

# BAB I

## PENDAHULUAN

### A. LATAR BELAKANG

Perkembangan teknologi yang pesat semakin maju para ahli mencoba untuk menggantikan sistem otak manusia kedalam sistem komputer. Dengan adanya kinerja dari sistem komputer manusia dapat lebih cepat untuk teliti dalam mengerjakan tugas dan akurat jika dibandingkan dengan kinerja manusia untuk mendorong percepatan pengembangan teknologi AI (*Artificial Intelligence*).

Jaringan syaraf tiruan (*artificial neural network*) adalah model komputasi yang terinspirasi secara biologis, jaringan syaraf tiruan terdiri dari beberapa elemen pengolahan (*neuron*) dan ada hubungan antara *neuron*. *Neuron-neuron* tersebut akan mentransformasikan informasi yang diterima oleh *neuron* satu menuju *neuron* lainnya, hubungan ini disebut dengan bobot. Deboek dan Kohonen menggambarkan jaringan syaraf tiruan (NN) sebagai kumpulan teknik matematika yang dapat digunakan untuk pemrosesan sinyal, peramalan dan pengelompokan dan disebut sebagai teknik *regresi paralel non linier, multilapis* (Shanmuganathan, 2016).

Salah satu penemuan model Jaringan Syaraf Tiruan yang diminati oleh banyak orang adalah *backpropagation*. *Backpropagation* merupakan algoritma pembelajaran yang terwarisi dan biasanya digunakan oleh *perceptron* dengan banyak lapisan untuk mengubah bobot-bobot yang

terhubung dengan *neuron-neuron* yang ada pada lapisan tersembunyinya. Algoritma *backpropagation error output* untuk mengubah nilai-nilai bobotnya dalam arah mundur (*backward*). Tahap perambatan maju (*forward propagation*) harus dikerjakan terlebih dahulu mendapatkan nilai *error* tersebut (Kusumadewi, 2004).

Pada *backpropagation* terdapat pengolahan elemen yang membuat perhitungan berdasarkan pada jumlah masukan (*input*). Sebuah kelompok pengolahan elemen disebut *layer* atau lapisan dalam jaringan. Lapisan pertama adalah *input* dan yang terakhir adalah *ouput*. Lapisan diantara lapisan *input* dan *output* disebut dengan lapisan tersembunyi atau *hidden layer* (Hermawan, 2006).

*Hidden layer* berfungsi untuk membantu proses pelatihan. Semakin banyak *neuron* dalam *hidden layer* yang digunakan, maka semakin bagus dan semakin tepat *output* yang dihasilkan.

Pada penelitian yang telah dilakukan sebelumnya pengujian terhadap algoritma pelatihan yang terdapat dalam jaringan *backpropogation* yang sejumlah 12 algoritma yaitu algoritma *Fletcher-Reeves Update (Traincgf)*, *Polak-Ribiëre (Traincgp)*, *Powell-Beale Restarts (Traincgb)*, dan *Scaled Conjugate Gradient (Trainscg)*, *Gradient Descent (Traingd)*, *Gradient Descent dengan Adaptive Learning Rate (Trainгда)*, *Gradient Descent dengan Momentum (Traingdm)*, *Gradient Descent dengan Momentum dan Adaptive Learning Rate (Traingdx)*, *Resilent Backpropagation (Trainrp)*, *BFGS (Trainbfg)*, *One Step Secant (Trainoss)*, *Levenberg-Marquardt (Trainlm)*. Fungsi dari pengujian ini untuk

mendapatkan algoritma pelatihan yang paling optimal ditinjau dari *error* yang dihasilkan berdasarkan pengenalan pola data masukan (Mustafidah dan Suwarsito, 2015b).

Dengan demikian diperlukan penelitian untuk menentukan algoritma pelatihan jaringan *backpropagation* yang paling optimal di tinjau dari *Error* jaringan yang di hasilkan dengan *hidden layer* jaringan sebagai salah satu parameternya. Hasil akhir yang diharapkan adalah bahwa penelitian ini dapat menjadi rujukan dalam membantu penelitian selanjutnya yang terkait dengan algoritma jaringan syaraf tiruan.

## **B. PERUMUSAN MASALAH**

Rumusan permasalahan dalam penelitian ini adalah belum adanya informasi tingkat *error* pada algoritma pelatihan jaringan *backpropagation* berdasarkan laju pemahaman pada model *neuron* 10-12-1 yaitu dengan 10 *neuron input*, 12 *neuron* dalam 1 *hidden layer*, dan 1 *neuron output*.

## **C. BATASAN MASALAH**

Batasan masalah dalam penelitian ini adalah Jaringan Syaraf tiruan *backpropagation* menggunakan satu *layer* yang tersembunyi, dengan 10 *neuron input*, 12 *neuron* dalam 1 lapisan *hidden layer*, 1 *neuron output* dengan *learning rate* 0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.