masukan mendekati sama.

Prinsip kerja dari algoritma LVQ adalah pengurangan node-node tetangganya (*neighbour*), sehingga pada akhirnya hanya ada satu *node output* yang terpilih (*winner node*). Pertama kali yang dilakukan adalah melakukan inisialisasi bobot untuk setiap *class*. Setelah diberikan bobot, maka jaringan diberi *input* sejumlah dimensi *node/neuron input*. Setelah *input* diterima jaringan, maka jaringan mulai melakukan perhitungan jarak vektor yang didapatkan dengan menjumlah selisih/jarak antara vektor *input* dengan vektor bobot menggunakan *Euclidean distance*. Secara matematis *Euclidean Distance* dapat dirumuskan (Hariri et al., n.d.):

$$d_j^2 = \sum_{i=0}^{n-1} (X1(t) - W_{ij})^2 \dots\dots\dots (1)$$

Dimana : d_j^2 = distance

X_t = Node data Input
 W_{ij} = Bobot ke-ij

Setelah diketahui tiap-tiap jarak antara nodeoutput dengan input maka dilakukan perhitungan jumlah jarak selisih minimum. Dimana node yang terpilih (winner) berjarak minimum akan di update bobot, update bobot node winner yang dirumuskan sebagai berikut: Jika sesuai target memakai rumus:

$$W_{ij}(t + 1) = W_{ij}(t) + a(t). (x_i(t) - W_{ij}(t)), j \in Ne \dots \dots \dots (2)$$

Dan jika tidak

$$W_{ij}(t + 1) = W_{ij}(t) - a(t). (x_i(t) - W_{ij}(t)), j \in Ne \dots \dots \dots (3)$$

$$0 < (t) < 1$$

Dimana :

- x = input pixel
- w = bobot
- Ne = Nilai *Neighborhood*
- t = waktu
- i = index input
- j = index node output
- α = alpha learning rate

$\alpha(t)$ merupakan *alpha/learning rate* yaitu faktor pengali pada perubahan bobot yang berubah terhadap perubahan *error*. Perubahan *alpha* ini sesuai dengan banyaknya input yang masuk. Faktor pengali *alpha/learningrate* ini akan selalu berkurang bila tidak ada perubahan error. Dalam penelitian ini *alphalearningrate* akan berubah berkurang secara geometris sebagai berikut (Hariri et al., n.d.):

$$(t + 1) = 0.1 * a(t) \dots \dots \dots (4)$$

II.6 Perbandingan Penelitian

Berikut ini adalah perbandingan penelitian yang akan dilakukan dengan penelitian-penelitian sebelumnya.

Tabel II. 1 Ringkasan Penelitian Sebelumnya

No	Makalah	Metode	Kelebihan	Kekurangan
1	Perbandingan Metode <i>Backpropagation</i> dan <i>Learning Vector Quantization</i> (LVQ) Dalam Menggali Potensi Mahasiswa Baru di STMIK PalComTech (Yarza Aprizal , Rabin Ibnu Zainal, Afriyudi)	<i>Backpropagation</i> dan <i>Learning Vector</i>	Tingkat akurasi yang didapat dari metode <i>backproagation</i> adalah 99,17%, lebih besar dibandingkan metode lvq yang hanya mencapai tingkat akurasi sebesar 96,67%	Dalam penelitian ini ruang lingkup hanya dikhususkan untuk mahasiswa di lingkungan Sekolah tinggi yang memiliki dua program studi saja.
2	Analisis Metode <i>Backpropagation</i> Dalam Memprediksi Kelulusan Mahasiswa Studi Kasus STIKOM Tunas Bangsa (Selvi Salsabillah Nasution, Harly Okprana, Ilham Syahputra Saragih)	<i>Backpropagation</i>	Model yang dihasilkan memiliki performa yang baik , dengan dengan MSE 0.27958 dan akurasi 88,57% %	Hasil pelatihan yang didapatkan dapat berbeda dikarenakan spesifikasi komputer sangat berpengaruh terhadap kinerja komputer, secara langsung juga akan berpengaruh terhadap lama waktu pelatihan
3	Sistem Pendukung Keputusan Penjurusan Di Sma Menggunakan Metode <i>Neural Network Backpropagation</i> (Studi Kasus Sma Islam Kepanjen Malang) (Danang Aditya Nugraha	<i>Backpropagation</i>	Model yang sudah memiliki performa yang cukup baik untuk data uji dengan akurasi sebesar 75% dan menghasilkan sebuah aplikasi.	Beberapa fitur pengambilan database dengan browse atau import database guna untuk mempermudah pengambilan

	Wiwik Retnowati)			data perlu ditambahkan.
4	Predicting Patterns of Student Graduation Rates Using <i>Naïve Bayes</i> Classifier and <i>Support Vector Machine</i> (Ayundyah Kesumawati and Dina Tri Utari)	<i>Naïve Bayes</i> Classifier and <i>Support Vector Machine</i>	Didapatkan hasil Klasifikasi Metode Support Vector Machine lebih baik dari pada Naïve Bayes Classifier, dengan akurasi sebesar 69,15%.	Hasil klasifikasi dari metode SVM masih kurang baik
5	Klasifikasi Kelulusan Mahasiswa Menggunakan Algoritma <i>Learning Vector Quantization</i> (Dwi Kartini, Radityo Adi Nugroho, Mohammad Reza Faisal)	<i>Learning Vector Quantization</i>	Model yang dihasilkan memiliki performa yang baik dengan menunjukkan tingkat akurasi 70% untuk data pengujian yang dikenal jaringan sesuai target dan 30 % tidak sesuai dengan target	Penelitian Tidak menghasilkan sebuah aplikasi

II.7 Ringkasan Bab

Pesantren sesungguhnya merupakan lembaga pendidikan tertua di Indonesia, yang secara nyata telah melahirkan banyak ulama'. Tidak sedikit tokoh Islam lahir dari lembaga pesantren. Bahkan Prof.Dr.Mukti Ali pernah mengatakan bahwa tidak pernah ada ulama yang lahir dari lembaga selain pesantren. Istilah "pesantren" berasal dari kata pe-"santri"-an, dimana kata "santri" berarti murid dalam bahasa Jawa. Istilah "pondok" berasal dari bahasa Arab "funduuq" (فندق) yang berarti penginapan. Khusus di Aceh, pesantren disebut juga dengan nama "dayah"

Jaringan syaraf tiruan merupakan salah satu cabang ilmu kecerdasan buatan yang berkembang. Di dalam jaringan syaraf tiruan terdapat berbagai metode yang dapat di adopsi atau dikembangkan diantaranya metode Feedforward Neural Network dan metode Learning Vector Quantization. Kedua metode ini memiliki tingkat akurasi yang cukup baik di dalam pengenalan pola-pola atau data yang terstruktur. Pada penelitian ini akan dilakukan sebuah pengenalan calon santri yang lulus dan tidak lulus berdasarkan data calon santri tahun 2021 berbentuk .csv.

Hasil yang diharapkan dalam penelitian ini yaitu terdapat sebuah analisis mengenai algoritma terbaik antara Feedforward Neural Network dan LVQ untuk digunakan dalam pengenalan kelulusan calon santri yang akan masuk pesantren.