# **BAB IV**

# **ANALISIS DAN IMPLEMENTASI**

Pada bab ini membahas bagian analisis dan implementasi yang meliputi sumber data, proses penerapan metode Jaringan Syaraf Tiruan (JST) pada bahasan pemograman Python.

## **Sumber Data**

Untuk melanjutkan ke tahap prediksi, maka akan diperlukan sebuah data dan informasi yang mencukupi kebutuhan. Pada penelitian ini, sumber data yang digunakan adalah data target penjualan air PDAM di Kota Padang, yang mana data target penjualan ini data yang sah dari tahun 2017 sampai tahun 2019. Pada data *training* sebagai *input* untuk proses melakukan prediksi pada data target penjualan air yang memiliki *variable* seperti sosial umum 1, sosial umum 2, rumah tangga 1, rumah tangga 2, rumah tangga 3, rumah tangga 4, instansi pemerintah 1, instansi pemerintah 2, niaga 1, niaga 2, dan niaga 3. Pada lampiran A tabel 1 akan menunjukan data target penjualan air PDAM pada tahun 2017 sampai 2019.

1. ***Pre-Pocessing* Data**

*Pre-processing* merupakan tahap salah satu tahap yang sangat penting di data mining. Data yang digunakan pada tahap ini adalah data yang tidak selalu ideal pada setiap kondisinya, tapi terkadang terdapat berbagai permasalahan yang dapat menganggu hasil dari proses mining itu sendiri seperti *missing value,* data redundant, ataupun format data yang tidak sesuai dengan sistem. Pada *pre-pocessing* data ini memiliki 4 teknik yang digunakan dalam *pre­-processing* data (Han, Kamber, & Pei, 2011), diantaranya:

1. Data *Cleaning*

Data *cleaning* merupakan proses untuk menghilangkan atau membersihkan data-data yang salah atau data yang *noise* dan data yang tidak konsisten. Dan memperbaiki data yang kacau dan memeriksa data yang tidak konsisten. Pada penelitian ini, penulis menggunakan menghilangkan dan menambah beberapa data yang masih kosong atau data yang belum konsisten pada dataset. Pada tabel 4.1 dibawah ini dapat dilihat data yang belum dan sesudah di *cleaning* pada tabel 4.2*.*

Tabel 4.1 Data yang belum di cleaning

|  |
| --- |
|  |

Tabel 4.2 Hasil Data Setelah Cleaning Data

|  |
| --- |
|  |

1. Data Integrasi

Data Integrasi merupakan suatu proses dalam menggabungkan data dari beberapa sumber data, seperti *database*, data *cube*, ataupun file ke dalam penyimpanan data yang telah di sesuaikan. Pada tahap ini bertujuan untuk menggabungkan data dari beberapa sumber dan peneliti memberi penamaan pada data pelanggan sosial umum dengan variabel X1, data rumah tangga 1 dengan variabel X2, data rumah tangga 2 dengan variabel X2, data rumah tangga 3 dengan variabel X3, data niaga dengan variabel X4, dan data target dengan variabel Y. Adapun hasil sebelum dan sesudah integrasi data PDAM dengan *excel* dapat dilihat pada tabel 4.3 dan tabel 4.4 sebagai berikut:

Tabel 4.3 Sebelum Data PDAM di Integrasi

|  |
| --- |
|  |

Tabel 4.4 Hasil Setelah Data PDAM di Integrasi

|  |
| --- |
|  |

1. Data Transformasi

Beberapa teknik data mining membutuhkan format data yang khusus agar dapat di aplikasikan. Proses ini sering juga disebut dengan *binning*, yang mana dilakukan pemilihan data yang diperlukan oleh teknik data mining yang di pakai. Data yang sudah dikumpulkan akan melakukan proses normalisasi bertujuan untuk merubah data menjadi data yang sederhana dan efektif. Pada penelitian ini diterapkan rangenya dari 0-1. Pada tabel 4.5 dan tabel 4.6 menunjukkan data sebelum dan sesudah dilakukan transformasi.

Tabel 4.5 Data sebelum transformasi

|  |
| --- |
|  |

Tabel 4.6 Data sesudah di Transformasi

|  |
| --- |
|  |

Data primer yang didapatkan dari PDAM kota Padang Sumatera Barat terlebih dahulu diproses sebelum data primer PDAM diolah menggunakan bahasa pemograman Python di *Jupyter Notebook*. *Jupyter notebook* merupakan sebuah aplikasi *web open-source* yang memungkinkan dalam membuat dan berbagi dokumen interaktif yang berisi kode *live*, persamaan, visualisasi, dan teks naratif. Data primer PDAM terlebih dahulu dimasukkan ke dalam *Microsoft Excel* untuk di kelompokkan yang mana termasuk data input dan data target pada PDAM, lalu data primer PDAM di *input* ke dalam *jupyter notebook* untuk melakukan normalisasi data PDAM. Pada tabel 4.7 dibawah ini akan menunjukkan proses data primer PDAM diolah dengan menggunakan *jupyter notebook*.

Tabel 4.7 Data PDAM

|  |
| --- |
|  |

Setelah dikelompokkan data PDAM tersebut, kemudian di *input*kan ke dalam *jupyter notebook* dengan menggunakan beberapa *library* yang sudah ada pada *jupyter notebook.* Setelah di *input* kan, data tersebut akan melakukan tahap normasilasi, yang mana dilakukan normalisasi untuk menghilangkan atau mengurangi nilai yang redudansi pada data. Tabel 4.8 dibawah akan menunjukkan hasil normalisasi yang sudah dilakukan dengan *jupyter notebook.*

Tabel 4.8 Hasil Normalisasi

|  |
| --- |
|  |

## **Perancangan Arsitektur Model JST *Backpropagation***

Pada perancangan model JST *Backpropagation* menggunakan Anaconda Navigator yaitu *Jupyter Notebook* dan juga menggunakan Bahasa pemograman Python. Perancangan ini menggunakan beberapa *library* seperti Pandas, Numpy, dan Matplotlip.pyplot, *sklearn preprocessing* dan *standar scaler* yang mana digunakan untuk mengimport data primer pada PDAM.

Pada *jupyter notebook* terdapat *library* yang sudah disediakan untuk mempermudah dalam menggunakan *jupyter notebook*, seperti *library* pandas digunakan untuk membuat tabel, mengubah dimensi pada data dan mengecek data yang sudah di *impor, library* numpy digunakan dalam melakukan perhitungan dalam aljabar linear terutama pada operasi vektor, *library* matplotlib pada python digunakan untuk memvisualisasi data secara 2D atau 3D untuk menghasilkan gambar yang berkualitas.

|  |
| --- |
| import pandas as pd  import numpy as np  import matplotlib.pyplot as plt  from sklearn import preprocessing  from sklearn.preprocessing import MinMaxScaler |

Untuk langkah selanjutnya ada terdapat 3 tahap untuk mentrainingkan data PDAM gunanya untuk mengatasi masalah yang ada pada data, 3 tahapan untuk proses mentraining yaitu:

1. *Forward Pass*

Tahap *forward pass* ini melakukan proses untuk meneruskan input ke dalam model dan mengalikannya dengan sebuah bobot (w) dan menambahkan bias (b) yang ada pada setiap layer sehingga pada akhirnya kita dapat menggunakan untuk menemukan nilai *output*.

1. *Loss Calculate*

Pada bagian ini nilai *weight* digunakan untuk tiap koneksi neuron akan diupdate sehingga output value dapat dilakukan untuk mendekati target value. Jika pada seingle layer percepton kita menggunakan Delta Rule gunanya untuk mengevaluasi nilai error, maka pada multi layer percepton kita menggunakan *backpropagation*. *Backpropagation* menghitung gradien dari *loss function* untuk tiap ‘*weight*’ menggunakan *chain rule* yang dapat menghitung gradien satu layer pada suatu waktu saat iterasi mundur dari layer terakhir untuk menghindari *redundant calculation.*

1. *Backward Pass*

Pada tahap ini error yang didapatkan dari proses sebelumnya digunakan untuk mengoreksi ‘*weight*’ menggunakan *backpropagation. Backpropagation* akan menghitung gradien *loss function* untuk tiap *weight* yang digunakan pada *output layer* (Vjk) begitu pila *weight* pada *hidden layer* (Wij). Kemudian hitung nilai bias yang kemudian akan digunakan untuk memperbaharui nilai dan dikirim ke neuron pada lapisan sebelumnya.

Prediksi yang dilakukan pada data PDAM memerlukan data yang banyak agar mendapatkan hasil prediksi yang tepat. Metode yang digunakan untuk memprediksi data PDAM yaitu metode JST (jaringan syaraf tiruan) *Backpropagation*. Metode ini digunakan untuk jaringan dengan banyak lapisan (*multilayer*) dengan menggunakan metode pembelajaran yang (*supervised learning*). Metode pembelajaran *supervised learning* itu sendiri ialah algoritma yang dilatih terlebih dahulu agar dapat melakukan prediksi maupun klasifikasi. *Supervised learning* memiliki data masukan dan data keluaran yang digunakan untuk mengoreksi bobot JST agar JST dapat menghasilkan jawaban yang semirip mungkin dengan jawaban yang benar dan telah diketahui oleh JST.

Pada jumlah *neuron* pada *hidden layer* ditentukan dengan kurang dari dua kali jumlah *neuron input*. Penentuan jumlah *neuron* yang dilakukan jika terlalu sedikit akan mengakibatkan *underfitting* pada jumlah *neuron* yaitu yang mana jaringan kurang dapat mendeteksi sinyal atau pola yang terdapat dalam set data. Dan jika jumlah *neuron* terlalu banyak maka akan mangakibatkan *overfiting* pada jumlah *neuron* dan jumlah informasi dalam *training* set akan terbatas dan juga tidak cukup untuk melatih *neuron* pada *hidden layer* tersebut (Costa & Santosa, 2014).

Ada terdapat rancangan rincian model prediksi yang mana terdapat beberapa parameter yang digunakan pada perancangan ini. Berikut tabel 4.9 rincian model prediksi yang dapat dilihat di bawah ini:

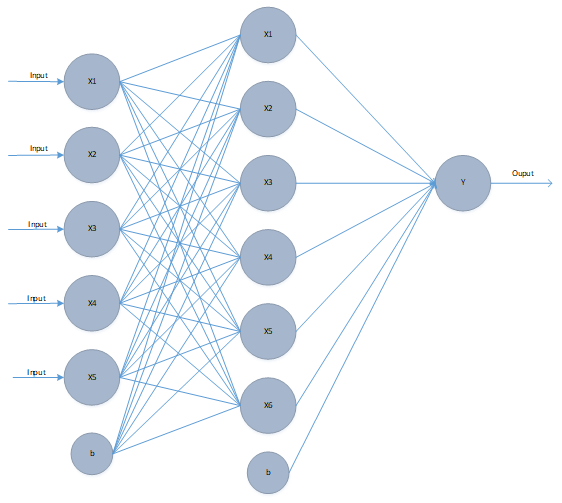
Tabel 4. 9 Rincian Rancangan Model Prediksi

|  |  |  |
| --- | --- | --- |
| Parameter | Jumlah | Keterangan |
| *Input layer* | 11 *neuron* | * Target penjualan air pada sosial umum 1 dan 2 * Target penjualan air pada rumah tangga 1,2,3,4 * Target penjualan air pada instansi pemerintah 1 dan 2 * Target penjualan air pada niaga 1,2 dan 3 |
| *Hidden layer* | *Trial and error* | 5 *neuron* |
| *Output layer* | 1 *neuron* | Prediksi target penjualan air PDAM |
| *Learning rate* | *Trial and error* | 0,01 |
| *Epoch* | *Trial and error* | 15000 |
| Fungsi aktivasi | *Trial and error* | *Sigmoid biner and linear* |

Keterangan:

1. Pada penelitian ini terdapat 2 variabel, yaitu data input dan data target. Yang mana variabel data input itu adalah data sosial umum 1 dan 2, rumah tangga1, rumah tangga 2, rumah tangga 3, rumah tangga 4, instansi pemerintah 1, instansi pemerintah 2, niaga 1 dan niaga 2. Sedangkan variabel data target adalah target hasil dari penjualan air perbulan berdasarkan jenis golongan pelanggan air PDAM dari tahun 2017 sampai 2019.
2. Arsitektur *layer* yang digunakan pada penelitian ini adalah *multilayer* yang mana terdiri dari *input layer, hidden layer* dan *output layer*. Dan pada *input layer* terdapat 11 variabel yaitu sosial umum 1 dan 2, rumah tangga 1, rumah tangga 2, rumah tangga 3, rumah tangga 4, instansi pemerintah 1 dan 2, niaga 1 dan niaga 2. Pada *hidden layer* terdapat jumlah *neuron* yang tersembunyi yang mana jumlah *neuron* tersebut harus kurang dari dua kali *input layer*(Panchal & Panchal, 2014). Menurut teori Heaton (2011) aturan yang digunakan saat menentukan jumlah *hidden layer* adalah:
3. Jumlah *hidden layer neuron* harus berada diantara ukuran *input layer* dan *output layer*.
4. Jumlah pada *hidden layer* harus 2/3 dari ukuran *input layer* dan juga *output layer*.
5. Dan jumlah pada *hidden layer* kurang dari dua kali jumlah pada *input layer*.
6. Pada perancangan model JST ini fungsi aktivasi pada lapisan tersembunyi yang digunakan proses pelatihan *sigmoid activation function* dan fungsi aktivasi yang digunakan pada lapisan keluarannya adalah *linear activation function*. Karena *sigmoid biner* dan *linear* ini dapat digunakan untuk menampilkan hasil perbandingan target penjualan air pada PDAM.
7. Parameter *epoch* (iterasi) sebagai perulangan untuk pemberhentian pada proses pelatihan yang dilakukan. Pada saat iterasi, apabila jumlah iterasi telah mencapai jumlah maksimum maka iterasi pada pelatihan akan berhenti, jika maksimum *epoch* tinggi maka pelatihan memerlukan waktu yang lama dan ketika *epoch* rendah maka pelatihan yang dilakukan tidak memerlukan waktu yang lama dan hasil akurasi kurang baik (Febrina et al., 2013).
8. *Learning rate* digunakan untuk *hyperparameter* yang mengontrol seberapa banyak model harus diubah sebagai respons terhadap estimasi *error* setiap kali bobot model diperbarui. Memilih kecepatan pembelajaran cukup menantang karena nilai yang terlalu kecil dapat mengakibatkan proses pelatihan yang lama yang dapat macet, sedangkan nilai yang terlalu besar dapat mengakibatkan pembelajaran kumpulan bobot yang kurang optimal terlalu cepat atau proses pelatihan yang tidak stabil.

Pada gambar 4.1 dibawah ini, merupakan hasil dari pemodelan yang terpilih dari arsitektur *backpropagation.*



Gambar 4. 1 Arsitektur Backpropagation

Pada bagian *input layer* terdapat data target penjualan air PDAM berdasarkan golongan pelanggan, seperti sosial umum 1 dan sosial umum 2, rumah tangga 1, rumah tangga 2, rumah tangga 3, rumah tangga 4, instansi pemerintah 1, instansi pemerintah 2, niaga 1 dan niaga 2. Dari *input layer* ke *hidden layer* terdapat bobot yang di inisialkan dengan V, dan dari nilai bobot W yaitu nilai bobot dari *hidden layer* ke *output layer*. Bagian akhir terdapat bagian *output layer* yang menjadi hasil dari pemodelan arsitektur *backpropagation*.

## **Pengujian**

Pada bagian ini melakukan tahap pengujian untuk melakukan penghitungan akurasi dan *error* prediksi pada PDAM Kota Padang.



### **Hasil Pengujian Menggunakan JST *Backpropagation***

Pada tahap ini data yang diperoleh dari PDAM Kota Padang sudah terdapat beberapa variabel yaitu data sosial umum 1 dan 2, rumah tangga 1, rumah tangga 2, rumah tangga 3, rumah tangga 4, instansi pemerintah 1, instansi pemerintah 2, niaga 1 dan niaga 2. Pengujian yang dilakukan menggunakan Bahasa pemograman pyhton dengan *jupyter notebook* agar untuk mendapatkan hasil prediksi yang sesuai dengan nilai *input* pada setiap parameter yang telah ditentukan paa rancangan model prediksi sebelumnya.

Fungsi aktivasi yang digunakan untuk mengolah data PDAM ini adalah fungsi aktivasi *sigmoid binner* dan *linear*. Untuk bagian *input* fungsi aktivasi yang digunakan adalah fungsi *sigmoid binner* dan pada bagian *output*nya menggunakan fungsi *linear*. Hasil pengujian parameter prediksi pada data PDAM menggunakan bahasa pemograman python dan JST *backpropagation* mendapatkan hasil pengujian sistem terhadap rancangan model dengan parameter yang telah ditentukan, maka diperoleh parameter yang menghasilkan nilai terkecil dan optimal yaitu dengan menggunakan *epoch* 15000 mendapatkan bentuk model yang terbaik dengan nilai akurasi 99,62013%, *learning rate* sebesar 0,01 dengan nilai iterasi berhenti pada iterasi ke-8800 dengan jumlah *error* 0,0001016, hasil pengujiannya dapat dilihat pada tabel 4.10 dibawah:

Tabel 4. 10 Hasil Pengujian Parameter

|  |  |  |  |
| --- | --- | --- | --- |
| Epoch | *Learning rate* | *Error* | Akurasi |
| 15000 | 0,01 | 0,0001016 | 99,62013% |

### **Penerapan Pelatihan Data *Training***

Pada pengujian ini dilakukan dengan mengukur tingkat keakurasian pada sistem berdasarkan model yang yang telah diimplementasikan sebelumnya. Pengujian ini dilakukan menggunakan data *training* yang dipergunakan saat proses *training* data, porses dari program dapat dilihat pada tabel 4.11 dan hasil pengujian sistem data *training* dapat dilihat pada tabel 4.12 dan tabel 4.13 dibawah ini:

Tabel 4.11 Proses dari Mentraining Data

|  |
| --- |
|  |

Tabel Tabel 4. 12 Training Pada Hidden layer

|  |  |  |
| --- | --- | --- |
| Tahun | Bulan | Hasil *Training* |
| 2017 | Januari | 0.5177531 |
| Februari | 0.4999168 |
| Maret | 0.40796322 |
| April | 0.6582525 |
| Mei | 0.5557192 |
| Juni | 0.5243352 |
| Juli | 0.6247556 |
| Agustus | 0.59990114 |
| September | 0.7223645 |
| Oktober | 0.47704798 |
| November | 0.7003395 |
| Desember | 0.6120584 |

Tabel 4. 13 Training Pada Output Layer

|  |  |  |
| --- | --- | --- |
| Tahun | Bulan | Hasil *Training* |
| 2017 | Januari | 0.28251076 |
| Februari | 0.23755956 |
| Maret | 0.00581777 |
| April | 0.6365979 |
| Mei | 0.37819314 |
| Juni | 0.29909897 |
| Juli | 0.55217886 |
| Agustus | 0.48954046 |
| September | 0.7981732 |
| Oktober | 0.17992544 |
| November | 0.74266577 |
| Desember | 0.5201793 |

### **Penerapan Pelatihan Data *Testing***

Pengujian selanjutnya dilakukan dengan data *testing* yang berupa data pada PDAM yang menjadi target penjualan air pada PDAM. Pada tabel 4.14 menampilkan proses dari program untuk mentesting data, tabel 4.15 menampilkan hasil data *testing* pada *hidden layer*, dan tabel 4.16 menampilkan hasil data *testing* pada *output layer* yang menggunakan data dari PDAM, seperti berikut:

Tabel 4.14 Proses dari Mentesting Data

|  |
| --- |
|  |

Tabel 4. 15 Hasil Data Testing Pada Hidden Layer

|  |  |  |
| --- | --- | --- |
| Tahun | Bulan | Hasil *Testing* |
| 2017 | Januari | 0.32696986 |
| Februari | 0.31536758 |
| Maret | -0.17477113 |
| April | 0.33216769 |
| Mei | 0.48084545 |
| Juni | 1.155466 |
| Juli | -0.5308348 |
| Agustus | 0.86386895 |
| September | 0.3605919 |
| Oktober | 0.45985103 |
| November | 1.3475832 |
| Desember | 0.581107 |

Tabel 4. 16 Hasil Testing Pada Output Layer

|  |  |  |
| --- | --- | --- |
| Tahun | Bulan | Hasil *Testing* |
| 2017  2017 | Januari | 0.44196403 |
| Februari | 0.43483913 |
| Maret | 0.12793565 |
| April | 0.44515145 |
| Mei | 0.5350243 |
| Juni | 0.8943069 |
| Juli | -0.08904445 |
| Agustus | 0.7505549 |
| September | 0.46253335 |
| Oktober | 0.5225022 |
| November | 0.9780444 |
| Desember | 0.5939332 |

### **Pengujian Perhitungan Manual**

Setelah mendapatkan model dari hasil dari *training* dan *testing* data menggunakan bahasa pemograman *python* dengan *jupyter notebook*, model tersebut akan diuji pada data *training* menggunakan metode *backpropagation*. Pengujian model ini akan dilakukan dengan perhitungan manual, yang mana terdapat nilai bobot dari *input layer* ke *hidden layer*, nilai bobot dari *hidden layer* ke *output layer,* dapat dilihat pada tabel 4.17 dan tabel 4.18 dibawah ini:

Tabel 4. 17 Bobot dari Input Layer ke Hidden Layer

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Bobot | X1 | X2 | X3 | X4 | X5 | X6 | X7 | X8 | X9 | X10 | X11 |
| 0.32032 | 0.09157 | 0.20364 | 0.20793 | 0.34731 | 0.33290 | 0.12150 | 0.13654 | 0.25012 | 0.07762 | 0.13148 |

Tabel 4. 18 Bobot dari Hidden Layer ke Output Layer

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Bobot | V1 | V2 | V3 | V4 | V5 |
| 0.50404 | 0.50404 | 0.50404 | 0.50404 | 0.50404 |

Setelah melakukan pelatihan pada data dengan menggunakan *jupyter notebook* dan memperoleh hasil dari pelatihan tersebut. Selanjutnya menguji hasil dengan menggunakan rumus *backprpopagation* dan nilai *output* yang didapatkan berdasarkan metode *backpropagation*. Prosedur pengujian proses *backpropagation* sebagai berikut:

**Langkah 0** : Inisialisasi bobot-bobot, konstanta laju pelatihan, toleransi *error* atau nilai bobot.

**Langkah 1** : Selama kondisi berhenti belum tercapai, maka lakukan langkah ke-2 hingga langkah ke-9.

**Langkah 2** : Untuk setiap pasangan pola pelatihan, lakukan langkah ke-3 sampai langkah ke-8.

**Tahap I: Umpan Maju (*Feedforward*)**

**Langkah 3** : Setiap unit *input* (dari unit ke-1 hingga unit ke-n pada lapisan *input*) mengirimkan sinyal *input* ke setiap input yang berada pada lapisan tersembunyi.

**Langkah 4** : Masing-masing unit di lapisan tersembunyi z\_j (dari uni ke -1 hingga unit ke-p) dikalikan dengan bobotnya dan dijumlahkan serta ditambahkan dengan biasnya. Lalu hitung fungsi aktivasi sebagai berikut:

*j=1,*

*j=2,*

*j=3,*

*j=4,*

*j=5,*

Fungsi Aktivasi

*j=1,*

*j=2,*

*j=3,*

*j=4,*

*j=5,*

**Langkah 5** : Masing-masing unit ouput (yk, k=1,2,3,…..n) dikalikan dengan bobot dan dijumlahkan serta ditambahkan dengan indeks bias outputnya.

++++ -1.02233

+++

+++

**Tahap II: Umpan Mundur (*backward propagation*)**

**Langkah 6 :** Hitung faktor berdasarkan kesalahan setiap unit keluaran Yk

= () (

= () (

Hitung perubahan bobot *Wjk* (digunakan dalam mengubah bobot *Wjk*) dengan (*Learning Rate*) α = 0.01 dengan persamaan seperti berikut:

= 0.01\* () \*

= -0.00051

= 0.01\* () \*

= -0.00035

= 0.01\* () \*

= -0.00043

= 0.01\* () \*

= -0.00043

= 0.01\* () \*

= -0.00051

**Langkah 7 :** Hitung faktor *error* δ unit tersembunyi (*hidden layer*) berdasarkan *error* di setiap unit tersembunyi (*hidden layer*)

Kemudian hitung faktor *error* δ di unit tersembunyi (*hidden layer*) dengan persamaan sebagai berikut:

Hitung koreksi nilai bobot yang kemudian digunakan untuk memperbaharui (digunakan untuk perbaikan bobot ):

(dimana α 0.01)

X1 = 0.22841

X2 = 0.52258

X3 = 0.09075

0.09075

X4 = 0.43976

X5 = 0.49285

Langkah selanjutnya hitung koreksi nilai Bias dari unit tersembunyi (*hidden layer*) berikut rumus persamaannya:

**Tahap III: Pengupdatean Bobot dan Bias**

**Langkah 8 :** Hitung semua perubahan bobot. Langkah selanjutnya adalah menghitung nilai bobot unit tersembunyi (*hidden*) dengan persamaan:

= +

= +

= +

= +

Hitung perubahan nilai bobot garis yang menuju ke unit keluaran (*output*) dnegan menggunakan rumus persamaan seperti berikut:

Kemudian hitung perubahan nilai bias dari unit tersembunyi (*hidden layer*) ke *output* dengan persamaan berikut:

Pada tabel 4.19 dan 4.20 dibawah menampilkan proses pengitungan pada program yang dijalankan di *jupyter notebook* menggunakan bahasa pemograman pyhton dan hasil dari perhitungan.

Tabel 4.19 Perhitungan dengan python menggunakan jupyter notebook

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

Tabel 4. 20 Hasil dari perhitungan dengan program

|  |
| --- |
|  |

### ***Post-Prepocessing* Data**

Setelah proses pelatihan selesai, maka didapatkan data keluaran (*output*) yang sudah didapatkan dari pelatihan jaringan atau hasil prediksi dalam bentuk proses normalisasi atau transformasi, sehingga mesti dilakukan proses denormalisasi atau biasa disebut dengan *post-prepocessing* yaitu suatu proses yang mana nilai *output* menghasilkan nilai dengan rata-rata sama dengan nol, maka dalam proses ini akan mengembalikan nilai sesuai dengan data awal atau data yang asli.

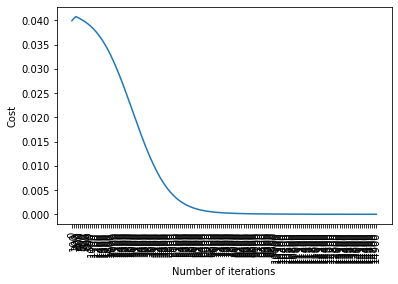
Proses denormalisasi bertujuan untuk mengembalikan nilai hasil keluaran data yang masih dalam bentuk normalisasi ke dalam bentuk data yang sebenarnya. Setelah seluruh proses prediksi dilakukan dengan bahasa pemograman python dengan menggunakan metode JST *Backpropagation* sehingga mendapatkan hasil akurasi dan tingkat kesalahan (*error*) yang optimal. Hasil denormalisasi dan hasil prediksi target penjualan air pada PDAM untuk tahun 2017 dapat ditunjukkan pada tabel 4.21 berikut.

Tabel Tabel 4.21 Hasil Sebelum dan Sesudah Denormalisasi Prediksi Target Penjualan Air PDAM

|  |  |  |  |
| --- | --- | --- | --- |
| Tahun | Bulan | Normalisasi | Denormalisasi |
| 2017 | Januari | 0.28269402 | 1496597.8 |
| Februari | 0.2373421 | 1494290.8 |
| Maret | 0.2674504 | 1394912.0 |
| April | 0.63460269 | 1497630.0 |
| Mei | 0.38741524 | 1526731.8 |
| Juni | 0.36185051 | 1643071.4 |
| Juli | 0.51452489 | 1324651.4 |
| Agustus | 0.44482395 | 1596523.0 |
| September | 0.73455754 | 1503258.4 |
| Oktober | 0.19433284 | 1522677.0 |
| November | 0.76112957 | 1670186.5 |
| Desember | 0.46912665 | 1545807.1 |

### **Implementasi Pada *Jupyter Notebook***

Pada pengimplementasi kan pada *jupyter notebook* pengujian prediksi pada data PDAM mendapatkan pola dan grafik yang baik, mendapatkan hasil pengujian data dengan pola 11-5-1 dengan proses pelatihan *epoch* 15000 dan pencapaian RMSE pada saat pengujian dengan RMSE = 0,0001016 dengan nilai akurasi 99,62013%. Pola grafik tersebut dapat dilihat pada gambar 4.2 berikut:

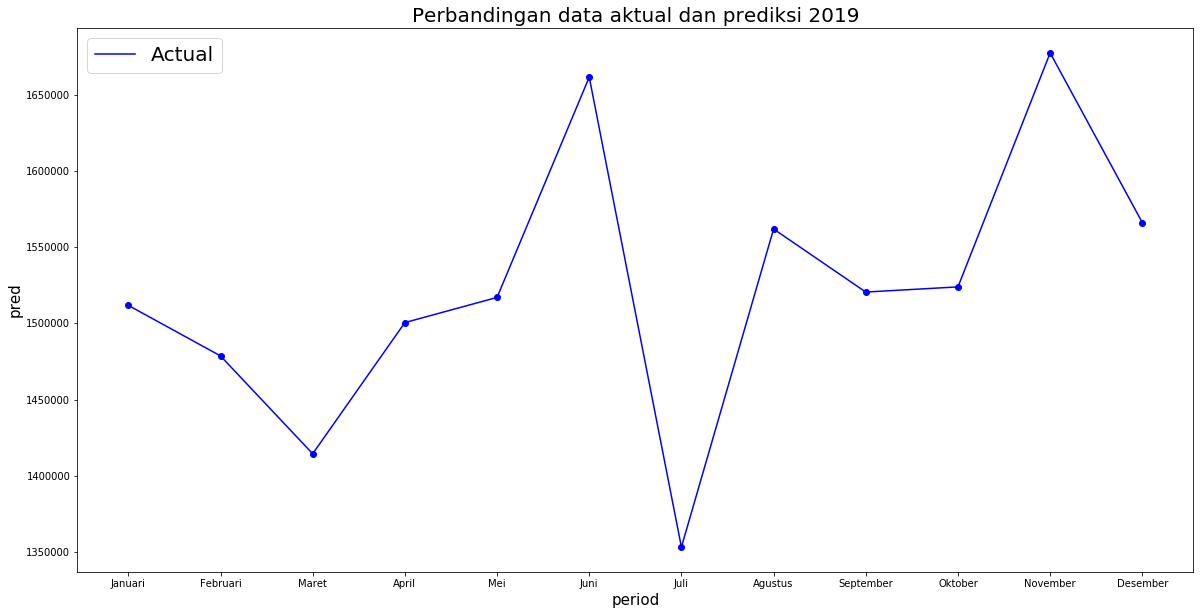


Gambar 4. 2 Grafik pengujian dengan pola 11-5-1

## **Anaslisis Pengujian**

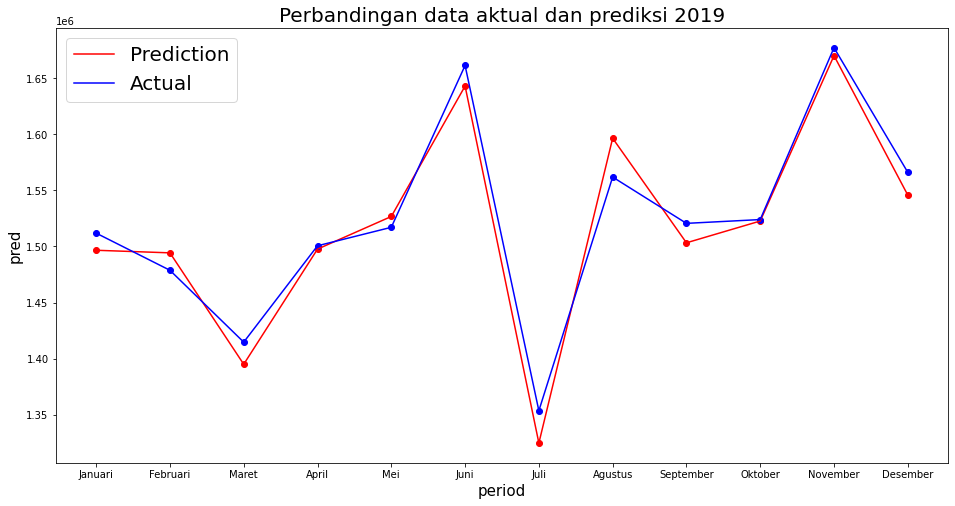
Setelah dilakukan tahap *testing* dan tahap *post-prepocessing* dengan melibatkan 37 *rows* pada data dan mendapatkan pencapaian RMSE pada saat pengujian dengan RMSE = 0,0001016 dengan proses pelatihan menggunakan *epoch* 15000 dengan nilai akurasi 99.62013626098633%.

Pada gambar 4.3 dan gambar 4.4 terdapat perbandingan data aktual target penjualan air PDAM dengan data denormalisasi atau hasil prediksi data target penjualan air PDAM pada tahun 2017-2019.



Gambar 4. 3 Data Aktual Target Penjualan Air PDAM

Gambar diatas menampilkan data aktual target penjualan air pada PDAM tahun 2019 dengan menampilkan nilai-nilai target penjualan air setiap bulan pada perusahaan PDAM Kota Padang.



Gambar 4. 4 Perbandingan Data Aktual dengan Hasil Prediksi

Pada gambar diatas, gambar dengan line warna biru menunjukkan data aktual dari target penjualan air PDAM dan pada gambar dengan line warna merah memperlihatkan hasil perbandingan data aktual dengan hasil prediksi menggunakan jaringan syaraf tiruan (JST). Dan dapat diketahui hasil perbandingan antara data aktual target penjualan air PDAM dengan data hasil prediksi target penjualan air PDAM dengan menggunakan metode JST (Jaringan Syaraf Tiruan) *Backpropagation* dengan menghasilkan hasil perbandingan yang tidak berbeda terlalu jauh dengan data aktual dengan nilai akurasi dan *error* yang baik, sehingga hasilnya dapat digunakan sebagai informasi yang berguna dan efektif dalam melihat informasi target penjualan air PDAM dengan tepat.

# **DAFTAR PUSTAKA**

Anjar, W. (2018). Jurnal Teknologi dan Sistem Informasi Optimasi Prediksi Dengan Algoritma Backpropagation Dan Conjugate Gradient Beale-Powell Restarts, *03*(2017), 370–380.

Anwar, B. (2011). Penerapan algoritma jaringan syaraf tiruan backpropagation dalam memprediksi tingkat suku bunga bank. *Jurnal SAINTIKOM*, *10*(2), 1–7.

Caesarendra, W., & Ariyanto, M. (2011). Panduan Belajar Mandiri MATLAB, (October), 37.

Chang, P., Wang, Y., & Liu, C. (2007). The development of a weighted evolving fuzzy neural network for PCB sales forecasting, *32*, 86–96. https://doi.org/10.1016/j.eswa.2005.11.021

Cömert, Z., & Fatih, A. (2017). A study of artificial neural network training algorithms for classification of cardiotocography signals, *7*(2), 93–103.

Costa, L. D. G. Da, & Santosa, S. (2014). PREDIKSI PENJUALAN AIR MINUM DALAM KEMASAN MENGGUNAKAN JARINGAN SYARAF TIRUAN RESILIENT, *10*, 186–210.

Defiyanti Sofi, M. K. (2013). Analisis dan Prediksi Kinerja Mahasiswa Menggunakan Teknik Data Mining. *Syntak*, *2*, 1–10.

Febrina, M., Arina, F., Ekawati, R., Teknik, J., Universitas, I., & Tirtayasa, A. (2013). Peramalan Jumlah Permintaan Produksi Menggunakan Metode Jaringan Syaraf Tiruan (JST) Backpropagation. *Jurnal Teknik Industri*, *1*(2), 174–179.

Hansun, S. (2013). Peramalan Data IHSG Menggunakan Metode Backpropagation. *Jurnal ULTIMATICS*, *5*(1), 26–30. https://doi.org/10.31937/ti.v5i1.310

Jauhari, D., Himawan, A., & Dewi, C. (2016). PREDIKSI DISTRIBUSI AIR PDAM MENGGUNAKAN METODE JARINGAN, *3*(2).

Kaur, D., & Kaur, J. (2017). Data Mining in Supermarket: A Survey. *International Journal of Computational Intelligence Research*, *13*(8), 1945–1951. Retrieved from http://www.ripublication.com

Khair, U., Fahmi, H., Hakim, S. Al, & Rahim, R. (2017). Forecasting Error Calculation with Mean Absolute Deviation and Mean Absolute Percentage Error. *Journal of Physics: Conference Series*, *930*(1). https://doi.org/10.1088/1742-6596/930/1/012002

Kusnadi, A., & Putra, I. (2015). Rancang Bangun Sistem Pakar Prediksi Stres Belajar Dengan Neural Network Algoritma Backpropagation. *Jurnal ULTIMATICS*, *7*(2), 105–112. https://doi.org/10.31937/ti.v7i2.361

Larose, D. T., & Larose, C. D. (2014). *Discovering Knowledge in Data*. *Discovering Knowledge in Data*. https://doi.org/10.1002/9781118874059

Lestari, N., & Van FC, L. L. (2017). Implementasi jaringan syaraf tiruan untuk menilai kelayakan tugas akhir mahasiswa (studi kasus di amik bukittinggi). *Digital Zone: Jurnal Teknologi Informasi Dan Komunikasi*, *8*(1), 10–24. https://doi.org/10.31849/digitalzone.v8i1.614

Matodang, Z. A. (2013). Jaringan Syaraf Tiruan Dengan Algoritma Backpropagtion Untuk Penentuan Kelulusan Sidang Skripsi. *Pelita Informatika Budi Darma*, *4*(1), 84–93.

Mustafa, M. S., Ramadhan, M. R., & Thenata, A. P. (2018). Implementasi Data Mining untuk Evaluasi Kinerja Akademik Mahasiswa Menggunakan Algoritma Naive Bayes Classifier. *Creative Information Technology Journal*, *4*(2), 151. https://doi.org/10.24076/citec.2017v4i2.106

Pakaja, F., & Naba, A. (2012). Jaringan Syaraf Tiruan dan Certainty Factor, *6*(1), 23–28.

Panchal, F. S., & Panchal, M. (2014). Review on Methods of Selecting Number of Hidden Nodes in Artificial Neural Network, *3*(11), 455–464.

Pramuntadi, A. (2017). Model Prediksi Rentet Waktu Neural Network Berbasis Particle Swarm Optimization Untuk. *Telematika*, *14*(02), 100–106. Retrieved from http://jurnal.upnyk.ac.id/index.php/telematika/article/view/2097

Prasetyo, E. (2015). REDUKSI DATA LATIH DENGAN K-SVNN SEBAGAI PEMROSESAN AWAL PADA ANN BACK-PROPAGATION UNTUK PENGURANGAN WAKTU, *6*(2), 223–230.

Riyanto, A. (2012). Politik Anggaran Provinsi Jawa Tengah: Analisis Realisasi APBD Provinsi Jawa tengah Tahun Anggaran 2008-2010. *Jurnal Ilmu Politik*.