

BAB II

TINJAUAN PUSTAKA

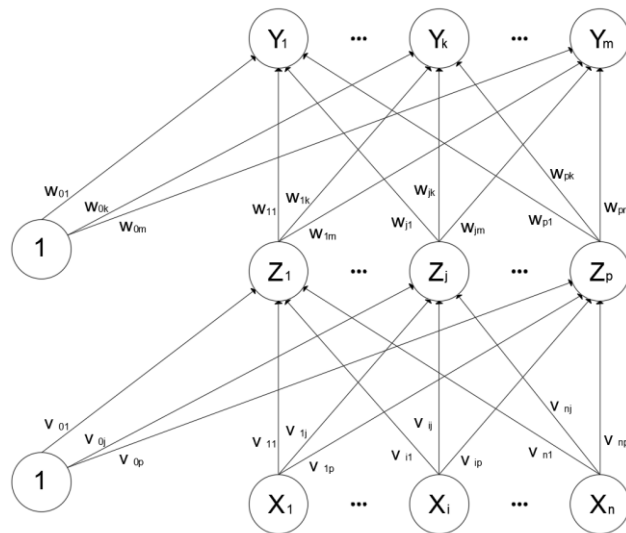
2.1 Kajian Teori

2.1.1 Definisi Jaringan Saraf Tiruan

Jaringan saraf tiruan (JST) merupakan sebuah sistem pemroses informasi yang memiliki kemiripan dengan jaringan saraf biologi dalam hal karakteristik. Berdasarkan buku yang dibuat oleh Jong Jek Siang pada tahun 2005 menjelaskan bahwa JST dibentuk sebagai generalisasi model matematika dari jaringan saraf biologi. JST mampu mengenali sebuah pola dari sebuah data yang sudah pernah ada sebelumnya. Data tersebut akan dipelajari oleh JST dengan cara membaca pola tersebut sehingga JST mempunyai kemampuan untuk memberikan sebuah hasil keputusan yang belum pernah ada atau biasa disebut *forecasting*. JST memiliki banyak metode atau algoritma yang dapat digunakan dalam sebuah peramalan atau prediksi. Algoritma yang cocok dan sering digunakan dalam memprediksi sebuah data yaitu algoritma propagasi balik. Algoritma propagasi balik juga dapat digunakan untuk menyelesaikan permasalahan yang rumit. Hal tersebut dapat terjadi dikarenakan pelatihan yang diberikan kepada jaringan dengan algoritma ini yaitu dengan cara mengenali pola masukan suatu data yang memiliki tingkat akurasi yang tinggi. Jaringan propagasi balik memiliki 3 lapisan yaitu lapisan masukan, lapisan keluran dan lapisan tersembunyi. Jaringan tersebut biasa disebut dengan jaringan arsitektur. Dalam proses pelatihan propagasi balik memiliki 3 fase yaitu fase maju, fase mundur dan terakhir fase modifikasi bobot. Pada fase maju, pola masukan dihitung maju mulai dari lapisan masukan hingga lapisan keluaran. Pada fase mundur, setiap unit keluaran menerima pola yang berhubungan dengan pola masukan untuk dihitung nilai kesalahan. Kesalahan yang sudah dihitung akan di propagasikan mundur. Pada fase

modifikasi bobot bertujuan untuk menurunkan terjadinya kesalahan. Semua fase tersebut akan terus berulang-ulang hingga mendapatkan kondisi yang diinginkan.

Pelatihan tersebut membuat jaringan propagasi balik dapat memberikan respon yang benar terhadap suatu pola yang diberikan, walaupun pola tersebut merupakan pola yang serupa namun tidak sama. Pelatihan tersebut dilakukan berulang-ulang sehingga akurasi dalam sebuah peramalan menjadi lebih tinggi. Propagasi balik memiliki beberapa unit yang ada dalam satu atau lebih layar tersembunyi. Gambar 2.1 adalah arsitektur backpropagation dengan n buah masukan (ditambah sebuah bias), sebuah lapisan tersembunyi yang terdiri dari p unit (ditambah sebuah bias), serta m buah unit keluaran.



Gambar 2.1 Arsitektur Jaringan dengan Satu *Hidden Layer*

v_{ji} merupakan bobot garis dari unit masukan x_i ke unit layar tersembunyi z_j (v_{j0} merupakan bobot garis yang menghubungkan bias di unit masukan ke unit layar tersembunyi z_j). w_{kj} merupakan bobot dari unit layar tersembunyi z_j ke unit keluaran y_k (w_{k0} merupakan bobot nilai dari bias di layar tersembunyi ke unit keluaran z_k).

2.1.2 Fungsi Aktivasi

Fungsi aktivasi merupakan salah satu point terpenting dalam melakukan pelatihan sebuah jaringan saraf tiruan. Fungsi aktivasi digunakan untuk membantu sebuah sistem dalam mengenali sebuah pola dari sebuah data. Syarat dari fungsi aktivasi untuk algoritma propagasi balik yaitu bersifat kontinu, terdifferensial dengan mudah, dan bukan merupakan sebuah fungsi yang tidak turun. Terdapat 3 fungsi yang memenuhi persyaratan tersebut yaitu, fungsi logsig, tansig, dan purelin. Fungsi yang sering digunakan oleh kebanyakan peneliti merupakan fungsi logsig dan fungsi tansig.

a. Fungsi logsig (0,1)

Fungsi logsig atau sigmoid biner merupakan sebuah fungsi yang dapat membaca sebuah pola data yang berupa angka dan memiliki range angka mulai dari 0 sampai 1. Artinya data yang dapat dibaca oleh fungsi sigmoid ini merupakan data yang memiliki nilai $0 < x < 1$. Berikut merupakan fungsi sigmoid biner :

$$f(x) = \frac{1}{1+e^{-x}} \quad (2.1)$$

Dengan turunan

$$f'(x) = f(x)(1 - f(x)) \quad (2.2)$$

b. Fungsi tansig (-1,1)

Fungsi tansig atau sigmoid bipolar merupakan sebuah fungsi yang sama cara kerjanya dengan fungsi sigmoid biner namun pola data yang dapat dibaca berbeda. Pola data yang dapat dibaca oleh fungsi sigmoid bipolar adalah $-1 < x < 1$. Berikut merupakan fungsi sigmoid bipolar :

$$f(x) = \frac{2}{1+e^{-x}} - 1 \quad (2.3)$$

Dengan turunan

$$f'(x) = \frac{(1+f(x))(1-f(x))}{2} \quad (2.4)$$

Fungsi sigmoid diatas memiliki batasan dari sebuah data, maka jika ada sebuah data yang memiliki angka lebih dari batasan yang dimiliki oleh fungsi aktivasi maka diperlukan normalisasi sebuah data. Menurut buku yang dibuat oleh Jong Jek Siang yang membahas mengenai langkah menggunakan jaringan saraf tiruan menggunakan aplikasi matlab menyatakan bahwa dalam melakukan normalisasi data di perlukan sebuah rumus untuk membantu proses menormalisasikan data menjadi range dari sebuah fungsi yang digunakan. Untuk melakukan hal tersebut digunakan transformasi linier agar data berada di interval [0.1,0.9] untuk sigmoid biner. Transformasi linier yang digunakan yaitu :

$$x' = \frac{0.8(x-b)}{a-b} + 0.1 \quad (2.5)$$

Ket :

- a = Data Maximum
- b = Data Minimum
- x = Data Primer
- x' = Hasil Transformasi

2.1.3 Optimalisasi Arsitektur *Backpropagation*

Terdapat beberapa masalah yang dihadapi ketika pembuatan arsitektur *backpropagation*, seperti lamanya waktu iterasi yang dilakukan dan tidak diketahui jumlah *epoch* yang digunakan untuk mencapai tujuan yang ingin diraih. Maka dari itu terdapat beberapa hal yang perlu diperhatikan dalam mengoptimalkan arsitektur *backpropagation*.

1. Pemilihan Bobot dan Bias Awal

Bobot awal akan mempengaruhi sebuah jaringan seperti kecepatan konvergensinya dan untuk mengetahui apakah jaringan tersebut mencapai titik minimum lokal atau titik minimum global. Nilai pada bobot tidak boleh terlalu kecil dan tidak boleh terlalu besar karena akan menyebabkan hasil dari nilai turunan fungsinya menjadi sangat kecil. Terdapat sebuah cara dalam menentukan bobot yang di utarakan oleh Nguyen

dan Widrow pada tahun 1990 yaitu melakukan inisialisasi bobot dan bias ke unit tersembunyi.

$$\begin{aligned} \text{Misal} \quad n &= \text{jumlah unit masukan} \\ p &= \text{jumlah unit tersembunyi} \\ \beta &= \text{faktor skala} = 0.7^n \sqrt{p} \end{aligned}$$

Inisialisasi yang diusulkan oleh Nguyen Widrow adalah sebagai berikut :

- a. Inisialisasi semua bobot (v_{ji} (lama)) menjadi bilangan acak yang memiliki interval $[-0.5, 0.5]$.
 - b. Menghitung $\|v_{ji}\| = \sqrt{v_{j1}^2 + v_{j2}^2 + \dots + v_{jn}^2}$.
 - c. Bobot yang digunakan untuk inisialisasi adalah $v_{ji} = \frac{\beta v_{ji}(\text{lama})}{\|v_{ji}\|}$.
 - d. Bias yang digunakan untuk inisialisasi $v_{j0} =$ bilangan acak antara $-\beta$ dan β .
2. Jumlah Unit Tersembunyi

Dari hasil teoritis menunjukkan bahwa lapisan tersembunyi yang dimiliki propagasi balik sudah cukup baik untuk mengenali sebuah data dan hubungannya dari data masukan dengan data target. Namun dalam penambahan jumlah lapisan tersembunyi ternyata juga dapat mempermudah pelatihan jaringan propagasi balik. Jika sebuah jaringan memiliki dua atau lebih lapisan tersembunyi maka ada hal yang harus di revisi seperti dalam propagasi maju, data keluaran perlu dihitung untuk tiap lapisannya mulai dari lapisan tersembunyi dari bawah.

3. Jumlah Pola Pelatihan

Jumlah pola tidak memiliki kepastian dalam membuat sebuah jaringan menjadi sempurna. Jumlah pola yang diperlukan dipengaruhi oleh jumlah bobot dalam suatu jaringan serta tingkat akurasi yang diharapkan. Hal tersebut dapat dirumuskan seperti berikut :

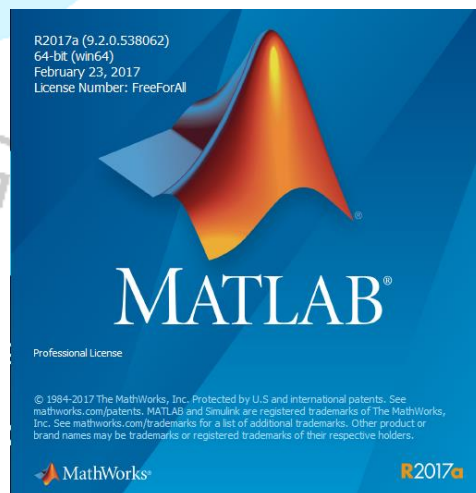
$$\text{Jumlah pola} = \text{jumlah bobot} / \text{tingkat akurasi}$$

4. Lama Iterasi

Penggunaan algoritma *backpropagation* memiliki tujuan utama yaitu mendapatkan keseimbangan antara pengenalan pola pelatihan dengan benar dan respon yang baik untuk pola yang sejenis namun berbeda. Sebuah jaringan dapat dilatih terus menerus sampai semua pola dapat dikenali, namun hal seperti itu tidak menjamin untuk dapat mengenali pola pengujian secara tepat. Jika dalam pelatihan semua kesalahan mengalami penurunan di setiap 10 *epochs* maka pelatihan dapat terus dijalankan. Sebaliknya jika kesalahan mengalami kenaikan maka pelatihan tidak perlu dilanjutkan.

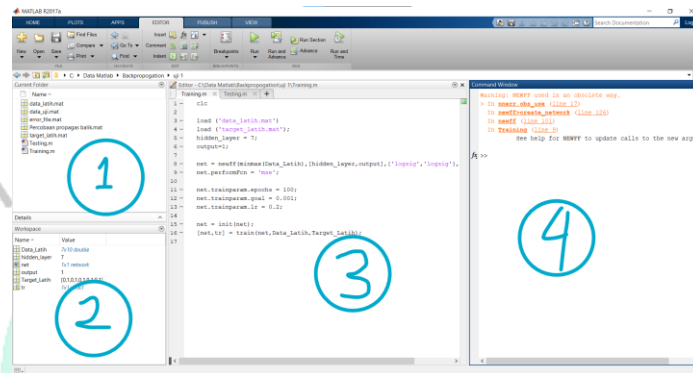
2.1.4 Aplikasi MATLAB

Aplikasi MATLAB merupakan sebuah platform yang sering digunakan karena kemampuan dalam mengolah angka dan bahasa pemrograman. Bahasa yang digunakan oleh matlab merupakan bahasa yang berbasis matriks. Dengan kemampuan yang dimiliki aplikasi ini, membuat banyak orang menggunakannya untuk menganalisis data, membuat algoritma, serta menciptakan pemodelan dan aplikasi. Pada penelitian ini menggunakan MATLAB versi R2017a.



Gambar 2.2 Software MATLAB versi R2017a.

MATLAB sudah memiliki akses dalam melakukan pemrograman dengan algoritma propagasi balik. Tidak hanya itu melainkan masih ada algoritma lainnya yang dapat digunakan melalui aplikasi tersebut. MATLAB memiliki 4 layar utama di dalamnya dan memiliki fungsinya masing-masing, seperti pada gambar 2.2.



Gambar 2.3 Tampilan Awal Aplikasi MATLAB Versi R2017a

Layar pertama merupakan layar yang menampilkan file serta folder dari sebuah project dan sebagai tempat penyimpanan sebuah file dan folder. Layar kedua merupakan layar kerja atau *workspace* biasanya menampilkan sebuah file yang sedang dikerjakan atau menyimpan sementara hasil dari *running* sebuah pemrograman. Layar ketiga merupakan tempat untuk memasukkan sebuah bahasa pemrograman untuk membuat algoritma serta memberikan sebuah perintah kedalam sebuah sistem yang nantinya akan menjadi pondasi untuk sebuah jaringan saraf tiruan. Layar keempat merupakan *command window* yang digunakan untuk menampilkan hasil dari bahasa pemrograman yang sudah disiapkan pada layar ketiga. Dan pada *command window* juga dapat digunakan untuk mencari informasi mengenai *tools* yang dapat digunakan untuk membuat sebuah bahasa pemrograman. Pada layar keempat ini dapat juga memberikan informasi jika bahasa pemrograman yang dimasukan tidak bekerja atau terjadi kesalahan dalam penyusunan bahasa pemrograman.

2.1.5 Komposisi Material Beton Normal

1. Agregat

Berdasarkan Standar Nasional Indonesia 03-2847-2002, yang termasuk agregat adalah material seperti pasir, kerikil, batu pecah yang pada umumnya digunakan sebagai bahan campuran pembuat beton normal. Terdapat dua jenis agregat yang biasa dipakai dalam sebuah campuran yaitu:

a. Agregat Kasar

Agregat kasar merupakan batuan kecil yang berasal dari pecahan batuan besar yang di kelola di industri dan pada umumnya memiliki ukuran butir sebesar 5 mm.

b. Agregat Halus

Agregat halus merupakan pasir yang berasal dari batuan pecah yang dikelola di industri dan memiliki ukuran butir antara 5 - 40 mm.

2. Air

Air merupakan salah satu bahan campuran pembuatan beton yang penting untuk menciptakan sebuah reaksi kimia dalam pembuatan beton. Dalam melakukan penakaran air yang akan digunakan sudah diatur dalam SNI 03-2847-2002, seperti:

a. Air yang dapat digunakan pada campuran merupakan air bersih dan bebas dari bahan campuran lain seperti air bekas mandi, air bekas oli dan sebagainya.

b. Air yang dapat digunakan pada campuran merupakan air yang tidak boleh mengandung ion klorida yang membahayakan beton.

c. Air yang dapat digunakan pada campuran beton merupakan air yang memiliki kandungan lumpur yang rendah dengan batasan tidak lebih dari 5 persen.

3. *Portland Cement* (PC)

Portland cement atau semen merupakan bahan pengikat dalam sebuah campuran beton yang akan bereaksi ketika diberikan air. Semen merupakan salah satu bahan utama yang sering digunakan dalam pembuatan beton mutu tinggi maupun mutu rendah.

2.2 Penelitian Terdahulu

Stefanus Santosa (2020) bersama rekan lainnya melakukan studi tentang penggunaan *H2O'S Deep Learning* untuk Pemodelan Prediksi Kuat Tekan Beton Umur Muda dengan menggunakan metode *NN (Neural Network)*. Pemodelan prediksi kuat tekan beton melibatkan *Machine Learning* yang memiliki kemampuan seperti manusia dalam memecahkan masalah dengan cara melakukan pelatihan. Data yang digunakan dalam penelitian ini adalah data kuat tekan rencana dari desain campuran beton yang sudah diuji coba serta memperoleh nilai kuat tekan beton dalam satuan Mpa. Komposisi campuran beton yang divariasikan berdasarkan beberapa variabel serta perbandingannya seperti, semen, agregat kasar, agregat halus, *fly ash*, *superplastizier* dan umur beton (3, 7, 14, 28 hari).

Pemodelan terbaik prediksi kuat tekan beton dengan *H2O'S Deep Learning* menggunakan arsitektur 2 lapisan tersembunyi - 50 jaringan tersembunyi dengan data agregat kasar sebesar 0.6 untuk pelatihan dan 0.4 untuk pengujian. Fungsi aktivasi yang digunakan merupakan fungsi *rectified linear unit (ReLU)* dengan nilai *root mean square error (RSME)* sebesar 6.801. Penelitian ini melakukan perbandingan antar jumlah lapisan tersembunyi untuk mendapatkan hasil RSME yang terendah guna mencapai akurasi dalam memprediksi.

Adapun penelitian yang menggunakan JST lainnya seperti penelitian yang dilakukan oleh Nazla Nurmila, Aris Sugiharto, dan Eko Adi Sarwoko yang berjudul *Algoritma Back Propagation Neural Network Untuk Pengenalan Pola Karakter Huruf Jawa*. Peneliti menggunakan bantuan software MATLAB untuk membuat pemodelan. Penelitian ini menghasilkan bahwa cara kerja dalam mengenali sebuah pola dari sebuah

data sangat baik sehingga menghasilkan akurasi sebesar 99.56% untuk pengenalan pola berdasarkan data sampel pelatihan, 61.36% untuk data yang di luar dari sampel data pelatihan dan 75% untuk data campuran.

Penelitian mengenai kuat tekan beton dengan menggunakan *backpropagation* juga dilakukan oleh Faezehossadat Khademi dan Sayed Mohammadmehdi Jamal pada bulan Mei 2016, yang berjudul *Predicting the 28 Days Compressive Strength of Concrete Using Artificial Neural Network*. Peneliti menggunakan bantuan software MATLAB untuk membuat jaringan arsitektur dengan variabel-variabel penelitian meliputi campuran beton dan kuat tekan beton. Dengan hasil penelitian menyebutkan bahwa analisis JST menunjukkan hubungan baik antara variabel masukan dan variabel keluaran. Dapat disimpulkan bahwa ANN dapat menyajikan akurasi yang baik untuk memprediksi kuat tekan beton 28 hari.

Selain itu juga terdapat artikel yang dibuat oleh Chia-Ju Lin dan Nan-Jing Wu dengan judul artikel An ANN Model for Predicting the Compressive Strength of Concrete. Peneliti ini menggunakan algoritma *backpropagation* karena memiliki keunggulan dari algoritma lain yaitu lapisan tersembunyi. Variabel yang digunakan pada penelitian kali ini pun sama seperti seperti lainnya yaitu campuran beton dan kuat tekan beton. Database proporsi campuran beton digunakan untuk membangun model prediksi JST untuk kuat tekan beton. Database terbagi menjadi 2 set, yaitu set pelatihan dan set pengujian dengan perbandingannya adalah 15% untuk set pengujian dan 85% untuk set pelatihan. Dari hasil penelitiannya menyatakan bahwa model prediksi yang dibentuk telah bekerja dengan sangat baik dan dapat dilakukan eksperimen lebih lanjut. Untuk hasil prediksi kuat tekan beton normal memiliki koefisien efisiensi (CE) sebesar 0.975 dan memiliki relatif *root mean square error* adalah 15.8%, menandakan bahwa prediksi memiliki hasil yang agak berdekatan dengan hasil sebenarnya.