TUGAS AKHIR – KS 141501

IMPLEMENTASI METODE ARTIFICIAL NEURAL NETWORK UNTUK PREDIKSI CURAH HUJAN DALAM PENANGGULANGAN DEMAM BERDARAH DI KABUPATEN MALANG

ANN METHOD IMPLEMENTATION TO PREDICT RAINFALL IN CASE OF DENGUE FEVER ANTICIPATION IN MALANG DISTRICT

Aditya Parama Hadi
NRP 5214 100 123 / 05211440000123

Dosen Pembimbing
Edwin Riksakomara, S.Kom., MT.

DEPARTEMEN SISTEM INFORMASI
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018

TUGAS AKHIR – KS 141501

IMPLEMENTASI METODE ARTIFICIAL NEURAL NETWORK UNTUK PREDIKSI CURAH HUJAN DALAM PENANGGULANGAN DEMAM BERDARAH DI KABUPATEN MALANG

ADITYA PARAMA HADI
NRP 5214 100 123 / 05211440000123

Dosen Pembimbing
Edwin Riksakomara, S.Kom., MT.

DEPARTEMEN SISTEM INFORMASI
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018

FINAL PROJECT – KS 141501

ANN METHOD IMPLEMENTATION TO PREDICT RAINFALL IN CASE OF DENGUE FEVER ANTICIPATION IN MALANG DISTRICT

ADITYA PARAMA HADI
NRP 5214 100 123 / 05211440000123

Supervisors
Edwin Riksakomara, S.Kom., MT.

INFORMATION SYSTEMS DEPARTMENT
Faculty of Information Technology and Communication
Institut Teknologi Sepuluh Nopember
Surabaya 2018

LEMBAR PENGESAHAN

IMPLEMENTASI METODE ARTIFICIAL NEURAL NETWORK UNTUK PREDIKSI CURAH HUJAN DALAM PENANGGULANGAN DEMAM BERDARAH DI KABUPATEN MALANG

TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Departemen Sistem Informasi
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh:

ADITYA PARAMA HADI
NRP. 05211440000123

Surabaya, 17 Juli 2018

KEPALA
DEPARTEMEN SISTEM INFORMASI

Dr. Ir. Aris Tjahyanto, M.Kom.
NIP.19650310 199102 1 001

LEMBAR PERSETUJUAN

IMPLEMENTASI METODE ARTIFICIAL NEURAL NETWORK UNTUK PREDIKSI CURAH HUJAN DALAM PENANGGULANGAN DEMAM BERDARAH DI KABUPATEN MALANG

TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Departemen Sistem Informasi
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh:

ADITYA PARAMA HADI

NRP. 05211440000123

Disetujui Tim Penguji: Tanggal Ujian: Juli 2018
Periode Wisuda: September 2018

Edwin Riksakomara, S.Kom, M.T **(Pembimbing I)**

Wiwik Anggraeni, S.Si, M.Kom **(Penguji I)**

Ahmad Mukhlason, S.Kom, M.Sc, Ph.D **(Penguji II)**

IMPLEMENTASI METODE ARTIFICIAL NEURAL NETWORK UNTUK PREDIKSI CURAH HUJAN DALAM PENANGGULANGAN DEMAM BERDARAH DI KABUPATEN MALANG

Nama Mahasiswa : Aditya Parama Hadi
NRP : 5214100123
Departemen : SISTEM INFORMASI FTIK-ITS
Dosen Pembimbing : Edwin Riksakomara, S.Kom., MT.

ABSTRAK

Nyamuk adalah salah satu spesies yang dapat membawa berbagai macam penyakit serius, misalnya malaria, filariasis, dan juga demam berdarah. Nyamuk berkembang biak dengan menempatkan telurnya di genangan air yang dapat terjadi karena hujan, banjir dan lain-lain. Karena itu dibutuhkan informasi curah hujan untuk mengestimasi siklus hidup nyamuk sesuai dengan ramalan hujan. Malang adalah salah satu contoh kota yang membutuhkan informasi tentang curah hujan. Kota Malang memiliki curah hujun cukup tinggi yaitu rata-rata 2088mm. Dengan curah hujan yang tinggi, maka akan berdampak pada berbagai sisi seperti banjir, kemacetan, dan juga kesehatan. Terutama pada pertumbuhan nyamuk penyebab demam berdarah. Dikarenakan ketika sering terjadi hujan dan banjir, maka akan banyak genangan air yang terbentuk. Genangan-genangan tersebut dapat menjadi tempat berkembang biaknya nyamuk. Untuk mengatasi hal tersebut dibutuhkan prediksi cuaca yang akurat agar dapat mengantisipasi sebelumnya. Penelitian ini dilakukan dengan tujuan untuk membuat prediksi curah hujan yang akurat. Di penelitian ini digunakan Artificial Neural Network (ANN) dengan model sistem backpropagation. Namun ada juga beberapa data intermittent yang berarti tidak konsisten sehingga harus dikelola terlebih dahulu dengan metode Bootstrap. Metode ini dapat memperkecil

nilai error pada akurasi sehingga diharapkan dapat menghasilkan data prediksi yang memiliki tingkat akurasi yang tinggi. Hasil yang didapatkan adalah model ANN yang memiliki performa yang baik dan dapat memprediksi dengan tingkat keberagaman yang tinggi. Setelah didapatkan data prediksi, diharapkan dapat dilakukan langkah antisipasi berdasarkan prediksi curah hujan tersebut. Model terbaik yang didapatkan dari penelitian ini menghasilkan MSE sebesar 163.2885 dan SMAPE sebesar 67.45%. ***Kata kunci: Peramalan, Artificial Neural Network (ANN), Intermittent Data, Bootstrap, Curah Hujan.***

ANN METHOD IMPLEMENTATION TO PREDICT RAINFALL IN CASE OF DENGUE FEVER ANTICIPATION IN MALANG DISTRICT

Student Name : **Aditya Parama Hadi**
NRP : **5214100123**
Departement : **SISTEM INFORMASI FTIK-ITS**
Supervisor : **Edwin Riksakomara, S.Kom., MT.**

ABSTRACT

Mosquitos are a species carrying most serious diseases, such as malaria, filariasis and also dengue fever. It reproduces by placing its eggs in water puddles which formed from rains, floods and others. Rainfall prediction is needed to estimate mosquito life cycle according to the rain forecast. Malang is an example of a city that needs informations regarding rainfall. Malang has a pretty significant rainfall rate which averages to 2088mm per year. With its relatively high rainfall, it will have impacts in various kinds of things such as floodings, water hazard, and also health. Especially in dengue fever mosquitos. With multiple rains and floods, there will be a lot of puddles formed. These puddles could potentially be an environment where mosquitos breed. To avoid such things, there needs to be an accurate rainfall forecast so those things can be avoided. This research's purpose is to create an accurate model to predict rainfall. This research will use Artificial Neural Network (ANN) with backpropagation system. However, there are some inconsistent intermittent data so it needs to be preprocessed beforehand by using Bootstrap method. This method could potentially reduce error values in the data so hopefully it could create prediction data with high level of accuracy. Expected result obtained from this research is an ANN model with exceptional performance and could predict data with high level of accuracy. After acquiring the prediction data, hopefully anticipation gesture could be made according to the rainfall prediction. The best model

resulted from this research produced MSE of 163.2885 and SMAPE of 67.45%. ***Keywords: Forecasting, Artificial Neural Network (ANN), Intermittent Data, Bootstrap, Rainfall.***

KATA PENGANTAR

Puji syukur kepada Allah SWT atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Penelitian Tugas Akhir dengan judul "**IMPLEMENTASI METODE ARTIFICIAL NEURAL NETWORK UNTUK PREDIKSI CURAH HUJAN DALAM PENANGGULANGAN DEMAM BERDARAH DI KABUPATEN MALANG"** yang merupakan syarat kelulusan pada Departemen Sistem Informasi Fakultas Teknologi Informasi dan Komunikasi Institut Teknologi Sepuluh Nopember Surabaya. Selama waktu melaksanakan Penelitian Tugas Akhir, banyak pihak pihak yang membantu penulis dan juga memberikan saran. Penulis menyampaikan banyak terima kasih kepada:

1. Allah SWT yang senantiasa memberikan segala rahmat dan hidayah sehingga penulis diberikan kemudahan, kelancaran dan kesehatan selama pengerjaan Penelitian Tugas Akhir di Sistem Informasi ITS.
2. Kedua orang tua (Karno Prihatin dan Prima Kristalina) serta Kakak Karina Setya Kartika yang selalu mendoakan kelancaran dan keberhasilan penulis serta banyak membantu dan mendukung dalam pengerjaan Penelitian Tugas Akhir
3. Bapak Edwin Riksakomara, S.Kom, M.T selaku dosen pembimbing yang memberikan arahan dan bimbingan selama pengerjaan Penelitian Tugas Akhir hingga penyusunan laporan.
4. Ibu Wiwik Anggraeni, S.Si, M.Kom dan Bapak Ahmad Mukhlason S.Kom, M.Sc, selaku dosen penguji yang selalu memberikan saran dan masukan selama Tugas Akhir ini.
5. Ibu Nur Aini Rakhmawati selaku dosen wali dari penulis, yang selalu memberikan dorongan, motivasi dan wejangan

selama penulis menempuh kuliah S1 di Sistem Informasi ITS

6. Teman-teman Warung Squad yang sering penulis datangi tempatnya untuk mengerjakan tugas, serta diajak makan-makan. Dan juga teman-teman E-Home yang juga sering penulis kunjungi dengan alasan yang sama.
7. Teman-teman OSIRIS dan angkatan lain yang memberi dukungan dan semangat selama pengerjaan Penelitian Tugas Akhir.
8. Seluruh dosen, staff, karyawan dan stakeholder lain di Departemen Sistem Informasi FTIK ITS serta departemen dan fakultas lain yang memberikan ilmu dan bantuan kepada penulis dalam waktunya di ITS.
9. Semua pihak yang tidak mampu penulis sebutkan yang telah memberikan dukungan dan bantuan pada penulis selama pengerjaan Tugas Akhir dan selama menempuh pendidikan di ITS.

Terima kasih atas segala bantuan, dukungan, serta doa yang diberikan. Semoga Allah SWT senantiasa memberikan kesehatan, keselamatan, karunia dan nikmat-Nya.
Penulis pun ingin memohon maaf karena penulis menyadari bahwa Tugas Akhir ini masih belum sempurna. Selain itu penulis bersedia untuk menerima kritik dan saran terkait dengan Tugas Akhir ini. Semoga Tugas Akhir ini dapat bermanfaat bagi seluruh pembaca.

Surabaya, Juni 2018

Penulis

DAFTAR ISI

ABSTRAK i
ABSTRACT iii
KATA PENGANTAR v
DAFTAR GAMBAR xi
DAFTAR TABEL xiii
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Permasalahan 3
1.3 Batasan Permasalahan 3
1.4 Tujuan 4
1.5 Manfaat 4
1.6 Relevansi Tugas Akhir 4
BAB II TINJAUAN PUSTAKA 7
2.1 Penelitian Sebelumnya 7
2.2 Landasan Teori 9
2.2.1 Badan Meteorologi Klimatologi dan Geofisika Stasiun Klimatologi Karangploso Malang 9
2.2.2 Peramalan 9
2.2.3 Data Intermiten 10
2.2.4 Bootstrap 10
2.2.5 Neural Network 12
2.2.6 Perangkat Jaringan 13
2.2.7 Perangkat Simpul 14

2.2.8 Model Artificial Neural Network dengan Sistem Backpropagation 14
2.2.9 Weight, Output dan Error 17
2.2.10 Tingkat Akurasi Model 18
BAB III METODOLOGI 19
3.1 Tahapan Pengerjaan Tugas Akhir 19
3.1.1 Studi Literatur 19
3.1.2 Pengumpulan dan Preproses Data 20
3.1.2.1 Interpolasi Data 20
3.1.2.2 *Bootstrapping* Data 20
3.1.3 Pembuatan Model ANN 21
3.1.4 Uji Coba dan Validasi Model 23
3.1.5 Analisis Hasil Peramalan 24
3.1.6 Pembuatan Laporan Tugas Akhir 24
BAB IV PERANCANGAN 25
4.1 Pengumpulan Data 25
4.1.1 Preproses Data 29
4.1.2 Bootstrap Data 31
4.2 Pembuatan Model Artificial Neural Network 32
4.2.1 Pembagian Data 32
4.2.2 Model Neural Network 33
4.2.3 Penentuan Input Layer 33
4.2.4 Penentuan Hidden Layer 35
4.2.5 Penentuan Parameter 35
BAB V IMPLEMENTASI 37

5.1 Pengolahan Data 37
5.1.1 Bootstrapping Data 43
5.1.2 Pembuatan Dataset Untuk Model 44
5.2 Pembuatan Model Artificial Neural Network 45
5.2.1 Model Artificial Neural Network 45
5.2.2 Data Input 51
5.2.3 Data Output 53
5.2.4 Parameter yang Digunakan 53
5.3 Penerapan Model 54
5.3.1 Penerapan Model Neural Network 54
5.3.2 Proses *Training* dan *Testing* 60
5.3.3 Uji Performa 61
BAB VI HASIL DAN PEMBAHASAN 65
6.1 Lingkungan Uji Coba 65
6.2 Percobaan Parameter 66
6.2.1 Pengujian Epoch 66
6.3 Percobaan Model 67
6.3.1 Model dengan Jumlah Node Input 1 Periode Sebelumnya 67
6.3.2 Model dengan Jumlah Node Input 2 Periode Sebelumnya 71
6.3.3 Model dengan Jumlah Node Input 3 Periode Sebelumnya 72
6.3.4 Model dengan Jumlah Node Input 4 Periode Sebelumnya 73

6.3.5 Model dengan Jumlah Node Input 5 Periode Sebelumnya 75
6.4 Kesimpulan Hasil Percobaan 78
6.5 Hasil Peramalan 79
6.6 Analisis Perbandingan Hasil Peramalan Data dengan Bootstrap dan Data Asli 80
BAB VII KESIMPULAN DAN SARAN 81
7.1 Kesimpulan 81
7.2 Saran 81
DAFTAR PUSTAKA 83
BIODATA PENULIS 87
LAMPIRAN A 89
LAMPIRAN B 95
LAMPIRAN C 97
LAMPIRAN D 99

DAFTAR GAMBAR

1 Gambar 1.1 Roadmap laboraturium RDIB 5
2 Gambar 2.1 Langkah-langkah metode bootstrap 11
3 Gambar 2.2 Model struktur jaringan saraf tiruan 13
4 Gambar 2.3 Fungsi aktivasi logsig 14
5 Gambar 3.1 Metodologi Pengerjaan Tugas Akhir 19
6 Gambar 3.2 Metodologi ANN 21
7 Gambar 3.3 Model struktur jaringan..... 23
8 Gambar 4.1 Suhu Minimum 26
9 Gambar 4.2 Kelembaban Rata-Rata 27
10 Gambar 4.3 Curah Hujan..... 27
11 Gambar 4.4 Kecepatan Angin Rata-Rata..... 28
12 Gambar 4.5 Kejanggalan Data dan Keterangan 29
13 Gambar 4.6 Contoh Pengerjaan Interpolasi 30
14 Gambar 4.7 Contoh Pengerjaan Interpolasi 2 31
15 Gambar 4.8 Referensi Program Bootstrap 32
16 Gambar 4.9 Input Layer menggunakan 1 periode..... 33
17 Gambar 4.10 Input Layer menggunakan 2 periode..... 34
18 Gambar 5.1 Training Suhu Minimum..... 39
19 Gambar 5.2 Training Kelembaban Rata-Rata..... 39
20 Gambar 5.3 Training Kecepatan Angin Rata-Rata 40
21 Gambar 5.4 Training Curah Hujan 40
22 Gambar 5.5 Testing Suhu Minimum 41
23 Gambar 5.6 Testing Kelembaban Rata-Rata..... 41
24 Gambar 5.7 Testing Curah Hujan 42
25 Gambar 5.8 Testing Kecepatan Angin Rata-Rata 42
26 Script 5.1 Program Boostrap 43
27 Gambar 5.9 Hasil data setelah di Boostrap 44
28 Gambar 5.10 Pembagian Dataset..... 45
29 Gambar 5.11 Model Artificial Neural Network 1 Periode 46
30 Gambar 5.12 Model Artificial Neural Network 2 Periode 47

31 Gambar 5.13 Model Artificial Neural Network 3 Periode 48
32 Gambar 5.14 Model Artificial Neural Network 4 Periode 49
33 Gambar 5.15 Model Artificial Neural Network 5 Periode 50
34 Script 5.2 Script Perubahan Parameter 55
35 Script 5.3 Multilayer Percetron Neural Network 56
36 Script 5.4 Script untuk Otomasi Iterasi (Nested Looping) 60
37 Script 5.5 Script untuk Training dan Testing 60
38 Script 5.6 Script untuk Uji Performa 63
39 Script 5.7 Script untuk Penyimpanan Hasil 63
40 Script 5.8 Script untuk Penyimpanan Hasil, Performa beserta Header 64
41 Gambar 6.1 MSE Node dengan 1 Periode Sebelum 70
42 Gambar 6.2 MSE pada Periode 2 72
43 Gambar 6.3 MSE pada Periode 3 74
44 Gambar 6.4 MSE pada Periode 4 75
45 Gambar 6.5 MSE pada Periode 5 77
46 Gambar 6.6 MSE terkecil tiap Jumlah Periode 78
47 Gambar 6.7 Model Optimal dengan Target 79
48 Gambar 6.8 Model Optimal dengan Data Asli 80

DAFTAR TABEL

1 Tabel 2.1 Referensi penelitian sebelumnya 7
2 Tabel 4.1 Penentuan Hidden Layer dan Parameter 36
3 Tabel 5.1 Data Input 1 Periode 51
4 Tabel 5.2 Data Input 2 Periode 52
5 Tabel 5.3 Penjelasan Script Perubahan Parameter 55
6 Tabel 5.4 Penjelasan Script Multilayer Perceptron Neural Network 56
7 Tabel 5.5 Penjelasan Script Training dan Testing 61
8 Tabel 5.6 Penjelasan Script Uji Performa 62
9 Tabel 6.1 Perangkat Keras dalam Lingkungan Uji Coba 65
10 Tabel 6.2 Perangkat Lunak dalam Lingkungan Uji Coba 66
11 Tabel 6.3 MSE Node 4 Periode 1 68
12 Tabel 6.4 Penjelasan Kode Model 69
13 Tabel 6.5 Hasil Model dengan 1 Periode 69
14 Tabel 6.6 Hasil Model Periode 2 71
15 Tabel 6.7 Hasil Model dengan 3 Periode Sebelum 73
16 Tabel 6.8 Hasil Model Periode 4 74
17 Tabel 6.9 Hasil Model Periode 5 76
18 Tabel 6.10 MSE terbaik pada tiap Jumlah Periode 78
19 Tabel 6.11 Perbandingan pada Model dengan Data yang berbeda 80

Halaman ini sengaja dikosongkan

BAB I
PENDAHULUAN

Bab ini berisikan gambaran secara umum dari penelitian tugas akhir. Gambaran meliputi latar belakang, rumusan permasalahan, batasan penelitian, tujuan serta manfaat yang dapat diambil. Relevansi penelitian juga akan dijelaskan sesuai dengan kebutuhan masyarakat dan bidang keilmuan

1.1 Latar Belakang

Curah hujan menurut definisinya adalah ketinggian curah hujan yang terjadi pada suatu kurun waktu di mana air tersebut terkonsentrasi, dengan satuan mm/jam [1]. Curah hujan ini mencakup tetes hujan, salju, batu es, embun dan juga embun kristal. Informasi dari banyaknya curah hujan ini adalah salah satu unsur yang penting terhadap aktifitas sehari-hari seperti keselamatan masyarakat, produksi pertanian, perkebunan, perikanan, dan salah satu contoh utamanya yaitu pengamatan siklus hidup nyamuk.

Peramalan curah hujan, dengan berbagai bentuk analisis dan informasi yang dihasilkan, akan terasa dampaknya untuk membantu dalam pengamatan siklus hidup nyamuk terutama nyamuk demam berdarah. Ini dapat berguna untuk melakukan tindakan preventif dalam menekan pertumbuhan jumlah nyamuk demam berdarah.

Indonesia termasuk dalam wilayah beriklim tropika basah, dengan ciri-ciri pola hujan yang berbeda dengan wilayah yang beriklim tropika atau beriklim sedang. Namun demikian karena Indonesia meliputi kawasan yang sangat luas, maka pola hujan yang jatuh di wilayah Indonesia sangat beragam, dipengaruhi oleh kondisi topografis dan geografis wilayah masing-masing [2].

Salah satu daerah yang memiliki curah hujan signifikan adalah Malang. Di lokasi ini hanya ada musim kemarau singkat, dengan suhu rata-rata 23.7 °C dan curah hujan rata-rata 2088 mm. Musim kemarau di Malang adalah hanya dari bulan Juni hingga September. Sisanya, dari Oktober hingga Mei, Malang mengalami musim penghujan dengan presipitasi terbesar di bulan Januari. Ini menunjukkan curah hujan di kota Malang cukup tinggi. Hal ini dapat berdampak pada beberapa hal, seperti banjir. Banjir sendiri dapat menyebabkan hal-hal lain yang lebih besar dampaknya seperti longsor, kerusakan, dan lain-lain. Selain itu, genangan air yang disebabkan oleh hujan dan banjir di berbagai tempat juga dapat menjadi tempat nyamuk berkembang biak.

Pada penulisan tugas akhir ini, yang diambil adalah data curah hujan dari BPS (Badan Pusat Statistik) Kota Malang. Perkembangan statistik dapat digunakan juga sebagai metode ilmiah dalam melakukan peramalan curah hujan, dan Badan Pusat Statistik Kota Malang adalah unit pelaksana teknis yang memiliki data curah hujan tiap bulannya di Kota Malang. Data curah hujan meliputi data temperatur (TM), kecepatan angin (WS), kelembapan (HM), dan presipitasi (PCP).

Ramalan yang dilakukan biasanya didasarkan oleh data masa lalu yang kemudian dianalisis dengan cara-cara tertentu. Data masa lalu tersebut dikumpulkan, dipelajari, lalu dianalisis hubungannya dengan waktu terkait. Dikarenakan adanya faktor waktu yang terkait, maka hasil analisis data masa lalu dapat juga digunakan untuk meramalkan data dari masa yang akan datang, namun juga ada ketidakpastian yang harus diperhitungkan sehingga hasil ramalan tidak mungkin tepat 100%.

Walaupun demikian, hasil ramalan tetap dapat digunakan untuk membantu dalam melakukan perencanaan, pengawasan, pembuatan dan juga pengambilan keputusan.

Menurut penelitian yang dilakukan oleh Ling Chen dan Xu Lai, model ANN yang mereka buat dapat bekerja lebih baik dibandingkan model ARIMA dalam melakukan peramalan jangka pendek [3]. Ini dikarenakan model ANN cocok dalam prediksi kecepatan angin karena karakteristik dari kecepatan angin ada banyak unit proses yang berhubungan dan identik satu sama lain. Teknik-teknik tersebut lebih tidak memakan waktu dibandingkan metode konvensional yang lain [4]. Selain itu, data *time series* yang menggunakan variasi yang berbeda juga digunakan model yang berbeda pula.

Berdasarkan latar belakang di atas, maka disusun Tugas Akhir dengan judul "Pengembangan Penelitian Curah Hujan Untuk Penelitian Demam Berdarah Dengan Metode ANN Pada Kabupaten Malang"

1.2 Rumusan Permasalahan

Rumusan masalah dari penelitian tugas akhir ini adalah sebagai berikut:

1. Bagaimana model ANN yang paling cocok dalam melakukan peramalan data?
2. Bagaimana tingkatan akurasi yang dihasilkan dengan model ANN yang digunakan?

1.3 Batasan Permasalahan

Batasan dalam pengerjaan penelitian tugas akhir ini adalah:

1. Data yang digunakan adalah data dari curah hujan yang terdokumentasikan oleh Stasiun Meteorologi Kabupaten Malang
2. Peramalan dilakukan dengan menggunakan data harian curah hujan periode Januari 2012 – Januari 2017
3. Hasil peramalan akan mencangkup 2 tahun mendatang terhitung dari tahun 2017 yakni tahun 2018 dan 2019.

1.4 Tujuan

Tujuan yang akan dicapai dari pengerjaan penelitian tugas akhir ini adalah sebagai berikut:

1. Mengetahui model ANN terbaik dalam melakukan peramalan
2. Mengetahui tingkatan akurasi dari model ANN

1.5 Manfaat

Manfaat yang dapat diambil dengan adanya penelitian tugas akhir ini adalah sebagai berikut:

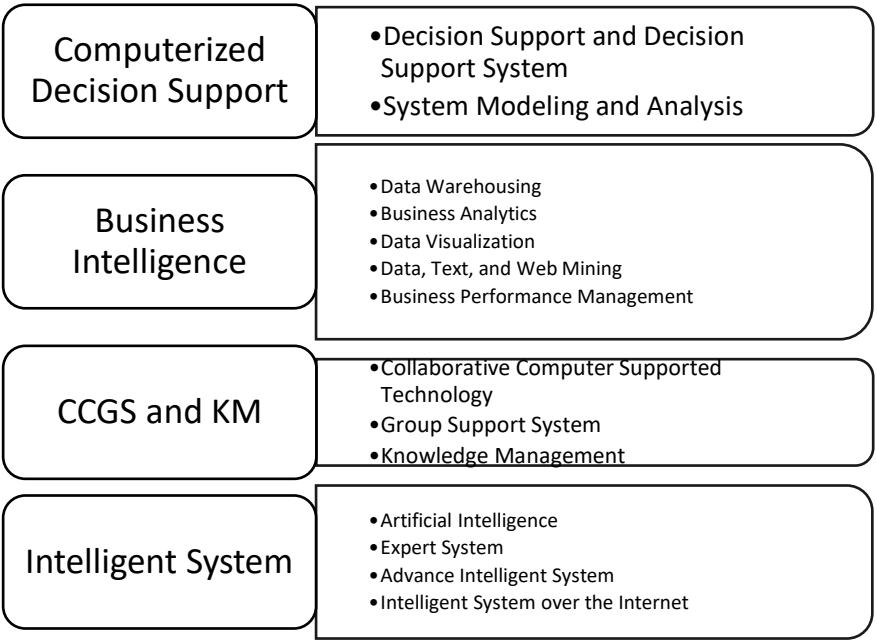
1. Instansi dapat melakukan prediksi curah hujan untuk tahun-tahun berikutnya di Kabupaten Malang sehingga dpat memudahkan proses pengambilan keputusan.
2. Instansi memiliki data untuk diberikann kepada pemerintah Malang agar dapat melakukan pencegahan dini terhadap intensitas curah hujan yang cenderung tinggi.
3. Institusi/departemen membuat dan menjaga hubungan baik dengan perusahaan yang memiliki ketertaitan.

1.6 Relevansi Tugas Akhir

Penelitian tugas akhir ini berkaitan dengan kebutuhan masyarakat akan prediksi curah hujan, terutama untuk menanggulangi dan mengantisipasi banjir, serta mengantisipasi berkembangnya jumlah nyamuk demam berdarah. Karena dengan curah hujan yang tinggi, maka diharapkan pihak-pihak yang terkait dapat menanggulangi efek curah hujan tersebut dan mempersiapkan secara dini.

Penelitian ini juga berhubungan dengan mata kuliah Teknik Peramalan, karena menggunakan metode peramalan yang diajarkan dalam mata kuliah tersebut. Selain itu juga menggunakan metode pengembangan *Artificial Neural Network* (ANN) yang diajarkan pada mata kuliah Sistem Cerdas.

Berdasarkan roadmap laboratorium Rekayasa Data dan Inteligensi Bisnis (RDIB) seperti pada gambar 1.1, tugas ini masuk pada kategori Computerized Decision Support. Ini karena data yang diperoleh akan diolah untuk menghasilkan analisis dan model yang dapat digunakan pada proses pembuatan dan pengambilan keputusan.



Gambar 1.1 Roadmap laboraturium RDIB

Halaman ini sengaja dikosongkan

BAB II
TINJAUAN PUSTAKA

Bab ini berisikan beberapa tinjauan pustaka yang dapat membantu pengerjaan penelitian tugas akhir

2.1 Penelitian Sebelumnya

Ada 3 penelitian sebelumnya yang digunakan sebagai referensi dalam pengerjaan penelitian ini, sebagaimana ditunjukkan oleh Tabel 2.1.

Tabel 2.1 Referensi penelitian sebelumnya

Penelitian 1	
Judul Penelitian	*Neural Network Load Forecasting with Weather Ensemble Predictions* [5]
Penulis/Tahun Penelitian	Taylor, J.W.; Buizza, R./2002
Gambaran Umum	Di penelitian ini ada investigasi penggunaan kumpulan prediksi dalam pengaplikasian *neural network* untuk mengambil hasil peramalan dari 1 hingga 10 hari ke depan. Kumpulan dari prediksi berisi beberapa skenario untuk variabel cuaca. Ini kemudian akan dimuat dan dirata-ratakan untuk dilihat bahwa hasil peramalan lebih akurat dibandingkan dengan menggunakan peramalan cuaca secara tradisional.
Keterkaitan Tugas Akhir	Penelitian ini bisa membantu mencari metode yang akan digunakan dan juga pebandingannya dengan metode peramalan cuaca secara tradisional.
Penelitian 2	
Judul Penelitian	*Dynamic ANN for precipitation estimation and forecasting from radar observations* [6]

Penulis/Tahun Penelitian	Yen-Ming Chiang, Fi-John Chang, Ben Jong-Dao Jou, Pin-Fang Lin/2007
Gambaran Umum	Pemanfaatan data radar meteorology untuk estimasi presipitasi kuantitatif sesaat (QPE) dan peramalan presipitasi kuantitatif (QPF) untuk operasional hidrologi di daerah aliran sungai. Teknik estimasi curah hujan berbasis radar yang paling umum digunakan adalah fungsi power-law. Di penelitian ini, dikenalkan pendekatan JN dinamis untuk membuat QPE dan QPF dengan struktur data radar tiga dimensi.
Keterkaitan Tugas Akhir	Sebagai acuan dalam pengerjaan tugas akhir, dan langkah-langkah estimasi dengan model ANN dinamis.
Penelitian 3	
Judul Penelitian	*Weather forecasting model using artificial neural network* [7]
Penulis/Tahun Penelitian	K. Abhishek, MP Singh/2012
Gambaran Umum	Penelitian ini memeriksa penerapan pendekatan ANN dengan cara mengembangkan model prediktif *nonlinear* yang efektif dan handal untuk analisa cuaca dan juga membandingkan dan mengevaluasi model yang dihasilkan dengan menggunakan fungsi-fungsi yang berbeda, *hidden layer* dan *neuron* untuk meramalkan temperatur maksimum selama 365 hari dalam satu tahun.
Keterkaitan Tugas Akhir	Penelitian ini dapat membantu sebagai referensi dalam pengembangan model prediktif *nonlinear* yang akan digunakan dalam peramalan cuaca dengan menggunakan metode ANN.

2.2 Landasan Teori

Berikut merupakan landasan teori yang digunakan dalam pengerjaan penelitian dan berkaitan dengan tugas akhir yang dikerjakan.

2.2.1 Badan Meteorologi Klimatologi dan Geofisika Stasiun Klimatologi Karangploso Malang

Ini adalah salah satu stasiun yang merupakan anggota dari Badan Meteorologi, Klimatologi dan Geofisika (BMKG) Indonesia yang bertugas untuk meliput daerah Malang. Di stasiun ini, diambil data perhari tentang Suhu Minimum (°C), Suhu Maximal (°C), Suhu Rata-Rata (°C), Kelembapan Rata-Rata (%), Curah Hujan (mm), Lama Penyinaran (jam), Kecepatan Angin Rata-Rata (knot), Arah Angin terbanyak (deg), Kecepatan Angin Terbesar (knot), dan Arah Angin saat Kecepatan Maksimum. Namun, tidak semua data didapatkan oleh stasiun. Ini dikarenakan data tidak dapat terukur atau tidak ada data yang direkap pada periode tersebut.

2.2.2 Peramalan

Peramalan secara definisi, menurut Gor dalam bukunya, adalah suatu usaha prediksi ke masa depan dengan menggunakan data dari masa lalu dan dengan menggunakan metode kualitatif atau kuantitatif [8]. Lalu ditambahkan dalam bukunya Moon dan Mentzer bahwa peramalan memegang kunci penting dalam suatu perusahaan ketika melakukan perencanaan produksi dan kegiatan operasional. Dan diharapkan ketika dilakukan peramalan ini dapat meminimalisir biaya dengan fasilitas yang cukup memadai. Fungsi bisnis juga dapat lebih efektif jika peramalan data penjualan yang didapatkan akurat [9].

2.2.3 Data Intermiten

Dalam melakukan peramalan, terkadang terdapat masalah yaitu permintaan yang mendekati nilai 0 (nol) dalam beberapa kali periode. Situasi ini seringkali terjadi ketika data bernilai kecil, dan seringkali dengan ukuran yang bervariasi [10]. Permasalahan ini disebut *intermittent data* oleh Waller (2016). Situasi ini juga sering terjadi pada industri seperti otomotif, manufaktur, dan lain lain.

Salah satu metode yang dapat digunakan untuk mengatasi hal ini adalah metode *Bootstrap* [10]. Hasil dari metode ini akan menjadi hasil prediksi yang akurat. Selain *bootstrap* juga ada metode lain yang dapat digunakan sebagai pembanding yaitu metode *Temporal aggregation*.

2.2.4 Bootstrap

Metode *Bootstrap* adalah teknik statistik yang menggunakan *random sampling* sebagai pengganti. Willemain [11] mengusulkan metode dengan melakukan *bootstrapping* pada observasi-observasi dari permintaan yang tidak nol untuk memprediksi permintaan dalam waktu *lead time* tertentu (jarak waktu antara pemesanan dan kedatangan).

Ada 2 hal penting yang patut diperhatikan dalam metode ini:

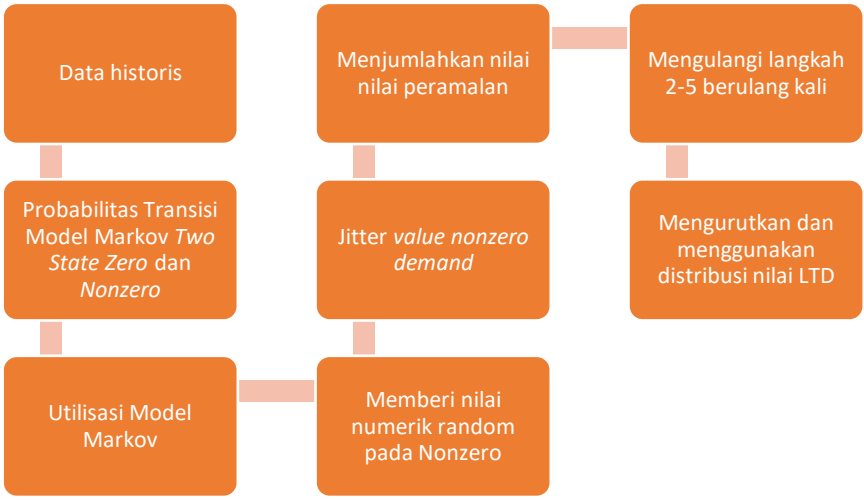
Yang pertama adalah untuk menghindari ramalan dengan nilai yang sama seperti sebelumnya, ada proses *jittering* yang digunakan agar mendapatkan hasil yang lebih bervariasi. Misalkan Y adalah nilai-nilai data sebelumnya yang telah dipilih, dan Z adalah *normal random variable*. Maka rumus dari proses *jittering* adalah sebagai berikut:

$$Y_{jittered} = 1 + INT\{Y + Z\sqrt{Y}\} \qquad (2.1)$$

Batasan lainnya adalah jika $Y_{jittered} \leq 0$ maka $Y_{jittered} = Y$.

Hal ke 2 yang patut diperhatikan adalah untuk memodelkan *autocorrelation* yang mungkin ada, digunakan model *two-stage*

Markov Chain sesuai dengan observasi data yang ada nol dan tidak (*zero* dan *non-zero*). Tujuannya adalah untuk meramalkan sebuah rangkaian data *zero* dan *non-zero* terlebih dahulu [12]. Secara lengkapnya adalah seperti pada Gambar 2.1



Gambar 2.1 Langkah-langkah metode bootstrap

Langkah pengerjaan *bootstrap* lebih jelasnya adalah sebagai berikut:

- Dengan data historis, dilakukan estimasi kemungkinan transisi untuk *two-state Markov Chain*
- Buat urutan peristiwa dari *Markov Chain* pada *lead-time* yang diinginkan
- Ganti setiap peristiwa *non-zero* dengan nilai historis, lalu lakukan *jitter*
- Jumlahkan nilai peramalan untuk menghasilkan sebuah *lead-time demand* (LTD)

Hasil dari penelitian oleh Willemain [11] menunjukkan bahwa metode *bootstrap* memperbaiki metode Croston dan SES secara signifikan dalam melakukan peramalan LTD.

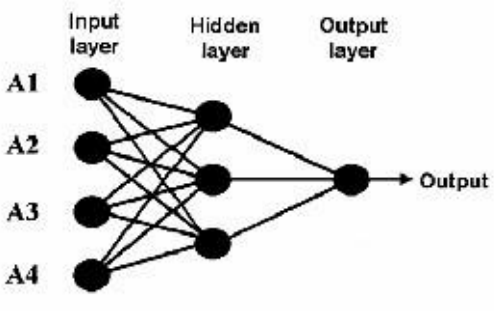
a. Temporal aggregation

Alternatif dari metode *bootstrap* adalah dengan agregasi temporal. Metode ini menggabungkan beberapa periode waktu menjadi blok-blok. Keuntungan dari metode ini adalah metode ini dapat menghilangkan nilai-nilai nol (*zero*) pada rangkaian waktu, namun kekurangannya adalah jumlah dari pengamatan historisnya jadi sangat berkurang. Ada 3 langkah utama dalam metode ini. Yang pertama adalah menentukan tipe agregasi. Ini termasuk memilih jumlah pengamatan individu yang akan digabungkan dalam satu blok, serta menentukan blok tersebut akan bertumpukan atau tidak. Bagian kedua adalah meramalkan nilai selanjutnya pada rangkaian agregasi, dapat dilakukan dengan metode peramalan yang biasanya. Terakhir, hasil peramalan dilakukan disagregasi, atau diturunkan, menjadi waktu periode seperti semula. Caranya adalah dengan sejumlah pemberat (*weighting*) berdasarkan rasio pengamatan sebelumnya dari tiap-tiap blok, atau juga bisa dengan berat yang sama.

2.2.5 Neural Network

Secara biologis jaringan saraf terdiri dari neuron-neuron yang saling berhubungan. Neuron merupakan unit struktural dan fungsional dari sistem saraf, mempunyai kemampuan untuk mengadakan respon bila dirangsang dengan intensitas rangsangan cukup kuat. Respon neuron bila dirangsang adalah memulai dan menghantarkan impuls. Jaringan saraf tiruan merupakan gabungan sejumlah elemen yang memproses informasi dari *input* sehingga memberikan suatu informasi keluaran. Sekelompok obyek dipelajari oleh sistem belajar dengan tujuan untuk mengenali bentuk pola setiap bentuk tersebut. Proses ini dilakukan dengan cara melatih sistem belajar (*train neural network*) melalui

pemberian bobot dan *bias* pada hubungan antar simpul. Hasil yang dicapai adalah didapatkannya sekelompok bobot dan *bias* (pada kesalahan minimum yang dicapai) untuk semua pola yang dipelajari, hal ini sesuai dengan anggapan menemukan energi terendah dalam proses mengenali sekelompok obyek pola yang dipelajari. Jaringan saraf tiruan mempunyai distribusi pararel arsitektur dengan sejumlah besar simpul mempunyai bobot dan *bias* tertentu, sebagaimana ditunjukkan pada gambar 2.2. Pada gambar 2.2 dapat dilihat bahwa ada 4 input layer, dengan 3 hidden layer dan 1 output layer. Pada penelitian kali ini tidak persis seperti yang ada di gambar, namun ada beberapa kemiripan dengan bentuk model struktur jaringan saraf tiruan pada gambar sehingga dapat menjadi sebuah referensi dalam melakukan pengerjaan [13].



Gambar 2.2 Model struktur jaringan saraf tiruan

2.2.6 Perangkat Jaringan

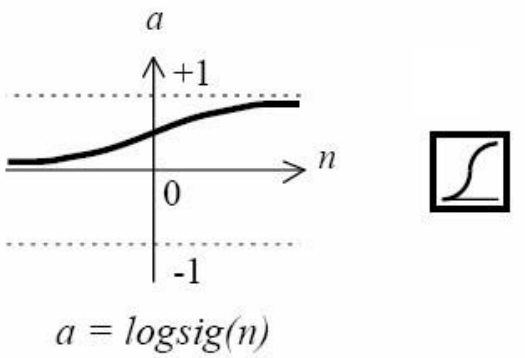
Jaringan saraf tiruan terdiri dari sejumlah lapisan dan simpul yang berbeda untuk tiap-tiap *layer*. Jenis *layer* dapat dibedakan menjadi:

1. *Input Layer*: terdiri dari unit-unit simpul yang berperan sebagai *input* proses pengolahan data pada *neural network.*

2. *Hidden Layer*: terdiri dari unit-unit simpul yang dianalogikan sebagai lapisan tersembunyi dan berperan sebagai lapisan yang meneruskan respon dari *input*.
3. *Output Layer*: terdiri dari unit-unit simpul yang berperan memberikan solusi dari data *input*.

2.2.7 Perangkat Simpul

Tingkat aktivasi dari simpul (*node*) dapat berharga diskrit yaitu 0 dan 1, atau kontinu yaitu antara 0 dan 1. Hal tersebut bergantung dari penerapan fungsi aktivasi itu sendiri. Jika menggunakan fungsi '*hard limitting*' , maka tingkat aktivasinya bernilai 0 (atau -1) dan 1. Apabila menggunakan fungsi *sigmoid* maka tingkat aktivasinya terbatas pada daerah antara 0 dan 1. Pada tugas akhir kali ini lebih banyak menggunakan fungsi *sigmoid* terutama *logsig*. Contoh fungsi *sigmoid* dengan fungsi aktivasi *logsig* yaitu seperti pada Gambar 2.3.



Gambar 2.3 Fungsi aktivasi logsig

2.2.8 Model Artificial Neural Network dengan Sistem Backpropagation

Backpropagation adalah salah satu program komputasi untuk penerapan *neural network* yang banyak digunakan untuk memecahkan masalah *non-linear* serta *network multilayer* dengan menggeneralisasi persamaan *widrow-hoff* [14]. Metode ini menggunakan metode penurunan gradien. *Backpropagation* menggunakan pelatihan terbimbing (*train neural network*) dan dalam pengaturan jumlah lapisan (*layer*) mudah dilakukan sehingga banyak diterapkan pada berbagai permasalahan.

Backpropagation merupakan sistem *train neural network* yang dapat menghitung tingkat kesalahan dari hasil keluarannya, sehingga *neural network* yang digunakan memiliki kesalahan terkecil. *Neural network* harus dilatih berulang-ulang dengan pola input yang sesuai, sehingga *neural network* dapat mengenali pola dan diperoleh bobot dan *bias* tiap simpul dengan kesalahan terkecil.

Kelemahan *Backpropagation* diantaranya adalah:

- *Backpropagation* dapat mengenali pola input yang telah diajarkan tetapi tidak dapat mengenali pola input yang baru.
- Dalam mengenali pola input yang baru, maka pola tersebut harus diajarkan sehingga pola yang lama akan dilupakan.

Parameter *backpropagation*:

- Inisiasi bobot: Memasukan nilai bobot dan nilai *bias* untuk tiap simpul dengan bilangan acak (*random*).
- Menghitung tingkat aktivasi:

1. Tingkat aktivasi dari simpul input tidak perlu dihitung
2. Menghitung tingkat aktivasi dari simpul *hidden* dan *output* dengan rumus:

- Untuk simpul *hidden*

$$a_j = (\sum W_{ij} . p_i + b_j) \tag{2.1}$$

- Untuk simpul *output*

$$a_k - \left(\sum W_{jk} \cdot a_j + b_k\right) \quad (2.2)$$

dengan
pi : nilai input
Wij : bobot ke simpul hidden
Wjk : bobot ke simpul output
bj : nilai bias simpul hidden
bk : nilai bias simpul output

3. Melatih bobot

- Penyesuaian bobot: mencari nilai bobot sesuai dengan keluaran yang diinginkan dengan persamaan:
- Perubahan bobot ke simpul *hidden*

$$W_{ij}(t+1) = W_{ij}(t) + \Delta W_{ij} \quad (2.3)$$

- Perubahan bobot ke simpul *output*

$$W_{jk}(t+1) = W_{jk}(t) + \Delta W_{jk} \quad (2.4)$$

- Perhitungan perubahan bobot dengan persamaan:
- Perubahan bobot ke simpul *hidden*

$$\Delta W_{ij} = \eta\delta_j \cdot p_i \quad (2.5)$$

- Perubahan bobot ke simpul *output*

$$\Delta W_{jk} = \eta\delta k \cdot a_j \quad (2.6)$$

4. Perhitungan *gradient error*

- Perubahan bobot ke simpul *hidden*

$$\delta_j = a_j(1 - a_j)\Sigma\delta_k \cdot W_{ij} \quad (2.7)$$

- Perubahan bobot ke simpul *output*

$$\delta_k = a_k(1 - a_k)(T_k - a_k) \quad (2.8)$$

dengan:

η : koefesien pembelajaran (antara 0 dan 1) δ_j : gradien error pada unit j

δ_k : gradien error pada unit k

T_k : harga aktivasi yang diinginkan dari simpul output ke k (target)

a_k *: harga aktivasi yang diperoleh pada simpul keluaran ke k*

5. Mengulang langkah algoritma diatas sehingga dapat menentukan nilai *error* terkecil (yang diinginkan).

2.2.9 Weight, Output dan Error

Hubungan antar *node* diasosiasikan dengan suatu nilai yang disebut dengan bobot atau *weight*. Setiap *node* pasti memiliki *output*, *error* dan *weight*nya masing - masing.

Output merupakan keluaran dari suatu *node*. *Error* merupakan tingkat kesalahan yang terdapat dalam suatu *node* dari proses yang dilakukan. *Weight* merupakan bobot dari *node* tersebut ke *node* yang lain pada *layer* yang berbeda. Nilai *weight* berkisar antara -1 dan 1.

Bobot – bobot atau *weight* yang tersimpan di dalam jaringan syaraf tiruan ini disebut sebagai bobot interkoneksi. Nilai bobot yang baik akan memberikan keluaran yang sesuai, dalam arti mendekati keluaran yang diharapkan (*target output*) untuk suatu *input* yang diberikan.

Bobot awal dalam suatu jaringan syaraf tiruan biasanya diperoleh secara *random* dan sebaiknya diinisialisasi dengan nilai yang relatif kecil, yaitu berkisar antara -0,1 sampai 0,1 [15]. Baru dalam tahap pelatihan, bobot tersebut akan mengalami penyesuaian melalui suatu proses perhitungan matematik agar tercapai nilai bobot yang sesuai.

2.2.10 Tingkat Akurasi Model

Dalam melakukan penghitungan, ada beberapa cara yang dapat dilakukan untuk menghitung tingkat akurasi dari model. Cara-cara ini digunakan sesuai dengan jenis dan bentuk data yang ada.

1. **Mean Squared Error (MSE)**
   Formula yang digunakan untuk mengelola *error* yang ada adalah dengan menggunakan persamaan (2.9)
   $$MSE = \sqrt{\frac{\sum_{t=1}^{n}(A_t - F_t)^2}{n}}$$
   (2.9)
   Dengan At adalah nilai data aktual periode t dan Ft adalah nilai *forecast* pada periode t, dan n adalah nilai jumlah data yang di *forecast*.

2. **Mean Absolute Percetage Error (MAPE) / Symmetrical MAPE (SMAPE)**
   Merupakan rata-rata *absolute* dari presentase *error* yang didapat dari peramalan yang dilakukan sebelumnya.
   $$MAPE = \frac{100\%}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| \quad (2.10)$$
   $$\text{SMAPE} = \frac{100\%}{n}\sum_{t=1}^{n}\frac{|F_t - A_t|}{|A_t| + |F_t|} \quad (2.11)$$
   Dengan At adalah nilai data aktual periode t dan Ft adalah nilai *forecast* pada periode t, dan n adalah nilai jumlah data yang di *forecast*.

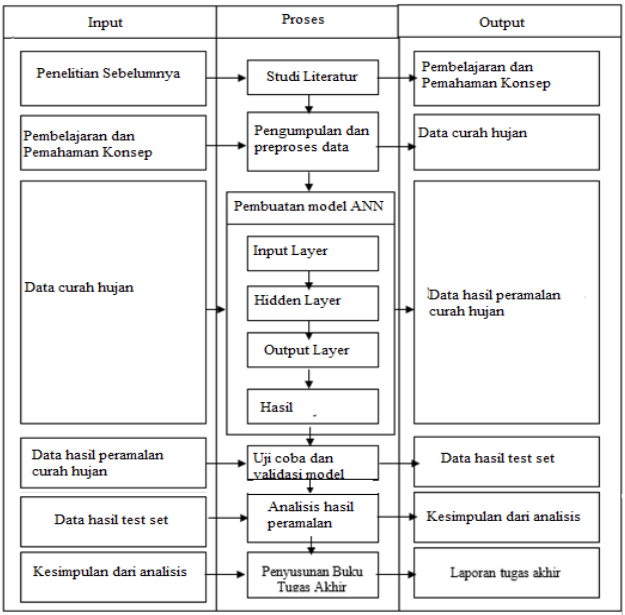
   SMAPE adalah alternatif dari MAPE ketika ada nilai 0 atau mendekati 0 pada data

BAB III
METODOLOGI

Bab ini berisikan metodologi dalam melakukan pengerjaan penelitian tugas akhir.

3.1 Tahapan Pengerjaan Tugas Akhir

Pada bab ini dijelaskan bagaimana pengerjaan secara sistematis yang akan dilakukan ketika menyelesaikan penelitian. Langkah pengerjaan ditunjukkan seperti pada gambar 3.1.



Gambar 3.1 Metodologi Pengerjaan Tugas Akhir

3.1.1 Studi Literatur

Di tahapan ini ada hal-hal yang harus dipersiapkan, seperti identifikasi masalah yang ada. Lalu dianalisa menggunakan

referensi dari beberapa studi literatur. Studi literatur ini didapat dari beberapa referensi seperti penelitian jurnal, seminar, tugas akhir lain, dan lain-lain. Setelah mempelajari studi literatur maka akan dilakukan pengumpulan data untuk analisa mendalam. Analisanya seperti apakah literatur sesuai dengan pengerjaan, dan apakah data sesuai dan dapat mendukung pengerjaan penelitian.

3.1.2 Pengumpulan dan Preproses Data

Di tahapan ini akan dilakukan pengumpulan dan analisis dari tipe data yang didapat. Pengumpulan data dilakukan dengan pengambilan dari website BMKG Malang. Data ini merupakan data curah hujan yang meliputi temperature (TM), kecepatan angin (WS), kelembapan (HM) dan presipitasi (PCP).

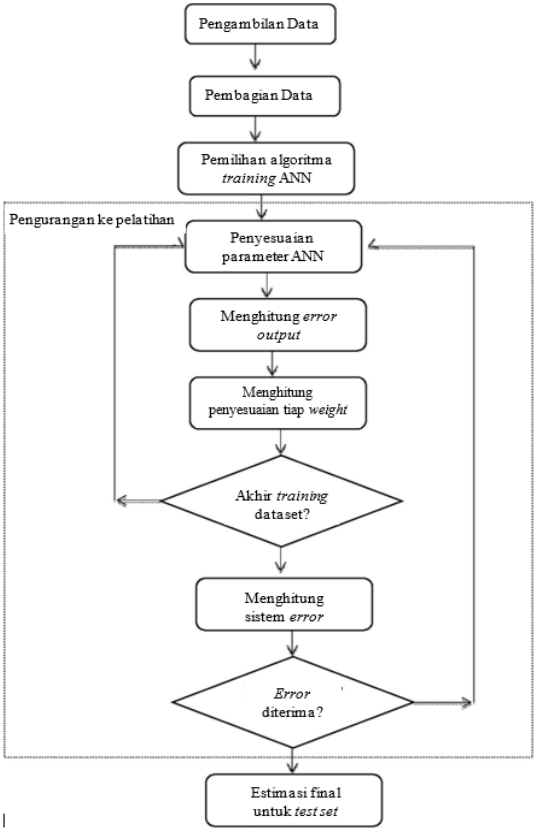
3.1.2.1 Interpolasi Data

Beberapa data ditemukan tidak terukur ataupun tidak ada data, sehingga perlu dilakukan interpolasi data dengan cara menghitung data tersebut dengan interpolasi dari data sebelum dan sesudahnya.

3.1.2.2 *Bootstrapping* Data

Data-data yang masih ada sifat *intermittent* akan di transformasi dengan metode *Bootstrapping*. Cara pengerjaan metode ini ada di Bab Landasan Teori. Data akan dibagi jadi data *training* dan data *testing*. Data *training* fungsinya adalah membuat rancangan model peramalan. Data *testing* fungsinya adalah untuk evaluasi rancangan model yang sebelumnya dibuat. Dalam menentukan model ANN yang terbaik untuk prediksi, dapat dilakukan kombinasi percobaan antara jumlah *input layer*, *hidden layer*, *output layer*, jumlah *hidden layer*, fungsi aktivasi antar *neuron* dan jenis *backpropagation training*. Kemudian selain dengan metode *Bootstrapping*, juga dapat dilakukan metode *Temporal aggregation* sebagai bentuk metode pembanding pada metode *Bootstrapping*.

3.1.3 Pembuatan Model ANN

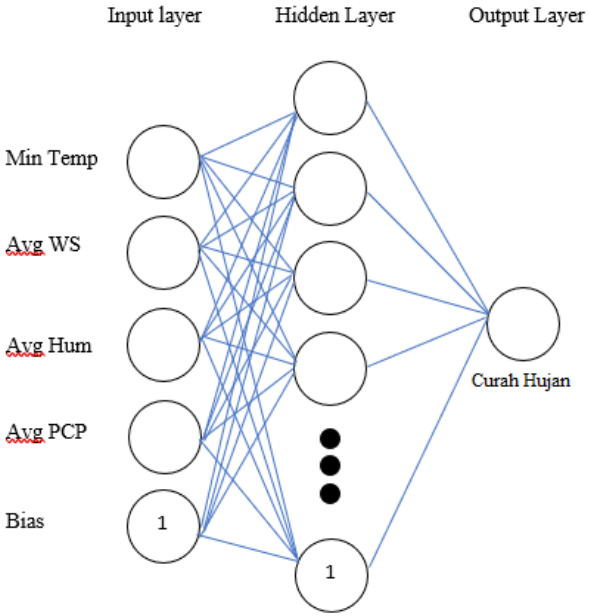


Gambar 3.2 Metodologi ANN

Di tahapan ini, data sudah diolah sehingga pembuatan model ANN dapat dilakukan. Model ANN (jaringan syaraf tiruan) yang digunakan pada penlitian adalah model Artificial Neural Network dengan Sistem Backpropagation. Tahapan setelah diperoleh model ANN adalah melakukan *forecast* dengan data

testing. Secara rinci metodologi peramalan menggunakan metode ANN akan dijelaskan pada Gambar 3.2. [16]

1. **Pengolahan Data**
   Pada tahap ini, dilakukan pengolahan pada data. Data ini akan menjadi faktor dalam pemilihan metode ANN. Data yang berbentuk data *training* akan menjadi input dalam metode ANN tersebut.
2. **Pelatihan ANN**
   Setelah itu, disesuaikan parameter pembelajaran dari ANN. Ini akan menghasilkan nilai *error* dalam bentuk *output*. Lalu dihitung penyesuaiannya dari masing-masing *weight*.
3. **Perhitungan Ulang Dataset *Training***
   Hasil dari langkah sebelumnya diperiksa kembali, apakah sudah merupakan akhir dari dataset atau belum. Jika belum, maka diulangi langkah dari awal. Jika sudah, maka dilanjutkan ke langkah selanjutnya.
4. **Perhitungan *Mean System Error***
   Dalam bagian ini dilakukan penghitungan rata-rata *system error*. Caranya adalah dengan perbandingan *output* sistem dengan nilai *output* yang diinginkan, dan menggunakan *error* tersebut untuk mengarahkan *training*nya. Algoritma dari pembelajaran *error* tersebut bertujuan untuk meminimalisir sinyal *error* dari tiap iterasi *training*.
5. **Cek Penerimaan *Error***
   Dari hasil penghitungan *system error* sebelumnya, akan didapatkan *error*nya. Kemudian setelah selesai akan dibandingkan. Apakah *error* tersebut dapat diterima ataukah perlu dilakukan penyesuaian kembali pada *parameter* ANN [17]. Jika masih perlu penyesuaian maka kembali ke langkah pelatihan ANN. Jika sudah, maka akan dilanjutkan ke langkah selanjutnya.
6. **Estimasi Final Dataset *Testing***
   Hasil dari langkah selanjutnya akan menjadi estimasi akhir dari dataset *testing*. Lalu akan dibandingkan dengan dataset *training*. Model paling akhir ini akan dilanjutkan menjadi acuan dalam melakukan peramalan data periode.



Gambar 3.3 Model struktur jaringan

Bentuk model dari Neural Network yang terbentuk adalah seperti pada Gambar 3.3. Ada sekelompok bobot yang diambil dari data curah hujan yaitu temperatur, rata-rata kecepatan angin, rata-rata kelembapan, rata-rata presipitasi, dan juga bias. Lalu ini diproses dalam hidden layer dan menjadi output prediksi curah hujan.

3.1.4 Uji Coba dan Validasi Model

Uji model akan dilakukan pada hasil yang didapat dari proses sebelumnya. Ada beberapa scenario yang akan diujicobakan, dan akan ditentukan model mana yang tepat untuk digunakan sebagai peramalan.

3.1.5 Analisis Hasil Peramalan

Setelah didapatkan hasil dari langkah-langkah forecasting sebelumnya, akan dilakukan testing tingkat akurasi dengan menghitung nilai MSE dan MAPE. Hasil dari tingkatan error akurasi ini akan dibandingkan dengan metode-metode tradisional ataupun individual.

3.1.6 Pembuatan Laporan Tugas Akhir

Terakhir adalah dokumentasi pengerjaan tugas akhir dengan penyusunan Laporan Tugas Akhir. Laporan Tugas Akhir disusun dengan format seperti pada bagian sub-bab Penyusunan Buku Laporan Tugas Akhir.

BAB IV
PERANCANGAN

Dalam bab ini, dijelaskan tentang perancangan model *Aritifical Neural Network* yang digunakan dalam peramalan curah hujan Kabupaten Malang, Jawa Timur.

4.1 Pengumpulan Data

Tahapan ini dilakukan untuk mengumpulkan data yang dibutuhkan dalam penelitian tugas akhir ini. Pengumpulan data yang dilakukan adalah dengan mengunduh data dari situs jaringan Badan Pusat Statistika tentang data curah hujan yang dikumpulkan dalam bentuk harian dari beberapa stasiun klimatologi di Kabupaten Malang. Periode yang digunakan adalah dari Januari 2012 – Januari 2017.

Ada 4 kumpulan data yang akan digunakan sebagai input, yaitu Suhu Minimum, Kelembaban Rata-rata, Curah Hujan (mm) dan Kecepatan Angin Rata-rata. Maka *plotting* dari data-data tersebut dapat dilihat di Gambar 4.1, 4.2, 4.3 dan 4.4. Data secara penuh dapat dilihat pada Lampiran A.

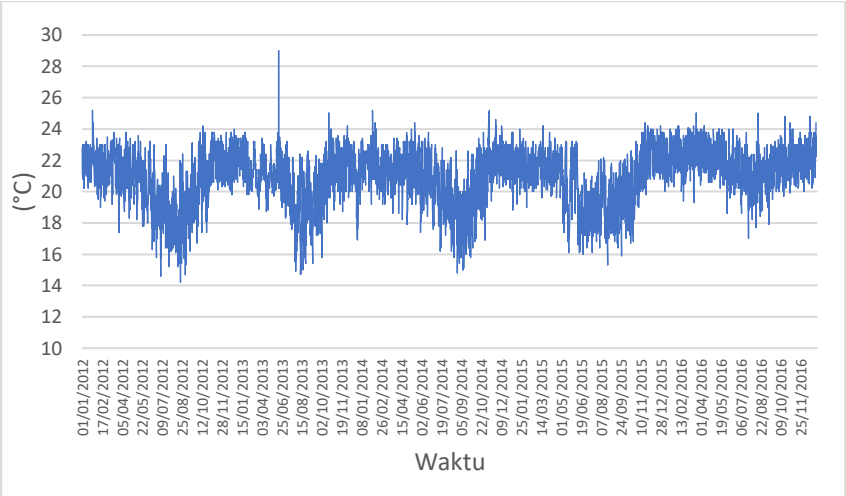
Pada Gambar 4.1 dapat kita lihat bahwa tren dari data adalah naik turun, ini dikarenakan suhu minimum berubah sesuai dengan musim. Ketika musim sedang musim kemarau, maka suhu minimum akan cenderung lebih tinggi dibandingkan dengan musim hujan. Sumbu x dari grafik ini adalah waktu, dan sumbu y adalah suhu dalam satuan derajat Celsius.

Pada Gambar 4.2 data juga dapat dilihat trennya naik turun dengan alasan yang serupa seperti pada gambar 4.1. Kelembaban rata-rata akan terpengaruh oleh musim, ketika musim penghujan maka kelembaban akan lebih tinggi dibandingkan dengan musim

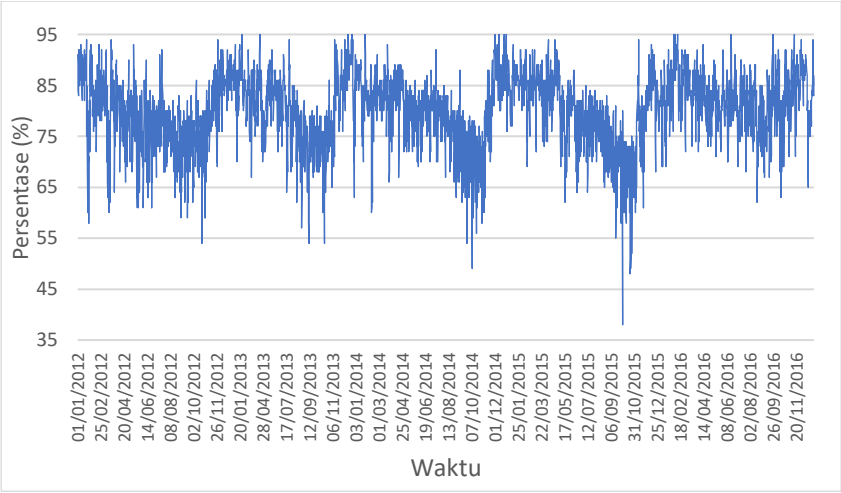
kemarau. Sumbu x pada grafik ini adalah waktu dan sumbu y adalah persentase kelembaban rata-rata pada udara dalam satuan persen.

Gambar 4.3 menunjukkan data curah hujan. Data curah hujan sudah jelas menunjukkan nilai curah hujan, sehingga ketika musim hujan nilai-nilai tersebut akan meninggi. Namun ketika musim kemarau terkadang tidak ada hujan sama sekali sehingga curah hujan mencapai nilai 0. Sumbu x pada grafik adalah waktu dan sumbu y adalah nilai curah hujan dalam satuan milimeter (mm).

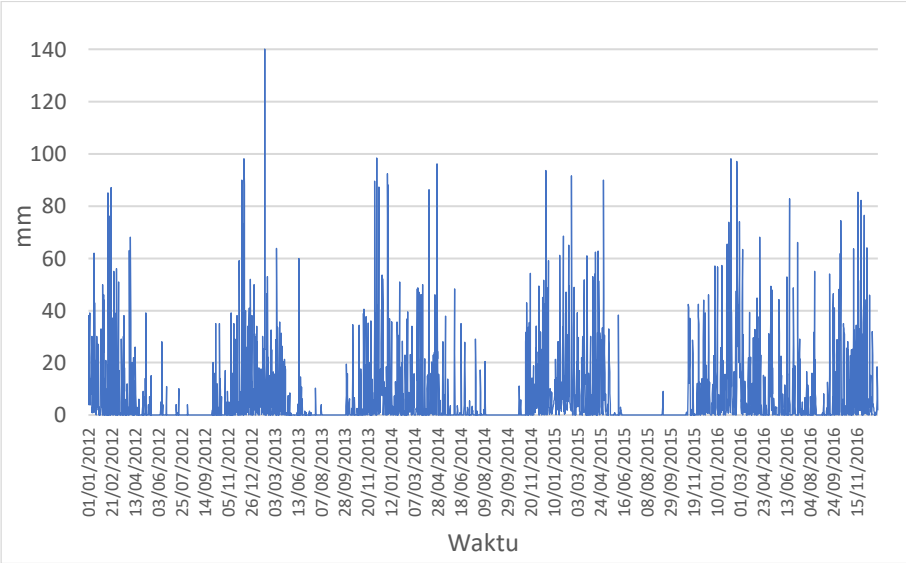
Gambar 4.4 menunjukkan kecepatan angin rata-rata, dan tren data tidak terlihat terlalu terpengaruh oleh musim hujan maupun kemarau. Sumbu x pada grafik adalah waktu sedangkan sumbu y adalah kecepatan angin rata-rata dengan satuan knot.



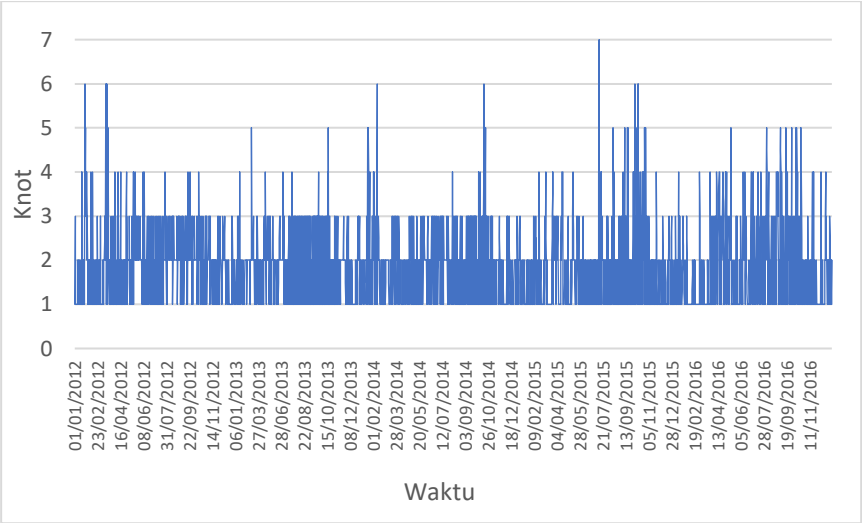
Gambar 4.1 Suhu Minimum



Gambar 4.2 Kelembaban Rata-Rata



Gambar 4.3 Curah Hujan



Gambar 4.4 Kecepatan Angin Rata-Rata

Adapun bentuk asli dari data-data tersebut sebenarnya ada beberapa kejanggalan intermiten, maksudnya adalah ada beberapa data yang tidak terukur dan tidak ada data. Dijelaskan pada Gambar 4.5 beserta keterangan dari Badan Pusat Statistik. Dalam *dataset* yang didapat dari BPS ada sejumlah data yang tidak terukur dan tidak ada data. Ini berbeda dengan data bernilai nol (0) karena nilai 0 pada data masih memiliki arti bahwa nilai dari data adalah 0 dan dapat digunakan pada penghitungan dan pencarian model. Sedangkan ketika nilai data tidak terukur ataupun tidak ada data berarti penghitungan tidak dapat dilakukan dan proses pencarian dan pembuatan model tidak dapat dilakukan dikarenakan penghitungan yang tidak dilakukan. Oleh karena itu kejanggalan data-data tersebut harus diproses terlebih dahulu, dengan kata lain dilakukan *preprocessing* sebelum diolah.

Keterangan
* 8888 : Data Tidak Terukur
* 9999 : Tidak Ada Data

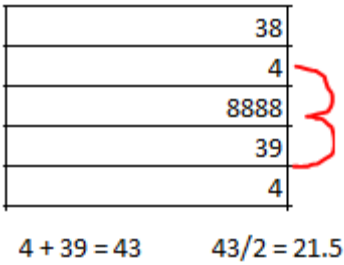
	Kelembaban Rata-rata	Curah Hujan (mm)	Lama Penyinaran (jam	Kecepatan Angin Rata-	Arah
5,1	91	22	1	1	E
23	89	4	1,3	1	E
24	84	38	3,4	3	NE
5,8	83	4	5	1	NW
2,7	90	8888	3	1	S
4,7	92	39	5,5	1	SE
5,2	91	4	0,5	1	SE
23	92	5	0,6	9999	SE
23	85	4	3	1	SE
4,9	89	14	3,5	1	W
2,5	90	13	1	9999	S
5,9	86	8	2	1	W
25	88	16	1	1	W
3,4	86	22	0,6	2	S
2,6	91	30	0	1	SE
4,2	93	7	0,5	1	W
3,4	87	1	1,4	2	SE
5,1	87	3	1,5	1	W
5,7	87	1	0,5	1	W
3,8	86	29	0,2	2	SE
3,8	85	1	0,6	2	SE
5,9	82	4	2	1	W
3,1	91	8888	1	9999	B
25	90	9	1,5	1	W

Gambar 4.5 Kejanggalan Data dan Keterangan

4.1.1 Preproses Data

Untuk mengatasi kejanggalan pada data, dilakukan interpolasi pada data-data tersebut. Langkah-langkah interpolasi adalah dengan menggunakan data sebelum dan sesudah dari data intermiten, dan dilakukan interpolasi titik tengah dari kedua data tersebut. Jika ada

data intermiten yang berurutan 2 atau 3 kali, maka tetap dilakukan interpolasi titik tengah dengan nilai yang sesuai dengan data sebelum dan sesudahnya.

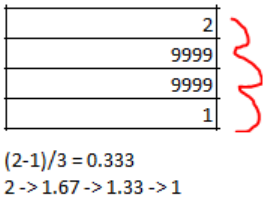


Gambar 4.6 Contoh Pengerjaan Interpolasi

Contoh pengerjaan interpolasi adalah seperti pada gambar 4.6 ketika ada data tidak terukur diantara 2 data (4 dan 39). Ambil nilai kedua data tersebut dan cari titik tengahnya. Karena data yang tidak terukur hanya satu, maka nilainya adalah titik tengah dari 4 dan 39 yaitu 21.5.

Kemudian pada gambar 4.7 dicontohkan ketika ada 2 data yang tidak terukur secara berurutan diantara 2 data yang terukur (2 dan 1). Maka cara menghitungnya adalah cari selisih data dari nilai sebelum dan sesudah (2 dan 1) kemudian bagi dengan jarak data (3 langkah: 2 -> x -> x -> 1). Nilai selisih ini adalah jarak nilai dari data sebelum (2) hingga data sesudah (1). Sehingga nilai dari data yang tidak terukur adalah 1.67 dan 1.33.

Jika ada 3 data ataupun lebih yang tidak terukur ataupun tidak ada data, maka caranya sama seperti yang telah dijelaskan sebelumnya. Yang perlu diperhatikan adalah selisih dari data sebelum dan sesudah, serta perhatikan jarak nilai kedua data tersebut.



Gambar 4.7 Contoh Pengerjaan Interpolasi 2

4.1.2 Bootstrap Data

Dalam penelitian ini ada beberapa data yang bernilai nol (0) yang dapat menyebabkan tingkatan *error* yang tinggi dalam penghitungan akurasi model. Data bernilai 0 banyak ditemukan dalam kolom curah hujan, dikarenakan memang tidak ada nilai curah hujan yang disebabkan beberapa faktor seperti sedang musim kemarau. Sehingga diperlukan adanya program *bootstrapping*. Bentuk program dari Bootstrapping ini berupa koding matlab, di mana data yang sudah diinterpolasi sebelumnya dimasukkan ke dalam program.

Program *bootstrap* yang digunakan dalam penelitian ini adalah merupakan modifikasi dari program *bootstrap* blok stasioner untuk vektor *time series* seperti yang terlihat pada Gambar 4.8 [18]. Pengubahan dilakukan pada bagian input data dan bentuk data output, dimana data input disesuaikan dengan *input layer* pada penelitian, yaitu 4 kolom data curah hujan, serta data output program *bootstrap* yang akan menjadi data *input layer* baru dengan data yang telah dilakukan *bootstrap*.

```
% ------------------------------------------------------------
% BLOCK SELECTION
% ------------------------------------------------------------
switch sim
case 1 % Stationary BB, geometric pdf
   b = geornd(1/L(1),1,n);
case 2 % Stationary BB, uniform pdf
   b = round(L(1)+(L(2)-1)*rand(1,n));
case 3 % Circular bootstrap (fixed block size)
   b = L(1) * ones(1,n);
end

% ------------------------------------------------------------
% BOOTSTRAP REPLICATION
% ------------------------------------------------------------
Zb = [];
for j=1:kz
   Zb = [Zb loopBB(Z(:,j),n,b,I)];
end

% ============================================================
% loopBB ==> UNIVARIATE BOOTSTRAP LOOP
% ============================================================
function xb = loopBB(x,n,b,I);

h=1;
for m=1:n
   for j=1:b(m)
      xb(h) = x(I(m)+j-1);
      h = h + 1;
      if (h == n+1); break; end;
   end
   if (h == n+1); break; end;
end

xb=xb';
```

Gambar 4.8 Referensi Program Bootstrap

4.2 Pembuatan Model Artificial Neural Network

Pada tahapan ini, dilakukan pembuatan model *Artificial Neural Network* dengan menggunakan program MATLAB. Lalu hasil dari proses pembuatan tersebut akan menciptakan model di mana diharapkan model tersebut dapat menghasilkan nilai peramalan curah hujan yang paling optimal.

4.2.1 Pembagian Data

Hasil data yang didapatkan dari situs Badan Statistika adalah data curah hujan dari bulan Januari tahun 2012 hingga bulan Januari tahun 2017 dalam bentuk harian seperti yang terlampir pada lampiran A.

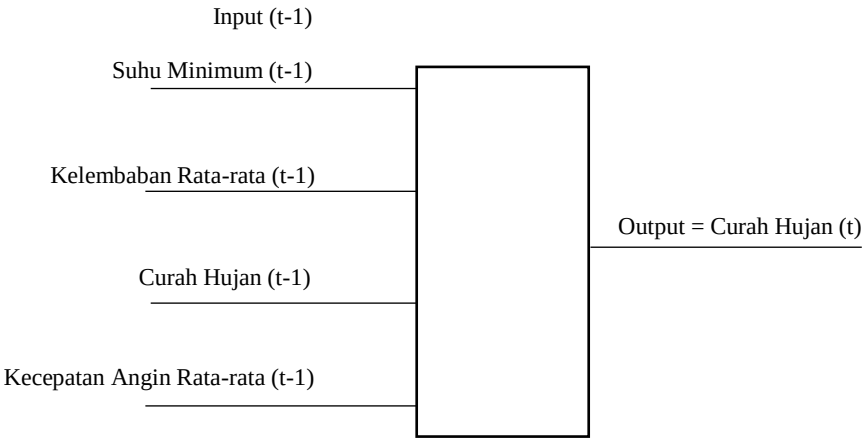
Kemudian data tersebut dibagi menjadi data latih (*training*) dan data uji (*testing*) dengan rasio 75%: 25%. Jumlah data adalah 3494 sehingga data training yang digunakan adalah 2600 dan data uji adalah 894.

4.2.2 Model Neural Network

Uji coba terhadap *neural network* perlu dilakukan untuk mendapatkan hasil pembuatan model yang terbaik. Hasil tersebut adalah nilai MAPE yang terkecil (minimum). Dalam melakukan percobaannya, dilakukan pengubahan pada *input layer* dan *hidden layer* dan dilakukan sebanyak tiga kali pada setiap pengubahan parameternya.

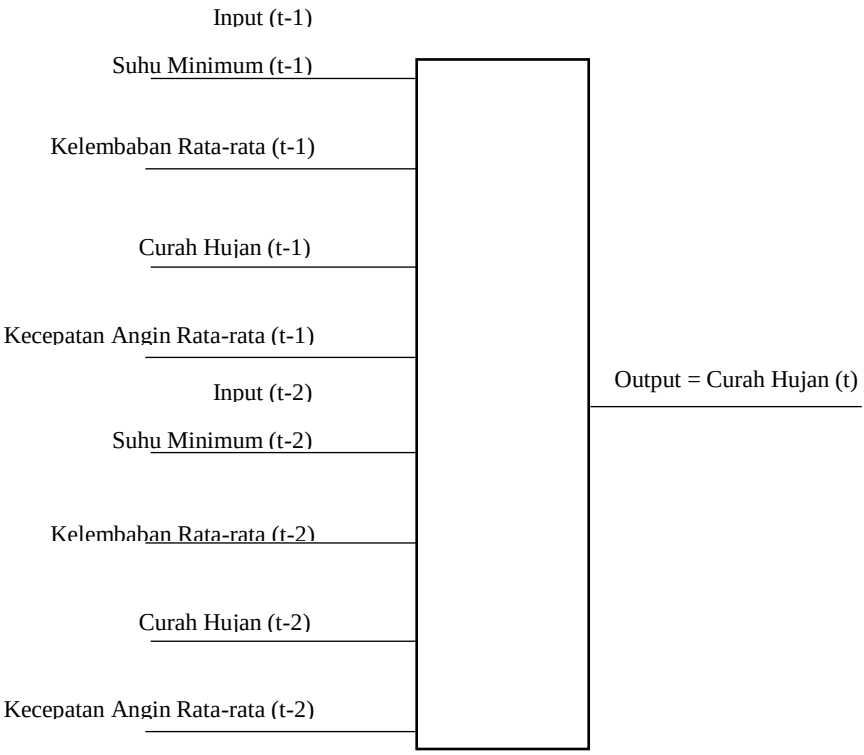
4.2.3 Penentuan Input Layer

Dalam tugas akhir ini, *input layer* memiliki data curah hujan. Namun ada beberapa input yang mendukung dalam curah hujan tersebut, yaitu suhu minimum, kelembaban rata-rata, curah hujan itu sendiri dan kecepatan angin rata-rata. Maka dari itu *input layer*



Gambar 4.9 Input Layer menggunakan 1 periode

adalah 4. Ada beberapa periode yang akan digunakan dalam *input layer* ini. Maksud dari periode itu sendiri adalah jumlah periode sebelumnya yang digunakan untuk meramalkan periode selanjutnya. 1 Periode berarti data yang digunakan adalah data 1 hari sebelumnya, 2 periode berarti digunakan data 1 hari + 2 hari sebelumnya, dan juga seterusnya. Untuk lebih jelasnya dapat dilihat pada Gambar 4.9 dan Gambar 4.10



Gambar 4.10 Input Layer menggunakan 2 periode

Pada Gambar 4.9 *input layer* menggunakan 1 periode sebelumnya, sedangkan pada Gambar 4.10 *input layer* menggunakan 2 periode sebelumnya sehingga ada 8 input yang dimasukkan, dibandingkan dengan Gambar 4.9 yang menggunakan 4 input saja.

4.2.4 Penentuan Hidden Layer

Jumlah node pada *hidden layer* ini dapat memberi pengaruh pada model *neural network.* Juga ada variasi node pada setiap periode, sebanyak n * 3 node [19]. Jumlah ini berdasarkan metode aturan praktis dalam menentukan jumlah neuron yang tepat dalam penggunaan di *hidden layer*. Isi dari peraturan praktis tersebut ialah "Jumlah *hidden neuron* sebaiknya berukuran 3 kali dari *input layer*, ditambah ukuran *output layer".* Dikarenakan *input layer* adalah 4, maka dimulai penghitungan node pada periode pertama juga adalah 4. Jadi misal pada periode 1, maka node yang digunakan adalah dari 4 hingga 4*3=12. Dalam periode 2 *input layer* nya adalah 8, sehingga nodenya mulai dari 8 hingga 24. Begitu pula seterusnya.

4.2.5 Penentuan Parameter

Agar didapatkan model yang terbaik diperlukan perubahan pada parameter yang ada. Tujuannya adalah untuk mencari nilai MAPE yang paling minimum.

Pengubahan parameter yang ada pada tugas akhir ini adalah sebagai berikut:

a. Fungsi Pelatihan (*training function*)
   Ada 3 *training function* yang digunakan pada tugas akhir ini yaitu *Levenerg-Marquardt backpropagation*, *Gradient descent with momentum and adaptive learning rate backpropagation*, dan *Gradient descent with adaptive learning rate backpropagation*

b. Fungsi Pembelajaran (*learning function*)
Untuk pencarian parameter *adaption learning function* digunakan parameter *learngdm*

c. Fungsi Transfer (*transfer function)*
Dilakukan percobaan untuk fungsi transfer dengan cara *trial and error* pada 3 jenis fungsi, yaitu Logsigmoid (*logsig*), Tansigmoid (*tansig*) dan juga Purelinear (*pureln*). Tujuan dari *transfer function* adalah untuk mendapatkan nilai dari hasil olah *weight* dengan *node*.

d. Fungsi Momentum (*momentum function*)
Pencarian nilai momentum yang akan berpengaruh pada performa model, dengan interval nilai 0.1. Momentum yang dicari adalah mulai dari 0.1 hingga 0.9.

e. Fungsi Learnrate (*learnrate function*)
Pencarian nilai learnrate, juga berpengaruh pada performa model. Nilai yang digunakan juga sama yaitu mulai dari 0.1 hingga 0.9 dan dengan interval 0.1.
Untuk lebih jelasnya pada penentuan *hidden layer* dan parameter yang digunakan pada penelitian tugas akhir ini, dapat dilihat pada Tabel 4.1

Tabel 4.1 Penentuan Hidden Layer dan Parameter

Jenis Parameter	Jumlah	Keterangan
Jumlah *Node* pada *Hidden Layer*	n – n*3 Node	*Trial and Error*
Fungsi Latih (*training function*)	3	*Trial and Error*
Fungsi Pembelajaran (*learning function*)	1	*Trial and Error*
Fungsi Transfer (*transfer function*)	3	*Trial and Error*
Fungsi Momentum (*momentum function*)	9	*Trial and Error*
Fungsi Learnrate (*learnrate function*)	9	*Trial and Error*

BAB V
IMPLEMENTASI

Di bab ini dijelaskan tentang implementasi model *Artificial Neural Network* dalam peramalan curah hujan. Juga akan dijelaskan tentang implementasi dari pencarian model hingga didapatkan model yang paling optimal dalam melakukan peramalan curah hujan.

5.1 Pengolahan Data

Data yang digunakan adalah data dari curah hujan dalam bentuk harian, di mana dalam satu hari ada 2 data. Data dari Januari 2012 – Januari 2017 berjumlah 3494. Untuk pengolahan data dibagi jadi dua, yaitu data pelatihan (*training*) dan juga data pengujian (*testing*). Data pelatihan digunakan sebesar 75% (2600 data) dan data pengujian sebesar 25% (894 data). Grafik-grafik di bawah akan menjelaskan mana data yang *training* dan mana data yang *testing*.

Gambar 5.1 menjelaskan tentang data suhu minimum yang digunakan untuk *training*. Data yang digunakan adalah 2600 data pertama dari suhu minimum. Sumbu x dari grafik ini adalah waktu, dan sumbu y adalah suhu dalam satuan derajat Celsius.

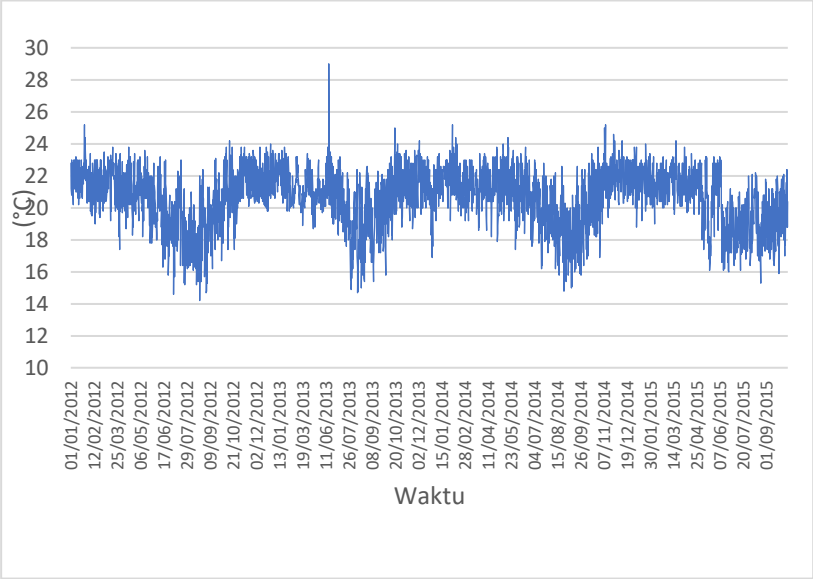
Pada gambar 5.2 dijelaskan tentang data kelembaban rata-rata yang digunakan untuk *training*. Dapat dilihat juga tren dari data yang naik turun sesuai dengan waktu pada musim. Data inipun juga digunakan 2600 data pertama. Sumbu x pada grafik ini adalah waktu dan sumbu y adalah persentase kelembaban rata-rata pada udara dalam satuan persen.

Pada gambar 5.3 dijelaskan kecepatan angin rata-rata, dan seperti sebelumnya dijelaskan, tidak terlalu terlihat tren data dari kecepatan angin. Karena naik-turunnya tidak sesuai dengan musim maupun waktu. Data yang digunakan adalah 2600 data pertama.

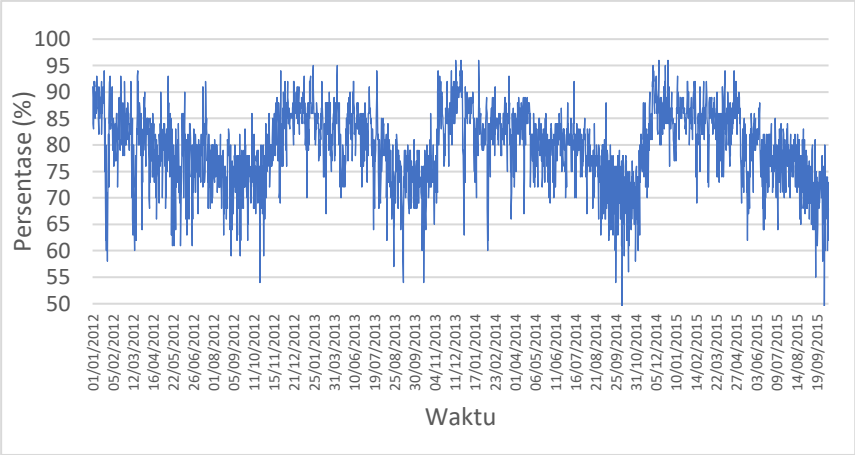
Sumbu x pada grafik adalah waktu dan sumbu y adalah nilai curah hujan dalam satuan milimeter (mm).

Gambar 5.4 menjelaskan tentang data *training* dari curah hujan. Tren data terlihat naik turun juga sesuai dengan musim kemarau maupun musim penghujan. Data *training* yang digunakan adalah 2600 data pertama. Sumbu x pada grafik adalah waktu sedangkan sumbu y adalah kecepatan angin rata-rata dengan satuan knot.

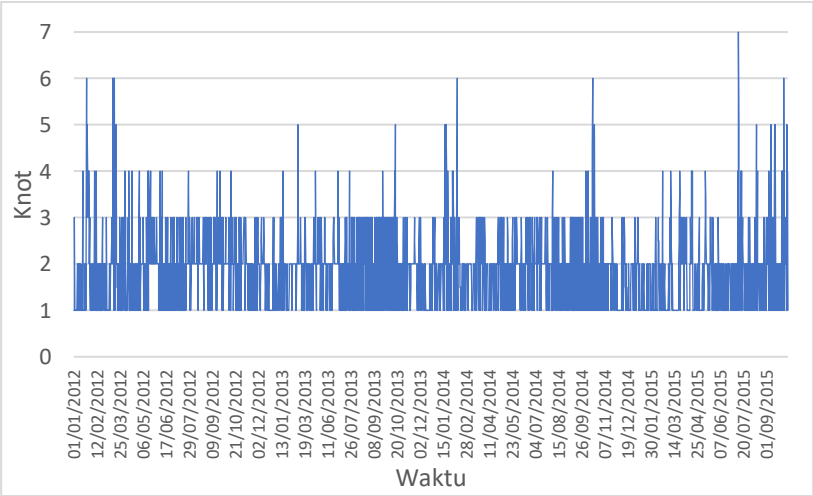
Seluruh data *training* diambil dari data yang telah dibagi sebelumnya, dan data-data tersebut akan menjadi acuan bagi program dalam melakukan proses *training* pada data. Selain data *training* juga ada data *testing* dengan prinsip yang sama namun tujuan yang berbeda karena data *testing* sesuai dengan namanya akan digunakan pada bagian *testing* model.



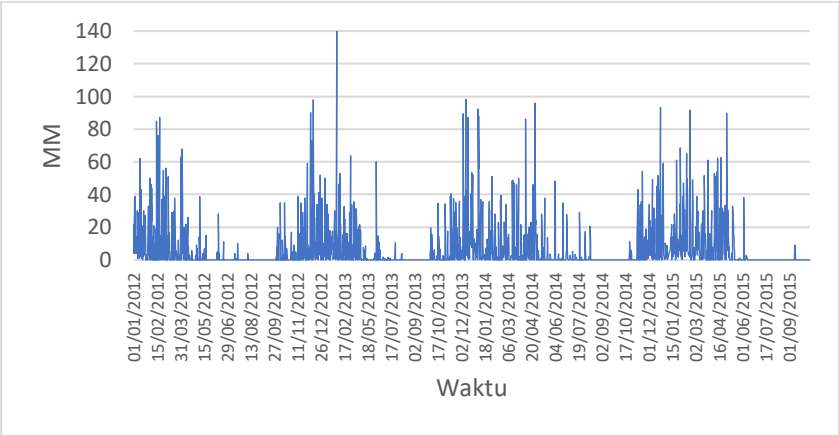
Gambar 5.1 Training Suhu Minimum



Gambar 5.2 Training Kelembaban Rata-Rata

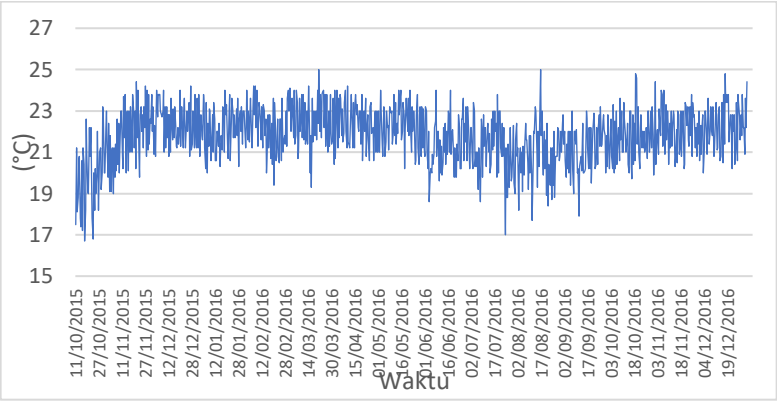


Gambar 5.3 Training Kecepatan Angin Rata-Rata

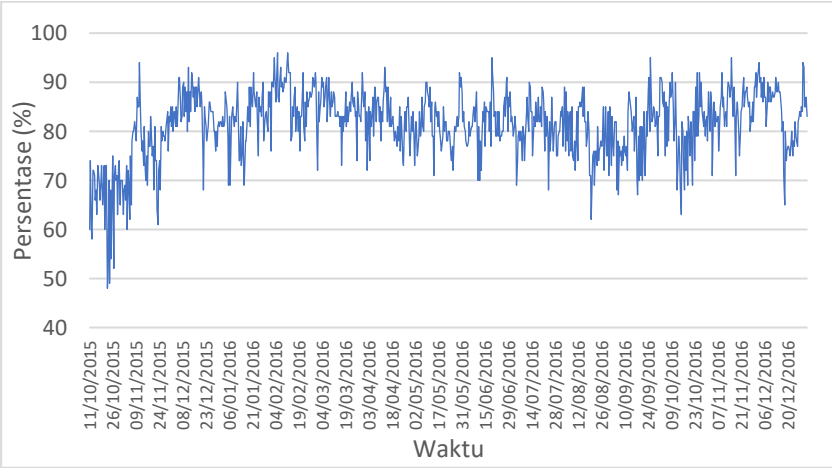


Gambar 5.4 Training Curah Hujan

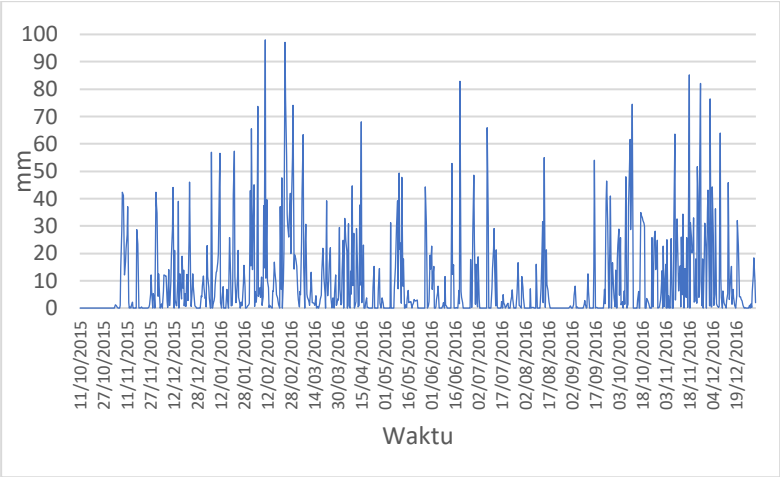
Untuk data *testing* dapat dilihat pada gambar 5.5, 5.6, 5.7 dan 5.8. Gambar 5.5 menjelaskan tentang data *testing* dari suhu minimum, gambar 5.6 menjelaskan kelembaban rata-rata, 5.7 tentang curah hujan dan 5.8 tentang kecepatan angin rata-rata. Seluruh data *testing* yang digunakan menggunakan 894 data terakhir. Sumbu x dari gambar 5.5 adalah waktu, dan sumbu y adalah suhu dalam satuan derajat Celsius. Sumbu x pada gambar 5.6 adalah waktu dan sumbu y adalah persentase kelembaban rata-rata pada udara dalam satuan persen. Sumbu x pada gambar 5.7 adalah waktu dan sumbu y adalah nilai curah hujan dalam satuan milimeter (mm). Sumbu x pada gambar 5.8 adalah waktu sedangkan sumbu y adalah kecepatan angin rata-rata dengan satuan knot.



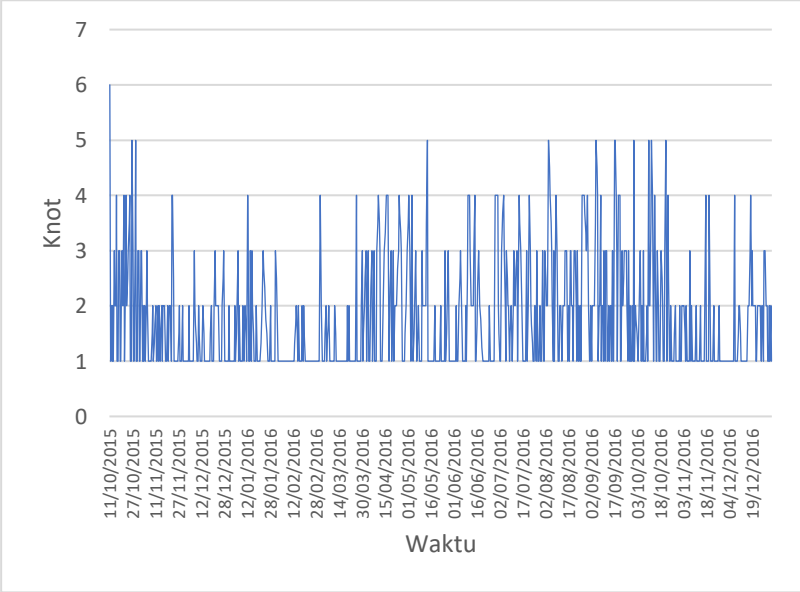
Gambar 5.5 Testing Suhu Minimum



Gambar 5.6 Testing Kelembaban Rata-Rata



Gambar 5.7 Testing Curah Hujan



Gambar 5.8 Testing Kecepatan Angin Rata-Rata

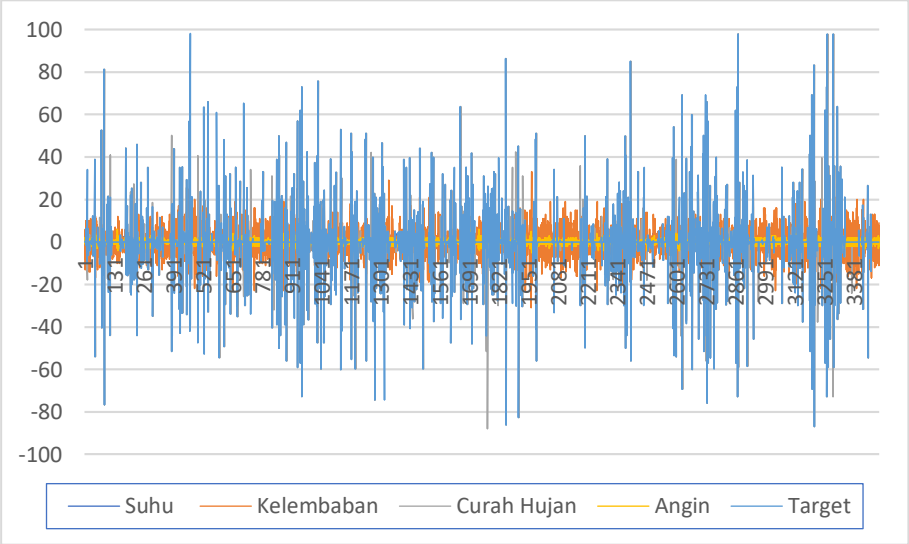
5.1.1 Bootstrapping Data

Setelah data dibagi berdasarkan *Training* dan *Testing*, data kemudian dimasukkan ke dalam program *Bootstrap*. Keempat data tersebut dimasukkan ke dalam file bernama 'datainput.xlsx' kemudian oleh program ini akan dijadikan data dalam bentuk bootstrap, dan disimpan dalam file bernama 'testdata1.xlsx'. Kemudian data yang setelah dibootstrap tersebut yang akan diinputkan ke dalam program pemodelan. Untuk lebih jelasnya dapat dilihat pada Script 5.1. Data *bootstrap* lengkap ada pada Lampiran B

```
% Filename: stationaryBB_new.m
% PURPOSE: demo of overlappingBB()
%          Time series bootstap: overlapping blocks
%-------------------------------------------------
% USAGE: overlappingBB_d
%-------------------------------------------------

clc; clear all; close all;

% Spanish Interbank Interest Rates: 1d, 1m, 3m, 6m, 1y

filename = 'datainput.xlsx';
Z = xlsread(filename);

% Filtering Z in order to achieve stationarity
dZ = filter([1 -1],1,Z);
dZ = dZ(2:end,:);


% Applying stationary boostrap (Romano-Politis): geometric pdf (sim=1) and
% expected block size L
sim = 1;
L = 10;
dZb = stationaryBB(dZ(:,:),sim,L);

% Plots
subplot(2,1,1);
plot(Z);
title('Original Data');
% axis tight;

subplot(2,1,2);
plot(dZb);
title('Bootstrapped Geometric Stationary');
axis tight;

filename = 'testdata1.xlsx';
xlswrite(filename,dZb,1,'A1:U3495')
```

Script 5.1 Program Boostrap

Hasil data setelah diboostrap akan berbentuk seperti ini, seperti yang terlihat dalam file ‘testdata1.xlsx’ pada Gambar 5.9



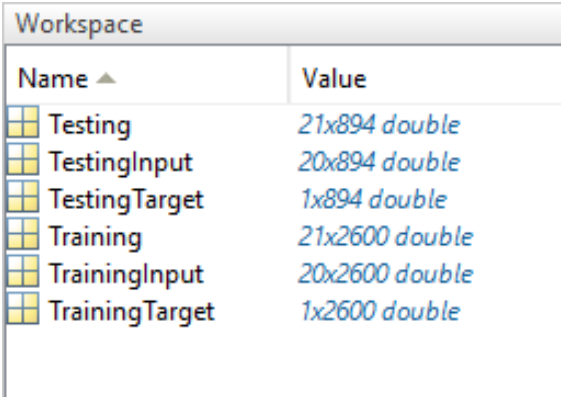
Gambar 5.9 Hasil data setelah di Boostrap

Data yang di*boostrap* juga di*transpose* dikarenakan program *MATLAB* membutuhkan data *timeseries* agar berbentuk semakin ke kanan, bukan ke bawah. Sehingga dilakukan *transpose* di mana yang baris adalah data-data dan kolom adalah waktu.

5.1.2 Pembuatan Dataset Untuk Model

Dalam menjalankan program, dibutuhkan data input dan output yang telah di*bootstrap* dan dipisah sesuai dengan kegunaannya. Dapat dilihat pada Gambar 5.10. Matrix Testing berisi keseluruhan data testing, dengan besar matrix 21x894. Matrix TestingInput

berisi nilai input untuk testing, dan TestingTarget berisi target dari testing. Semua nilai ini juga berada di matrix testing. Untuk matrix Training juga sama, berisi data Training. Bedanya hanya yang digunakan adalah data Training dan luas matrix juga lebih luas dikarenakan data Training yang berjumlah 2600 dibandingkan dengan data Testing yang berjumlah 894 data.



Gambar 5.10 Pembagian Dataset

Gambar 5.10 menggunakan contoh dari dataset pada periode 5. Sehingga nilai dari *array matrix* nya besar, dikarenakan input datanya pun besar, notabene input data berada pada periode 5. Ini akan berbeda-beda tiap periode dikarenakan setiap periode berbeda besar jumlah *input* datanya.

5.2 Pembuatan Model Artificial Neural Network

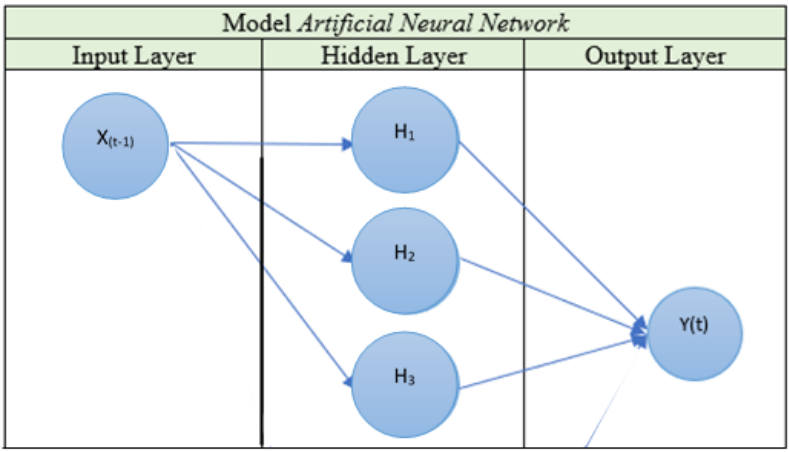
Di penelitian tugas akhir kali ini, model dari *neural network* yang digunakan adalah berjenis *multilayer perceptron* yang memiliki bagian *input layer, hidden layer* dan *output layer.*

5.2.1 Model Artificial Neural Network

Dalam penelitian tugas akhir ini model *Artificial Neural Network* terdiri dari *input layer* yang isinya adalah neuron-neuron data-data curah hujan di masa lalu, *hidden layer* yang terdiri dari satu *layer* berfungsi aktifasi dan *output layer* yang berisikan target curah hujan. Persamaannya adalah:

$$Y(t) = f(x(t-1), x(t-2), x(t-3), x(t-4), x(t-5))$$

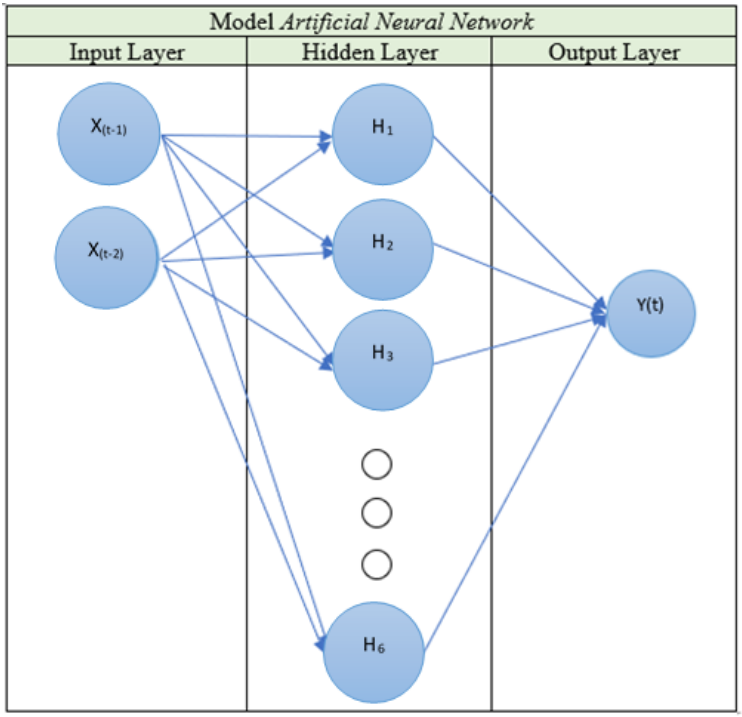
Hidden layer memiliki nilai yang bergerak mulai dari neuron 4 pada periode 1 hingga neuron 60 pada periode 5. (H_4, H_5, H_6, H_7, ..., H_{60}). Pada penelitian tugas akhir, hanya digunakan hingga 5 periode dikarenakan *running time* yang cukup lama dimana pada *running time* 5 periode membutuhkan hingga waktu 48 jam.
Model *Artificial Neural Network* yang digunakan di penelitian tugas akhir ini adalah seperti yang terlihat di Gambar 5.11 hingga 5.15. Gambar model ANN ini digunakan dengan referensi konsep jaringan syaraf tiruan peramalan [13].



Gambar 5.11 Model Artificial Neural Network 1 Periode

Keterangan dari Gambar 5.11 adalah:

$X_{(t-1)}$ = data 1 periode sebelumnya
H_1 = Node pada hidden layer ke 1
H_2 = Node pada hidden layer ke 2
H_3 = Node pada hidden layer ke 3

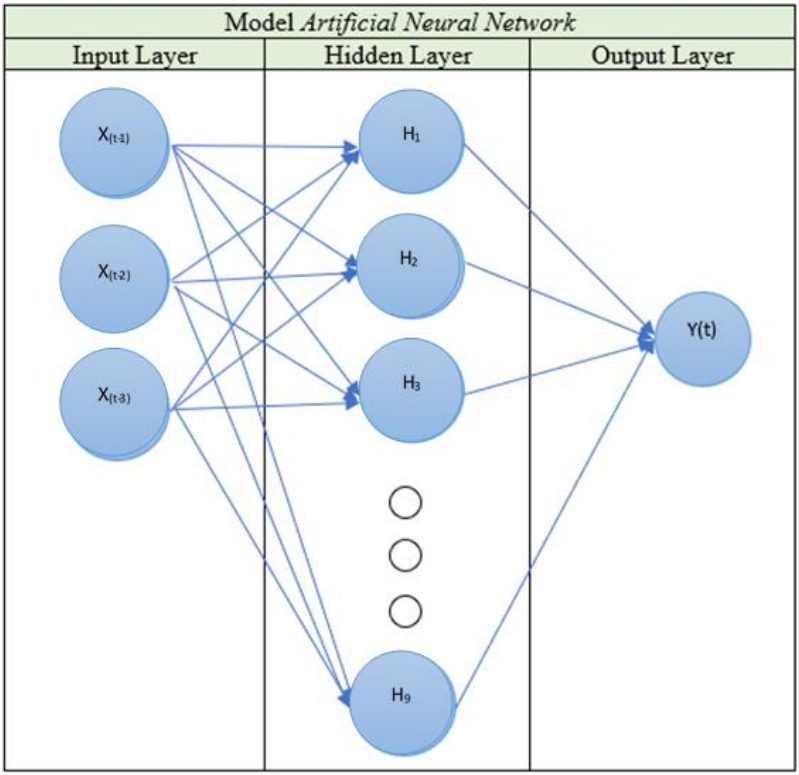


Gambar 5.12 Model Artificial Neural Network 2 Periode

Keterangan dari Gambar 5.12 adalah:
$X_{(t-1)}$ = data 1 periode sebelumnya
$X_{(t-2)}$ = data 2 periode sebelumnya
H_1 = Node pada hidden layer ke 1
H_2 = Node pada hidden layer ke 2

H_3 = Node pada hidden layer ke 3

H_6 = Node pada hidden layer ke 6



Gambar 5.13 Model Artificial Neural Network 3 Periode

Keterangan dari Gambar 5.13 adalah:

$X_{(t-1)}$ = data 1 periode sebelumnya

$X_{(t-2)}$ = data 2 periode sebelumnya

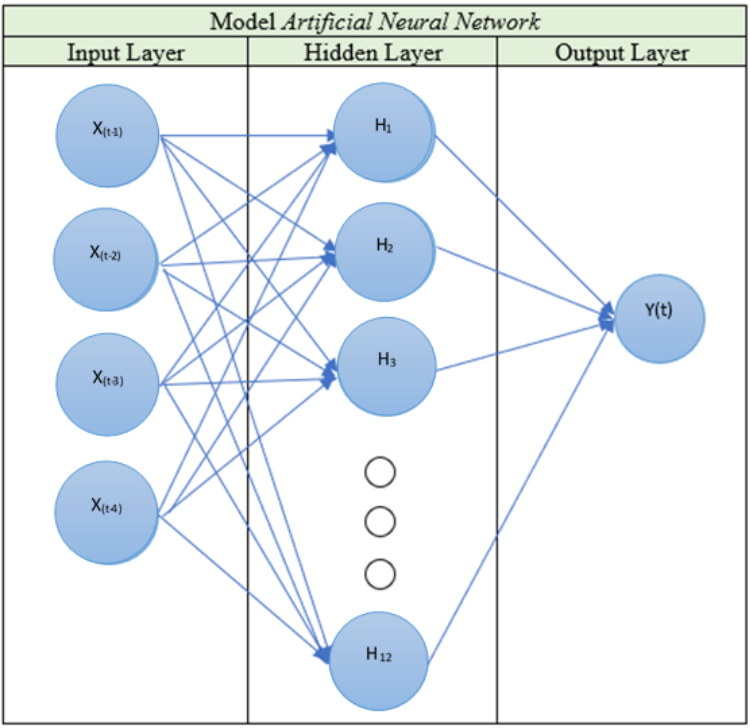
$X_{(t-3)}$ = data 3 periode sebelumnya

H_1 = Node pada hidden layer ke 1

H_2 = Node pada hidden layer ke 2

H_3 = Node pada hidden layer ke 3

H_9 = Node pada hidden layer ke 9



Gambar 5.14 Model Artificial Neural Network 4 Periode

Keterangan dari Gambar 5.14 adalah:

$X_{(t-1)}$ = data 1 periode sebelumnya

$X_{(t-2)}$ = data 2 periode sebelumnya

$X_{(t-3)}$ = data 3 periode sebelumnya

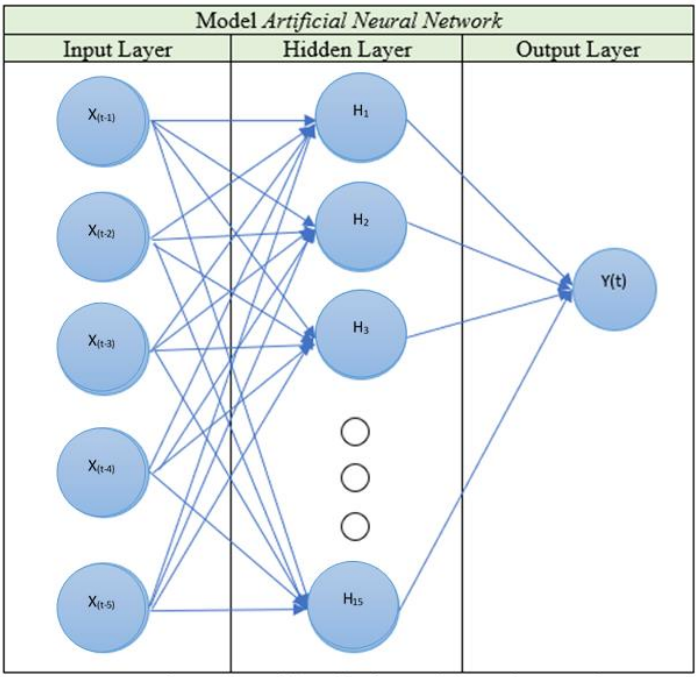
$X_{(t-4)}$ = data 4 periode sebelumnya

H_1 = Node pada hidden layer ke 1

H_2 = Node pada hidden layer ke 2

H_3 = Node pada hidden layer ke 3

H_{12} = Node pada hidden layer ke 12



Gambar 5.15 Model Artificial Neural Network 5 Periode

Keterangan dari gambar 5.15 adalah:

$X_{(t-1)}$ = data 1 periode sebelumnya

$X_{(t-2)}$ = data 2 periode sebelumnya

$X_{(t-3)}$ = data 3 periode sebelumnya

$X_{(t-4)}$ = data 4 periode sebelumnya

$X_{(t-5)}$ = data 5 periode sebelumnya

H_1 = Node pada hidden layer ke 1

H_2 = Node pada hidden layer ke 2

H_3 = Node pada hidden layer ke 3

H_{15} = Node pada hidden layer ke 15

5.2.2 Data Input

Data dalam penelitian tugas akhir ini menggunakan data *input* dari data curah hujan periode Januari 2012 – Januari 2017 yang terdokumentasikan oleh Stasiun Meteorologi dari Kabupaten Malang, dan data diambil dari Badan Pusat Statistik. Isi data beragam namun yang digunakan adalah Suhu Minimum, Kelembaban Rata-rata, Curah Hujan (mm) dan Kecepatan Angin Rata-rata. Untuk lebih jelasnya tentang *preprocessing* data yang dilakukan untuk data *input* telah dijelaskan pada bab sebelumnya.

Untuk pengubahan *input layer*, ini disesuaikan dengan periode yang digunakan. Lebih jelasnya dapat dilihat di Tabel 5.1 dan 5.2

Tabel 5.1 adalah input data periode pertama (sebelum dilakukan *bootstrap*), berjumlah 4 kolom. Ditambah 1 kolom lagi di sebelahnya yang berupa kolom ‘target’. Kolom ini berisikan data dari kolom ‘Curah hujan’ namun dikurangi 1 kolom yang paling atas. Jadi total ada 5 kolom.

Tabel 5.1 Data Input 1 Periode

Suhu Minimum	Kelembaban Rata-rata	Curah Hujan	Kecepatan Angin Rata-rata	Target
22,8	91	22	1	4
21,2	89	4	1	38
20,8	84	38	3	4
23	83	4	1	22,5
20,9	90	22,5	1	39
22,8	92	39	1	4

Suhu Minimum	Kelembaban Rata-rata	Curah Hujan	Kecepatan Angin Rata-rata	Target
22	91	4	1	5
21,3	92	5	1	4
20,2	85	4	1	14
22,8	89	14	1	13
…	…	…	…	…

Berikutnya adalah periode 2, dapat dilihat isi data pada tabel 5.2. Dalam Tabel 5.2, ada 9 kolom. 8 kolom pertama sebenarnya serupa dengan periode 1, namun dikarenakan periode 2 menggunakan 2 periode sebelumnya, sehingga 4 kolom yang ke dua (kolom 5-8) adalah nilainya sama dengan 4 kolom pertama (kolom 1-4) namun nilai yang paling atas dihilangkan. Kemudian untuk kolom target juga sama seperti tabel 5.1, menggunakan nilai dari curah hujan, namun kali ini 2 kolom teratas dihilangkan. Jadi jika dibandingkan dengan target pada tabel 5.1 yang memulai dari nilai 4, target pada tabel 5.2 memulai dari nilai bawahnya, yaitu 38. Dan hal ini berlanjut hingga periode-periode selanjutnya, dengan cara yang sama. Semakin banyak periodenya maka kolom akan bertambah 4 lagi dan target yang digunakan juga merupakan target bawahnya.

Tabel 5.2 Data Input 2 Periode

SM	KR	CH	KAR	SM	KR	CH	KAR	TRGT
22,8	91	22	1	21,2	89	4	1	38
21,2	89	4	1	20,8	84	38	3	4
20,8	84	38	3	23	83	4	1	22,5

SM	KR	CH	KAR	SM	KR	CH	KAR	TRGT
23	83	4	1	20,9	90	22,5	1	39
20,9	90	22,5	1	22,8	92	39	1	4
22,8	92	39	1	22	91	4	1	5
22	91	4	1	21,3	92	5	1	4
21,3	92	5	1	20,2	85	4	1	14
20,2	85	4	1	22,8	89	14	1	13
22,8	89	14	1	21,2	90	13	1	8
21,2	90	13	1	23	86	8	1	16
23	86	8	1	22,8	88	16	1	22
22,8	88	16	1	21,2	86	22	2	30
…	…	…	…	…	…	…	…	…

5.2.3 Data Output

Data output adalah data-data yang berada dalam kolom ‘target’, seperti yang terlihat pada tabel 5.1 dan tabel 5.2. Penjelasan dari data output ini adalah data input pada periode n+1. Seperti yang ditunjukkan pada tabel 5.1, kolom ‘target’, yang merupakan data output, isinya adalah data input ‘curah hujan’ namun dilebihi satu kolom. Ini artinya data input adalah data ke-n, dan outputnya adalah data ke n+1. Begitu juga dengan periode selanjutnya. Jika data input adalah data ke-n+1, maka outputnya n+2.

Data output menggunakan kolom dari data input ‘Curah Hujan’, dikarenakan target dari peramalan ini adalah curah hujan.

5.2.4 Parameter yang Digunakan

Pembuatan model dalam tugas akhir kali ini menggunakan beberapa parameter yang ditentukan yaitu Fungsi Pelatihan (*training function*), Fungsi Pembelajaran (*learning function*), Fungsi Transfer (*transfer function*), Fungsi Momentum (*momentum function)* dan Fungsi Learnrate (*learnrate function*). Untuk lebih jelasnya tentang masing-masing parameter ini, sudah dijelaskan di bab sebelumnya.

5.3 Penerapan Model

Pada pembuatan model dilakukan pembuatan *script*, dengan tujuan untuk mengubah-ubah parameter yang ditentukan secara otomatis. Sehingga dapat ditemukan model yang paling optimal dari semua model yang ditemukan, dan dengan basis percobaan *trial and error*. Script dibuat dalam bentuk file .m yang dijalankan pada *software* MATLAB. Hasilnya akan keluar dalam bentuk model-model file .m dan juga dalam bentuk ringkasan file .csv.

5.3.1 Penerapan Model Neural Network

Pembuatan model *neural network* dilakukan dengan menggunakan *script* untuk mengubah-ubah parameter, seperti yang sudah dijelaskan sebelumnya. Isi dari parameter dideklarasikan pada kode untuk dipanggil oleh program Matlab. Berikut parameter-parameter yang diubah beserta dengan *script*nya, ditunjukkan oleh Script 5.2.

Isi dari parameter berhubungan dengan masing-masing fungsinya, sehingga proses pencarian model akan bergantung pada isi dari parameter-parameter tersebut. Parameter ini berasal dari *function* Matlab yang dapat dimanfaatkan dalam penghitungan dan pencarian model. Untuk fungsi momentum dan fungsi learnrate ditentukan interval sebanyak 0.1 dan dimulai dari 0.1 hingga 0.9. Bagian akhir dari script menunjukkan nilai pengembalian bagi masing-masing fungsi mulai dari TrainFunc hingga LearnrateFunc.

Nilai pengembaliannya dijadikan variabel hutuf A hingga E. Kemudian nilai-nilai tersebut dibaca dengan *function* numel.

```
%perubahan parameter

%pergantian fungsi Training
TrainFunc={'trainlm' 'traingdx' 'traingda'};
%pergantian fungsi Learning
LearnFunc={'learngdm'};
%pergantian fungsi Transfer
TransFunc={'logsig' 'tansig' 'purelin'};
%pergantian fungsi Momentum
MomenFunc=[0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9];
%pergantian fungsi Learnrate
LearnrateFunc=[0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9];

A=numel(TrainFunc);
B=numel(LearnFunc);
C=numel(TransFunc);
D=numel(MomenFunc);
E=numel(LearnrateFunc);
```

Script 5.2 Script Perubahan Parameter

Penjelasan dari Script 5.2 dapat dilihat pada Tabel 5.3.

Tabel 5.3 Penjelasan Script Perubahan Parameter

Script	Fungsi
TrainFunc	Membaca Fungsi Pelatihan (*training function*)
LearnFunc	Membaca Fungsi Pembelajaran (*learning function*)
TransFunc	Membaca Fungsi Transfer (*transfer function*)

Script	Fungsi
MomenFunc	Mengganti nilai Fungsi Momentum (*momentum function*)
LearnrateFunc	Mengganti nilai fungsi Learnrate (*learnrate function*)
Numel	Membaca jumlah elemen dalam array

Lalu untuk script dari pembuatan struktur *multilayer perceptron neural network* bentuknya seperti yang ada di bawah ini, pada Script 5.3

```
%Membuat Jaringan Model Baru dengan Parameter yang
%diset
net=newff(TrainingInput, TrainingTarget, node, {cell2mat(TransFunc(c)),'purelin'});
net.trainFcn=cell2mat(TrainFunc(a));
net.trainParam.epochs=1500;
net.trainParam.lr=LearnrateFunc(e);
net.trainParam.mc=MomenFunc(d);
net.trainParam.max_fail=1500;
net.performFcn='mse';
net.layerWeights{1,1}.learnFcn=cell2mat(LearnFunc(b));

net.trainParam.showWindow = false;
net.trainParam.showCommandLine = true;
```

Script 5.3 Multilayer Percetron Neural Network

Untuk Script 5.3 penjelasan berada pada Tabel 5.4.

Tabel 5.4 Penjelasan Script Multilayer Perceptron Neural Network

Script	Fungsi
Net	Membuat jaringan (*network*)
Newff	Jaringan *feedforward(ff)-backpropagation*
TrainingInput	Membaca data set berisi *input* data dari bagian *training*
TrainingTarget	Membaca data set berisi *target* data data bagian *training*

Script	Fungsi
cell2mat(TransFunc(c))	Membaca parameter dari *transfer function* untuk *hidden layer* (c karena parameter TransFunc di script sebelumnya menggunakan variable c)
purelin	Deklarasi *transfer function* untuk bagian *output layer* adalah menggunakan fungsi *purelin*
net.trainFcn	Melakukan *training* pada data
cell2mat(TrainFunc(a))	Membaca parameter dari *train function* untuk *hidden layer* (a)
net.trainParam.epochs	Menentukan total *epoch* yang akan digunakan pada jaringan (1500)
net.trainParam.lr	Melakukan *learnrate function* pada data
LearnrateFunc(e)	Membaca elemen parameter *learnrate function*
net.trainParam.mc	Melakukan *training* dengan parameter *momentum function*
MomenFunc(d)	Membaca elemen parameter *momentum function*
net.trainParam.max_fail =1500	Menentukan total nilai maksimum kegagalan 1500 sesuai dengan epoch
net.performFcn	Menentukan perhitungan dari uji performa, jenis mana yang digunakan
mse	*Mean Squared Error.* Salah satu jenis uji performa

Script	Fungsi
net.layerWeights{1,1} .learnFcn	Menentukan parameter dari *learning function* yang digunakan sekarang
cell2mat(LearnFunc(b))	Membaca parameter dari *learn function* untuk *hidden layer* (b)
net.trainParam.showWindow	Menunjukkan jendela untuk menunjukkan hasil latih atau tidak
net.trainParam. showCommandLine	Menunjukkan hasil latih pada *command line* di *software* MATLAB

Script 5.3 berfungsi sebagai pembuat struktur dan jaringan yang telah disesuaikan dengan parameter. Parameter-parameter yang diset oleh program akan dibaca oleh script bagian ini dan disesuaikan oleh program. TrainingInput dan TrainingTarget adalah fungsi untuk membaca data set Training di bagian *input* dan *target* nya. Fungsi fungsi yang lainnya disesuaikan dengan parameter yang sedang disetel pada model yang sedang dibuat. Cara penyesuaiannya adalah program membaca variabel mana yang sesuai dengan fungsi dan variabel tersebut sedang menggunakan parameter yang mana pada saat ini. Kemudian setelah proses pembuatan model saat itu selesai, parameter akan berganti dengan nilai selanjutnya dan variabel akan membaca pergantian itu. Dan proses ini akan terus berlanjut hingga program menyelesaikan seluruh perhitungan. Perhitungan yang dilakukan ditentukan oleh nilai yang diinputkan, sebagai contoh pada gambar Script 5.4 jumlah node yang diinputkan adalah dari node 51 hingga 52. Ini berarti program akan terus menghitung dari model-model pada node 51 dan 52, dan selesai sampai node 52. Program tidak akan meneruskan hingga node 53 karena dalam script tidak dideklarasikan untuk menghitung hingga node 53. Program baru akan menghitung node 53 jika telah dideklarasikan sebelumnya.

Selain itu ada *script* untuk menjalankan program tersebut dengan fungsi sebagai *nested looping*, tujuannya agar dapat membuat otomasi pada seluruh iterasi. Masing-masing parameter dan variabel sudah dijelaskan dalam gambar Script 5.4

```
%Nested Loop untuk Otomasi Pembuatan Model
%Pergantian Node
for node=51:52
    %Pergantian Train Function
    for a=1:A
        %Pergantian Learn Function
        for b=1:B
            %Pergantian Transfer Function
            for c=1:C
                %Pergantian Nilai Fungsi Momentum
                for d=1:D
                    %Pergantian Nilai Fungsi Learnrate
                    for e=1:E
```

Script 5.4 Script untuk Otomasi Iterasi (Nested Looping)

5.3.2 Proses *Training* dan *Testing*

Di bagian proses ini, script yang digunakan adalah untuk melakukan proses *training* dan *testing* pada data. Sebagaimana ditunjukkan oleh Script 5.5

```
%Membuat network dan Penyimpanan Training
[netTrain,tr]=train(net,TrainingInput,TrainingTarget);

%Hasil Output dari Input Training Data
TrainResult=netTrain(TrainingInput);
%Hasil Output dari Input Testing Data
SimResult=sim(netTrain,TestingInput);
```

Script 5.5 Script untuk Training dan Testing

Penjelasan untuk script 5.5 dapat dilihat pada Tabel 5.5

Tabel 5.5 Penjelasan Script Training dan Testing

Script	Fungsi
[netTrain,tr]	Melakukan pengambilan model network dan juga penyimpanan model
train	Melakukan *training*
TrainingInput	Fungsi pembaca dataset yang berisikan data *input* proses *training*
TrainingTarget	Fungsi pembaca dataset yang berisikan data *target (output)* proses *training*
TrainResult	Hasil *output* dari *input training data*
sim	Melakukan *simulation (testing)*
SimResult	Hasil *output* dari *input testing data*
TestingInput	Fungsi pembaca dataset yang berisikan data *input* proses *testing*

5.3.3 Uji Performa

Hasil dari proses *Training* dan *Testing* dapat dihitung menjadi nilai Error, MSE dan MAPE. Uji performa ini dilakukan untuk melihat tingkat *error* yang dihasilkan oleh model. Scriptnya adalah seperti pada Script 5.6. Dalam Script 5.6 terlihat cara penghitungan nilai dari error, MSE dan MAPE. Nilai dari TrainError sendiri adalah hasil dari TrainResult dikurangi dengan TrainingTarget. Kemudian nilai hasil dari mseTrain adalah nilai TrainError yang dilakukan

fungsi mse dari matlab. Penjelasan dari Script 5.6 ada di Tabel 5.6, dan beberapa ada dari Tabel 5.5 sebelumnya. Hasil dari semua proses disimpan dalam bentuk file excel (.csv) agar dapat dianalisis dengan lebih mudah. Script untuk proses penyimpanan dapat dilihat pada Script 5.7. Penyimpanan dilakukan bersamaan dengan penghitungan nilai error pada hasil training maupun testing. Hasil dari nilai-nilai inilah yang akan dibandingkan nilainya untuk dicari nilai error yang paling minimum.

Tabel 5.6 Penjelasan Script Uji Performa

Script	Fungsi
TrainError	Fungsi sebagai nilai *error* dari hasil data *training*
mse	Menghitung *mean-squared error* dari *array* yang ditentukan
mean	Menghitung rata-rata nilai dari suatu *array*
mseTrain	Fungsi berisi nilai MSE dari data *training*
mapeTrain	Fungsi berisi nilai MAPE dari data *training*
mseTest	Fungsi berisi nilai MSE dari data *testing*
mapeTest	Fungsi berisi nilai MAPE dari data *testing*
TestingTarget	Fungsi pembaca dataset yang berisikan data *output/target* proses *testing*

Script 5.6 dan 5.7 berisi fungsi-fungsi yang memiliki tujuan utama untuk melakukan penghitungan pada *function* yang telah dideklarasikan pada program. Isi dari script dapat dilihat pada gambar Script 5.6 dan Script 5.7

```
%Menghitung Nilai dari Error, MSE, MAPE
%Output Training dan Output Testing
TrainError=TrainResult - TrainingTarget;

mseTrain=mse(TrainError);
mseTrain;

mapeTrain=(abs(TrainingTarget-TrainResult))./TrainingTarget;
mapeTrain=mean(mapeTrain);
mapeTrain=mapeTrain*100;

TestError=SimResult - TestingTarget;

mseTest=mse(TestError);
mseTest;

mapeTest=(abs(TestingTarget-SimResult))./TestingTarget;
mapeTest=mean(mapeTest);
mapeTest=mapeTest*100;
```

Script 5.6 Script untuk Uji Performa

```
%Simpan hasil jaringan, output training & testing
TrainResult=transpose(TrainResult);
SimResult=transpose(SimResult);
ModelNameNet=[num2str(a),'_',num2str(b),'_',num2str(c),
    '_', num2str(d),'_',num2str(e),'_',num2str(node),'netTrain','.mat'];
save(ModelNameNet, 'netTrain');

%Simpan hasil Error, output dari training
errorTrain=transpose(errorTrain);

%Simpan hasil Error, MSE output dari Testing
errorTest=transpose(errorTest);

%Penggantian Variabel Input untuk dimasukkan di Excel
zl = [num2str(a), '_', num2str(b), '_', num2str(c), '_',
    num2str(d), '_', num2str(e), '_', num2str(node)];
```

Script 5.7 Script untuk Penyimpanan Hasil

Selain pada Script 5.7, ada beberapa fungsi *header* yang dimasukkan dengan tujuan mempermudah pencarian data yang

telah dimasukkan dalam file *excel*. Untuk lebih jelasnya dapat dilihat pada Script 5.8. Dalam Script 5.8, program akan menuliskan data hasil performa pada tempat yang sesuai sehingga hasil dari model dapat dibaca dengan mudah pada file *excel*. Script 5.8 dilakukan dengan tujuan untuk mengumpulkan dan memberi label pada nilai-nilai *error* sehingga dapat diketahui mana nilai *error* dari hasil latih MSE, hasil uji MSE, hasil latih MAPE serta hasil uji MAPE. Pada script juga diberikan perintah untuk menempatkan nilai yang baru pada bagian paling bawah dalam dokumen excel, sehingga nilai yang tertulis akan memanjang kebawah. Script 5.8 memiliki beberapa header seperti ResultHeader, TrainResult, errorTrain, SimResult dan errorTest dengan kode masing-masing adalah A1, A2, B2, C2 dan D2. Maksud dari kombinasi kode huruf dan angka ini adalah letak sel (*cell*) dimana excel memasukkan data. Jadi ResultHeader dimasukkan pada sel A1, TrainResult pada A2, errorTrain pada B2, SimResult pada C2 dan errorTest pada D2. TrainResult dan errorTrain berisikan data hasil *training* serta *error training*, sedangkan SimResult dan errorTest berisi data hasil *testing* dan *error testing*.

```
%Membuat Excel berisi hasil Train dan Test model,
%dengan header
ResultHeader = {'Training', 'Error Training', 'Testing', 'Error Testing'};
xlswrite1('HasilHidden.csv',ResultHeader,z1,'A1');
xlswrite1('HasilHidden.csv',TrainResult,z1,'A2');
xlswrite1('HasilHidden.csv',errorTrain,z1,'B2');
xlswrite1('HasilHidden.csv',SimResult,z1,'C2');
xlswrite1('HasilHidden.csv',errorTest,z1,'D2');

%Simpan hasil performa model di excel
z2 = [num2str(a), '_', num2str(b), '_', num2str(c), '_',
    num2str(d), '_', num2str(e), '_', num2str(node)];
nilai = {z2, mseTrain, mseTest, mapeTrain, mapeTest};

%Excel kumpulan hasil performa, dengan header
PerformHeader = {'Kode Model', 'Training MSE', 'Test MSE', 'Training MAPE', 'Test MAPE'};
xlswrite1('HasilHidden.csv',PerformHeader,'HasilPerforma','A1');
N=size(xlsread1('HasilHidden.csv','HasilPerforma'),1);

%Nilai baru di baris paling bawah excel
AA=strcat('A',num2str(N+2));
xlswrite1('HasilHidden.csv',nilai,'HasilPerforma',AA);
```

Script 5.8 Script untuk Penyimpanan Hasil, Performa beserta Header

BAB VI
HASIL DAN PEMBAHASAN

Di bab ini dijelaskan tentang hasil dan pembahasan terkait dengan peramalan curah hujan. Juga dijelaskan penentuan model *Artificial Neural Network* yang paling optimal dan yang tepat digunakan dalam peramalan curah hujan.

6.1 Lingkungan Uji Coba

Data yang digunakan adalah data dari curah hujan dalam bentuk harian, di mana dalam satu hari ada 2 data. Data dari Januari 2012 – Januari 2017 berjumlah 3494. Untuk pengolahan data dibagi jadi dua, yaitu data pelatihan (*training*) dan juga data pengujian (*testing*). Data pelatihan digunakan sebesar 75% (2600 data) dan data pengujian sebesar 25% (894 data). Grafik-grafik di bawah akan menjelaskan mana data yang *training* dan mana data yang *testing*.

Lingkungan uji coba adalah lingkungan pengujian ketika menguji model yang dibuat dalam penelitian. Dalam lingkungan uji coba ada dua hal yaitu perangkat keras (*hardware*) dan perangkat lunak (*software*). Perangkat keras yang digunakan dalam pengerjaan tugas akhir adalah seperti pada Tabel 6.1.

Tabel 6.1 Perangkat Keras dalam Lingkungan Uji Coba

Perangkat Keras	Spesifikasi
Jenis	Laptop, PC, Mini-PC
Processor	Intel Core i7 7700HQ, i5 8400, i5 3230M
RAM	8 GB, 16 GB, 4 GB
VGA	Nvidia GeForce GTX 1050 Ti, GTX 1060, Intel HD Graphics 4000

Harddisk Drive	1 TB, 1 TB, 500GB

Kemudian untuk perangkat lunak yang digunakan dalam lingkungan uji coba adalah seperti pada Tabel 6.2

Tabel 6.2 Perangkat Lunak dalam Lingkungan Uji Coba

Perangkat Lunak	Fungsi
Windows 10	Sistem Operasi (OS)
Matlab R2016a (9.0.0.341360)	Membuat dan membentuk model, menjalankan otomasi iterasi, melakukan pelatihan dan pengujian pada model.
Microsoft Office Excel Version 1805	Memasukkan data, mengelola dan merangkum data

6.2 Percobaan Parameter

Di penelitian tugas akhir ini, dilakukan percobaan pengubahan pada parameter-parameter. Ini sudah dijelaskan sebelumnya pada bab 4 dan bab 5. Namun ada beberapa yang akan dijelaskan lebih lanjut pada subbab berikut.

6.2.1 Pengujian Epoch

Pada penelitian tugas akhir kali ini, *epoch* ditentukan nilainya 1500. *Epoch* sendiri gunanya adalah untuk membatasi iterasi ketika sedang dilakukan pencarian model. Ketika dilakukan uji coba, sebagian model berhenti iterasi sebelum mencapai *epoch* 1500. Namun ada juga yang belum selesai ketika sudah mencapai *epoch* 1500. Ini dikarenakan sistem otomatis menghentikan iterasi ketika nilai *error* pada model yang sedang dalam proses menunjukkan

kenaikan. Lalu akan dipilih nilai *error* yang paling rendah dalam sistem.

6.3 Percobaan Model

Di bagian ini dijelaskan proses dari percobaan model yang dilakukan pada tugas akhir. Ada kombinasi 1-5 node pada *input layer*. *Input layer* ini berbeda-beda setiap nodenya. *Input layer* pada node 1 adalah 4, sedangkan untuk node 2 *input layernya* adalah 8. Untuk *output layer (target)* adalah selalu 1 *layer* namun dimulai dari nilai yang berbeda tergantung periode yang digunakan.

Selanjutnya akan dibahas hasil analisis dari tiap node pada *input layer* dengan cara membandingkan nilai MSE dari tiap-tiap model.

6.3.1 Model dengan Jumlah Node Input 1 Periode Sebelumnya

Pada percobaan yang pertama node *input layer* nya adalah 4. Ini berarti periode yang digunakan adalah hanya 1, namun karena input datanya 4 kolom jadi node *input layer* nya 4. Hasil dari percobaan ini berupa tabel yang berisi hitungan MSE serta kombinasi model dengan parameter apa saja yang digunakan. Langkah pertama adalah mencari nilai dari Test MSE yang terkecil, dan mengetahui kombinasi model manakah yang merupakan model yang memiliki nilai Test MSE tersebut. Nilai Test MSE digunakan karena hasil tersebut yang dapat diuji menjadi ramalan curah hujan, serta nilai tersebut dapat dihitung oleh rumus penghitungan MSE. Dalam penelitian tugas akhir ini ada 728 buah kombinasi model dari setiap node, sedangkan pada periode pertama ada 8 node mulai dari node 4 hingga 12. Sehingga didapatkan total jumlah kombinasi model yang dihasilkan adalah 5824 model. Dapat kita lihat cuplikan datanya pada Tabel 6.3.

Tabel 6.3 MSE Node 4 Periode 1

Kode Model	Training MSE	Test MSE
...	...	...
1_1_2_5_6_4	147,8591	240,391
1_1_2_5_7_4	146,0951	242,4604
1_1_2_5_8_4	144,392	236,9321
1_1_2_5_9_4	146,6019	243,3944
1_1_2_6_1_4	146,08	238,5125
1_1_2_6_2_4	144,1408	233,7045
1_1_2_6_3_4	144,4474	255,8802
1_1_2_6_4_4	145,8336	239,9478
1_1_2_6_5_4	146,2022	241,866
1_1_2_6_6_4	147,2262	242,9697
1_1_2_6_7_4	143,0743	275,7529
1_1_2_6_8_4	147,8242	241,4641
1_1_2_6_9_4	146,0222	241,1504
1_1_2_7_1_4	153,2718	241,7591
...	...	...

Terlihat dari Tabel 6.3 bahwa nilai MSE terkecil dari keseluruhan model dengan node 4 adalah 233,7045. Kode model yang terlihat adalah 1_1_2_6_2_4. Angka-angka tersebut mewakili variabel dari parameter yang telah disetel, agar mudah mencari nama modelnya. Jika angka-angka tersebut di*breakdown* maka dapat dicari parameter apa saja yang digunakan. Agar dapat dijelaskan lebih mudah, angka-angka tersebut dapat diubah menjadi huruf terlebih dahulu. Contohnya seperti: **a_b_c_d_e_f**. Masing-masing huruf mewakili variabel masing-masing. Untuk lebih jelasnya dapat dilihat pada Tabel 6.4. Kolom kode model berisikan huruf variabel, dan dijelaskan maksud dari kode tersebut pada kolom arti dan nilainya.

Tabel 6.4 Penjelasan Kode Model

Kode Model	Artinya	Nilai
a	*train function*	1 = trainlm 2 = traingdx 3 = traingda
b	*learn function*	1 = learngdm
c	*transfer function*	1 = logsig 2 = tansig 3 = purelin
d	*momentum function*	1 = 0.1 … 9 = 0.9
e	*learnrate function*	1 = 0.1 … 9 = 0.9
f	Node	Sesuai angka node

Kode model yang didapat bisa dibaca sesuai dengan penjelasan di Tabel 6.4. Kode model yang didapatkan dari percobaan pertama adalah **1_1_2_6_2_4**. Ini berarti model yang ditemukan adalah menggunakan *train function trainlm, learn function learngdm, transfer function tansig, momentum function* ‘0.6’ dan *learnrate function* ‘0.2’. Angka 4 yang paling terakhir adalah node yang digunakan, yaitu 4.

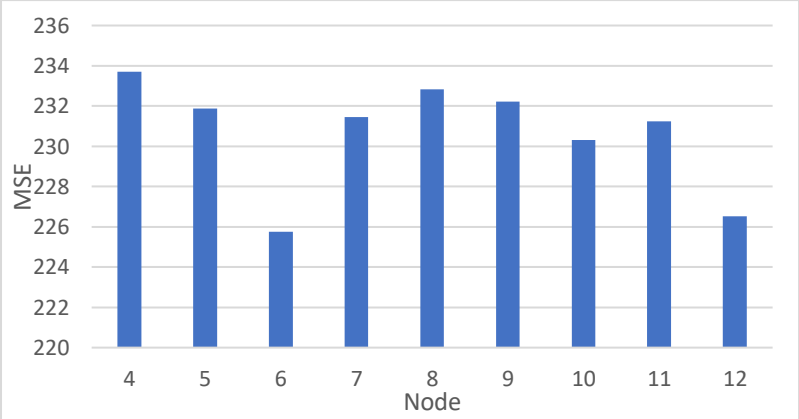
Setelah itu dicari kembali untuk periode yang sama namun node selanjutnya, yaitu 5, kemudian 6, seterusnya hingga node paling akhir untuk periode tersebut. Berikut hasil grafik dan tabel perbandingan MSE yang didapatkan dari keseluruhan model dalam 1 periode.

Tabel 6.5 Hasil Model dengan 1 Periode

Periode	Node	MSE	Kode Model
1	4	233,7045	1_1_2_6_2_4
	5	231,8743	1_1_2_7_7_5

Periode	Node	MSE	Kode Model
	6	225,7476	1_1_1_3_2_6
	7	231,4642	1_1_1_1_7_7
	8	232,8377	1_1_1_5_9_8
	9	232,2233	1_1_2_4_7_9
	10	230,3076	1_1_2_4_1_10
	11	231,2491	1_1_2_4_6_11
	12	226,52	1_1_2_6_1_12

Tabel 6.5 adalah hasil dari keseluruhan node yang dijalankan pada periode 1, dan hasil grafik dapat dilihat pada Gambar 6.1. **Periode 1** yang dimaksud adalah model menggunakan 1 periode sebelumnya (t-1). Dituliskan sebagai Periode 1 untuk memudahkan penulisan dan membedakan mana yang menggunakan 1 periode (t-1) dan mana yang menggunakan 2 periode ((t-1), (t-2)). Dimana penggunaan 2 periode dituliskan sebagai Periode 2.



Gambar 6.1 MSE Node dengan 1 Periode Sebelum

Dengan menggunakan grafik pada Gambar 6.1 dapat dengan mudah ditemukan bahwa nilai MSE yang paling kecil adalah nilai dari node 6. Maka nilai MSE terbaik dari periode 1 adalah

225,7476 pada kode model **1_1_1_3_2_6**. Nilai ini disimpan untuk dibandingkan dengan model-model terbaik pada periode yang lainnya.

6.3.2 Model dengan Jumlah Node Input 2 Periode Sebelumnya

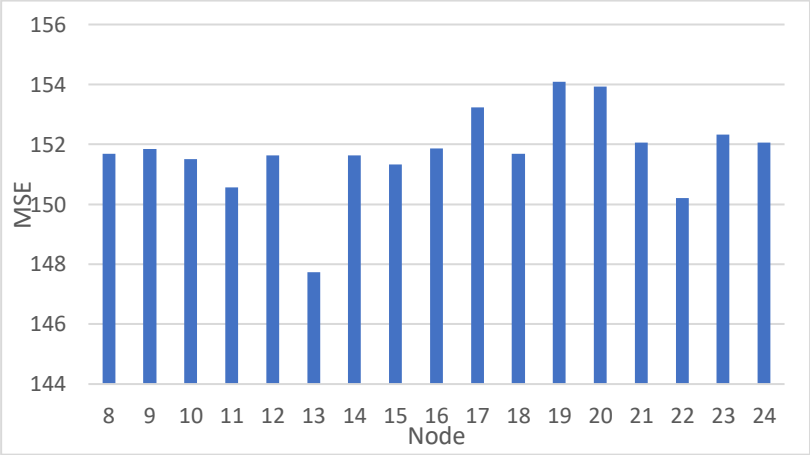
Sekarang percobaan diteruskan dengan menggunakan node dengan *input layer* 8. Dengan demikian node yang digunakan adalah 8 – 24. Sama seperti percobaan sebelumnya, yang dicari dari percobaan ini adalah nilai MSE terkecil. Ditemukan nilai MSE terkecil adalah 147,7297 dan kode modelnya adalah **1_1_2_7_7_13**. Untuk lebih jelasnya dapat dilihat pada Tabel 6.6 dan Gambar 6.2. **Periode 2** yang dimaksud adalah model menggunakan 2 periode sebelumnya ((t-1), (t-2)). Dalam periode ini dihasilkan 11.648 kombinasi model.

Tabel 6.6 Hasil Model Periode 2

Periode	Node	MSE	Kode Model
2	8	151,6769	1_1_2_8_7_8
	9	151,8387	1_1_1_4_5_9
	10	151,5003	1_1_1_8_3_10
	11	150,569	1_1_2_7_5_11
	12	151,6313	1_1_1_2_3_12
	13	147,7297	1_1_2_7_7_13
	14	151,6379	1_1_1_7_3_14
	15	151,3367	1_1_1_3_6_15
	16	151,8585	1_1_2_4_5_16
	17	153,2312	1_1_2_6_6_17
	18	151,6771	1_1_1_5_5_18
	19	154,0934	1_1_1_9_9_19

Periode	Node	MSE	Kode Model
	20	153,9365	1_1_2_6_1_20
	21	152,0632	1_1_2_6_9_21
	22	150,2142	1_1_1_1_5_22
	23	152,3326	1_1_1_6_1_23
	24	152,0616	1_1_2_6_2_24

Pada Gambar 6.2 dijelaskan bahwa MSE yang terkecil ada pada node 13.



Gambar 6.2 MSE pada Periode 2

6.3.3 Model dengan Jumlah Node Input 3 Periode Sebelumnya

Sekarang percobaan diteruskan dengan menggunakan node dengan *input layer* 12. Dengan demikian node yang digunakan adalah 12 – 36. Seperti percobaan sebelumnya, yang dicari dari percobaan ini adalah nilai MSE terkecil. Ditemukan nilai MSE terkecil adalah

187,0894 dan kode modelnya adalah **1_1_1_1_5_32**. Untuk lebih jelasnya dapat dilihat pada Tabel 6.7 dan Gambar 6.3. **Periode 3** yang dimaksudkan adalah model menggunakan 3 periode sebelumnya. Artinya adalah model menggunakan input dari 1 periode sebelumnya (t-1), ditambah 2 periode sebelumnya (t-2), ditambah lagi dengan 3 periode sebelumnya (t-3). Dalam periode ini dihasilkan 17.472 kombinasi model.

Tabel 6.7 Hasil Model dengan 3 Periode Sebelum

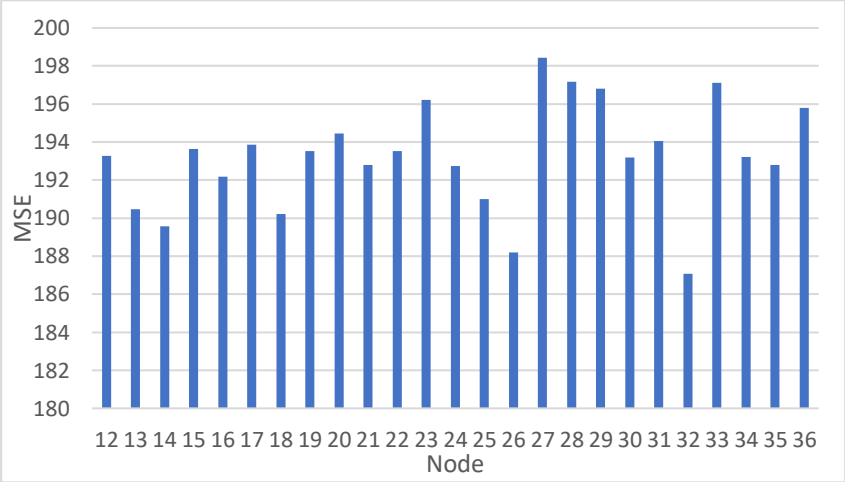
Periode	Node	MSE	Posisi
3	…	…	…
	29	196,7947	1_1_2_9_7_29
	30	193,19	1_1_1_1_3_30
	31	194,0554	1_1_1_1_8_31
	32	187,0894	1_1_1_1_5_32
	33	197,1249	1_1_1_1_6_33
	34	193,2096	1_1_2_2_6_34
	35	192,7923	1_1_2_2_6_35
	…	…	…

Pada Gambar 6.3 dijelaskan bahwa MSE yang terkecil ada pada node 32.

6.3.4 Model dengan Jumlah Node Input 4 Periode Sebelumnya

Lalu percobaan diteruskan dengan menggunakan node dengan *input layer* 16. Dengan demikian node yang digunakan adalah 16 – 48. Seperti percobaan sebelumnya, yang dicari dari percobaan ini adalah nilai MSE terkecil. Ditemukan nilai MSE terkecil adalah 117,7592, dan kode modelnya adalah **1_1_2_8_4_41**. Untuk lebih jelasnya dapat dilihat pada Tabel 6.8 dan Gambar 6.4. Atau untuk melihat hasil lebih lengkapnya pada periode 4 dapat dilihat pada Lampiran C. **Periode 4** yang dimaksud adalah model

menggunakan 4 periode sebelumnya. Artinya model menggunakan 1 periode sebelumnya (t-1), ditambah dengan 2 periode sebelumnya (t-2), ditambah lagi dengan 3 periode sebelumnya (t-3) dan juga ditambah 4 periode sebelumnya (t-4). Dalam periode ini dihasilkan 23.296 kombinasi model.



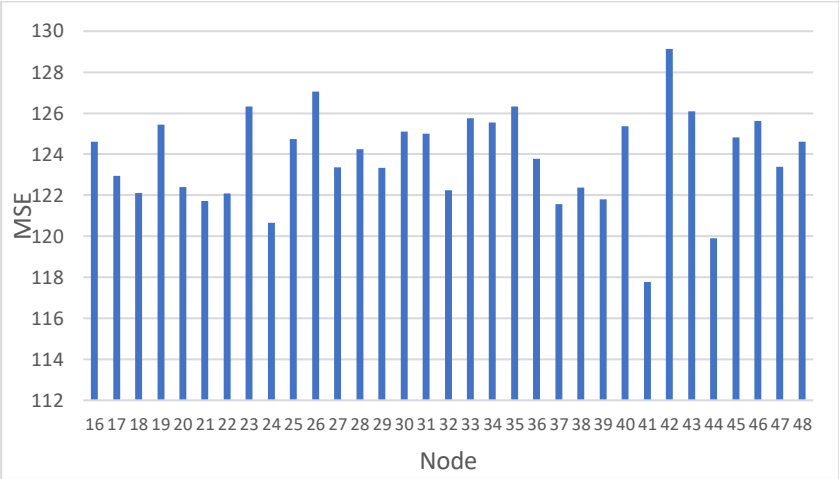
Gambar 6.3 MSE pada Periode 3

Tabel 6.8 Hasil Model Periode 4

Periode	Node	MSE	Posisi
4	…	…	…
	34	125,5572	1_1_2_5_5_34
	35	126,3246	1_1_1_7_8_35
	36	123,7683	1_1_1_9_2_36
	37	121,5722	1_1_2_8_5_37
	38	122,3599	1_1_1_6_8_38
	39	121,7933	1_1_1_8_3_39
	40	125,3724	1_1_1_1_1_40

Periode	Node	MSE	Posisi
	41	117,7592	1_1_2_8_4_41
	42	129,1292	1_1_2_1_7_42
	43	126,0926	1_1_1_2_6_43
	44	119,9004	1_1_1_6_2_44
	45	124,8158	1_1_2_3_7_45
	46	125,6338	1_1_1_4_3_46
	47	123,3905	1_1_1_8_2_47
	48	124,607	1_1_2_7_5_48

Pada Gambar 6.4 dijelaskan bahwa MSE yang terkecil ada pada node 41.



Gambar 6.4 MSE pada Periode 4

6.3.5 Model dengan Jumlah Node Input 5 Periode Sebelumnya

Kemudian percobaan diteruskan dengan menggunakan node dengan *input layer* 20. Dengan demikian node yang digunakan adalah 20 – 60. Seperti percobaan sebelumnya, yang dicari dari

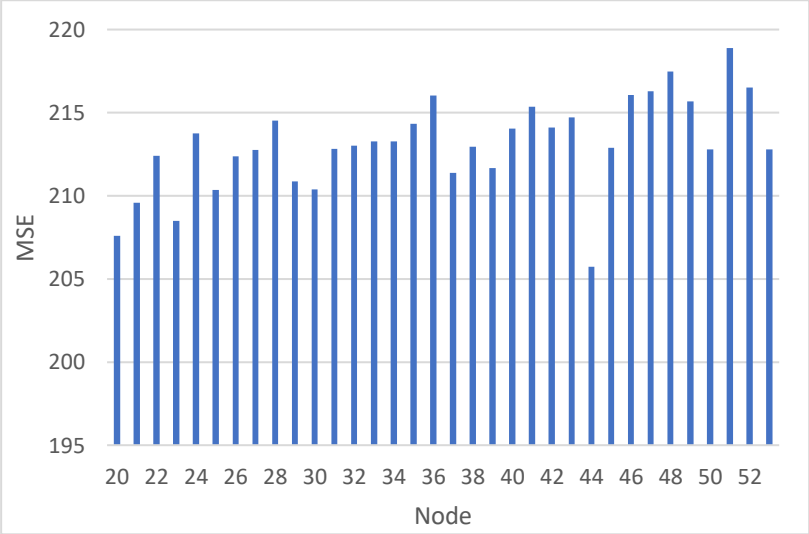
percobaan ini adalah nilai MSE terkecil. Ditemukan nilai MSE terkecil adalah 205,7370, dan kode modelnya adalah **1_1_2_2_5_44**. Untuk lebih jelasnya dapat dilihat pada Tabel 6.9 dan Gambar 6.5. Periode 5 yang dimaksud adalah model menggunakan kelima periode sebelumnya ((t-1), (t-2), (t-3), (t-4), (t-5)).

Tabel 6.9 Hasil Model Periode 5

Periode	Node	MSE	Model
5	…	…	…
	37	211,3714	1_1_1_9_8_37
	38	212,9581	1_1_1_1_5_38
	39	211,678	1_1_2_6_4_39
	40	214,0555	1_1_1_3_6_40
	41	215,3655	1_1_1_9_1_41
	42	214,0971	1_1_2_3_2_42
	43	214,7118	1_1_1_4_5_43
	44	205,7371	1_1_2_2_5_44
	45	212,8938	1_1_2_7_5_45
	46	216,0696	1_1_2_9_7_46
	47	216,2729	1_1_2_7_1_47
	48	217,4808	1_1_2_9_3_48
	49	215,6859	1_1_1_4_6_49
	50	212,7993	1_1_2_6_7_50
	51	218,8817	1_1_2_3_5_51
	52	216,5112	1_1_1_5_4_52
	…	…	…

Pada Gambar 6.5 terlihat bahwa MSE yang terkecil ada pada Node 44. Nilai pada Node 44 relatif lebih kecil dibandingkan dengan node-node yang lainnya. Nilai dari node 44, sebagaimana yang

terlihat pada tabel 6.9, menunjukkan nilai yang signifikan dibandingkan dengan node-node yang lainnya. Namun, jika dibandingkan dengan model yang menggunakan 4 periode sebelumnya, hasil ini tidak terlalu memuaskan. Terlebih lagi nilai MSE ini merupakan hasil kedua terbesar jika dibandingkan dengan model-model terbaik dari model lainnya. Walaupun begitu, nilai ini tetap dapat digunakan sebagai pembanding oleh node-node dengan periode yang sama (5 periode sebelumnya). Juga dapat dilihat parameter model ini adalah *tansig* dan *traingdx*. Kemungkinan besar parameter tersebut dapat menghasilkan model dengan nilai mse yang cukup baik. Namun demikian faktor-faktor lain seperti kombinasi *learning rate*, *momentum function* serta jumlah node juga berpengaruh pada hasil nilai mse sehingga cara terbaik tetap dilakukan *trial and error*.



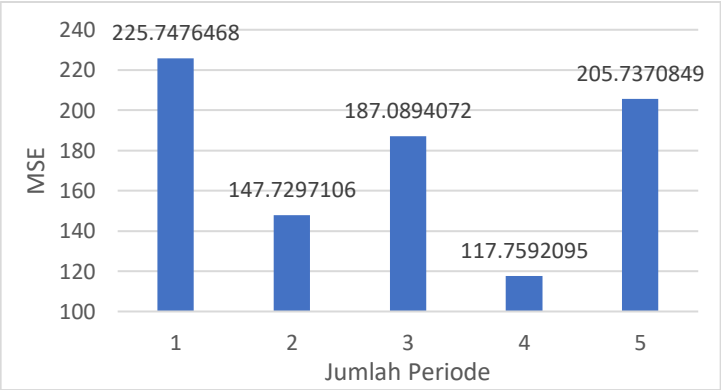
Gambar 6.5 MSE pada Periode 5

6.4 Kesimpulan Hasil Percobaan

Dari seluruh hasil percobaan yang dilakukan pada semua model beserta perubahan parameternya, analisis dilakukan lebih lanjut pada model-model dengan nilai yang paling optimal. Hasil parameter terbaik dari masing-masing model dan masing-masing periode akan ditunjukkan pada tabel 6.10 dan Gambar 6.6

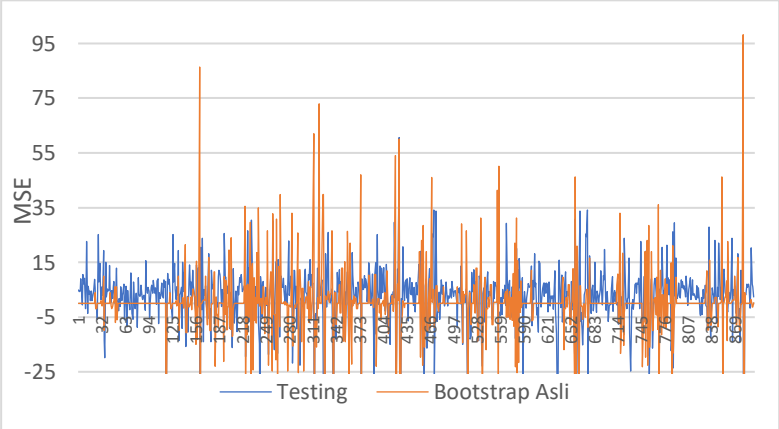
Tabel 6.10 MSE terbaik pada tiap Jumlah Periode

Periode	Node	MSE	Kode
1	6	225,7476	1_1_1_3_2_6
2	13	147,7297	1_1_2_7_7_13
3	32	187,0894	1_1_1_1_5_32
4	41	117,7592	1_1_2_8_4_41
5	44	205,7371	1_1_2_2_5_44



Gambar 6.6 MSE terkecil tiap Jumlah Periode

Untuk memilih model yang paling optimal, maka digunakan model **1_1_2_8_4_41**. Kode dari model ini di*breakdown* sehingga hasilnya model yang paling optimal adalah model periode 4 node 41 dengan parameter fungsi pelatihan *trainlm*, fungsi pembelajaran *learngdm*, fungsi transfer *tansig*, fungsi momentum 0.8, fungsi *learnrate* 0.4. Dapat dilihat perbandingan data *target* dan data hasil *test* pada Gambar 6.7. Data dapat dilihat pada Lampiran D.

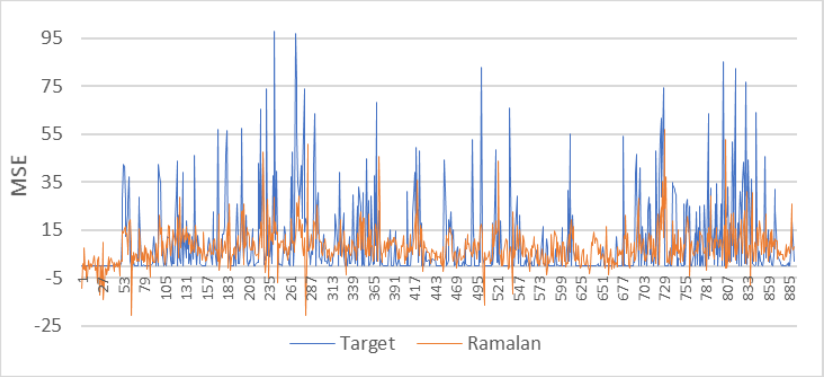


Gambar 6.7 Model Optimal dengan Target

6.5 Hasil Peramalan

Setelah model yang paling optimal ditemukan, model tersebut diimplementasikan untuk melakukan peramalan. Caranya adalah model tersebut dimasukkan pada sistem dan dijalankan dataset untuk mendapatkan hasil Test MSE yang akan menjadi hasil peramalan. Namun, dataset yang digunakan bukanlah merupakan data hasil *bootstrap* melainkan data curah hujan beserta data pendukungnya yang asli, dibuat sesuai dengan periode yang tepat. Dapat dilihat hasilnya seperti pada Gambar 6.8. Untuk nilai dari

SMAPE didapatkan dari *tool* NumXL, dengan melakukan perbandingan nilai *testing* dengan *error testing* [20]. Dilakukan dua kali untuk data hasil *bootstrap* dan juga dengan data asli. Kemudian kedua hasilnya dapat dibandingkan. SMAPE digunakan karena data ada yang bernilai 0 ataupun mendekati 0 [21].



Gambar 6.8 Model Optimal dengan Data Asli

6.6 Analisis Perbandingan Hasil Peramalan Data dengan Bootstrap dan Data Asli

Model dibandingkan ketika menggunakan dataset menggunakan Bootstrap dan dataset yang asli. Hasil menggunakan pembandingan nilai MSE dan SMAPE, dengan nilai masing-masing tertera pada Tabel 6.11.

Tabel 6.11 Perbandingan pada Model dengan Data yang berbeda

Dataset	MSE	SMAPE
Test - Bootstrap	163.2885	67.45%
Test – Data Asli	256.5257	102.55%

BAB VII
KESIMPULAN DAN SARAN

Dalam bab ini dijelaskan tentang kesimpulan dari penelitian yang dilakukan dan juga saran yang dapat diberikan agar dapat melanjutkan pengembangan penelitian dengan lebih baik.

7.1 Kesimpulan

Berdasarkan hasil dari uji coba yang dilakukan, kesimpulan yang dapat diambil adalah:

1. Metode *Artificial Neural Network* (ANN) dapat digunakan untuk peramalan curah hujan, sehingga dapat membantu pengambilan keputusan pada periode-periode tertentu yang sekiranya membutuhkan perlakuan sebelum musim hujan datang atau sebelum musim kering.
2. Model yang terbaik untuk digunakan adalah model periode 4 node 41 dengan parameter fungsi pelatihan *trainlm*, fungsi pembelajaran *learngdm*, fungsi transfer *tansig*, fungsi momentum 0.8, fungsi *learnrate* 0.4. Model ini menghasilkan MSE 163.2885 dan hasil SMAPE 67.45%

7.2 Saran

Berdasarkan kesimpulan dari penelitian tugas akhir, beberapa saran yang diberikan bagi penelitian selanjutnya adalah:

1. Uji coba pada penelitian tugas akhir ini terbatas pada data periode Januari 2012 – Januari 2017. Dengan menambah periode data, data yang didapat dan diproses akan lebih bervariasi dan periode lebih panjang.
2. Dapat ditambahkan metode-metode lain dalam pengerjaan, seperti ARIMA, ARIMAX, dan metode-metode lainnya.

3. Ke depannya dapat ditambahkan *input* periode lebih dari 5, menggunakan mesin komputasi yang lebih *high-end* sehingga tidak memakan waktu yang lama.

DAFTAR PUSTAKA

[1] Suroso, “Analisis Intensitas Durasi Frekuensi Kejadian Hujan di Kabupaten Cilacap, Jawa Tengah,” *EMAS*, vol. 16, no. No. 2, 2006.

[2] B. HARYONO and SUHARDI, *Prosedur Analisa untuk Bahan Makanan dan Pertanian*, 4th ed. Yogyakarta: Liberty, 1997.

[3] L. Chen and X. Lai, “Comparison between ARIMA and ANN models used in short-term wind speed forecasting,” in *Asia-Pacific Power and Energy Engineering Conference, APPEEC*, 2011.

[4] D. Solomatine, L. M. See, and R. J. Abrahart, “Data-Driven Modelling: Concepts, Approaches and Experiences,” *Pract. Hydroinformatics Comput. Intell. Technol. Dev. Water Appl.*, vol. 68, pp. 17–30, 2008.

[5] J. W. Taylor and R. Buizza, “Neural Network Load Forecasting with Weather Ensemble Predictions,” *IEEE Power Engineering Review*, vol. 22, no. 7. p. 59, 2002.

[6] Y. M. Chiang, F. J. Chang, B. J. D. Jou, and P. F. Lin, “Dynamic ANN for precipitation estimation and forecasting from radar observations,” *J. Hydrol.*, vol. 334, no. 1–2, pp. 250–261, 2007.

[7] K. Abhishek, M. P. P. Singh, S. Ghosh, and A. Anand, “Weather Forecasting Model using Artificial Neural Network,” *Procedia Technol.*, vol. 4, no. 0, pp. 311–318, 2012.

[8] R. M. Gor, “Forecasting Techniques,” *Time Ser. Methods*, no. Bodakdev : ICFAI Business School.

[9] M. A. Moon, J. T. Mentzer, C. D. Smith, and M. S. Garver, “Seven keys to better forecasting,” *Bus. Horiz.*,

vol. 41, no. 5, pp. 44–52, 1998.

[10] D. Waller, “Methods for Intermittent Demand Forecasting,” pp. 1–6, 2016.

[11] T. R. Willemain, C. N. Smart, and H. F. Schwarz, “A new approach to forecasting intermittent demand for service parts inventories,” *Int. J. Forecast.*, vol. 20, no. 3, pp. 375–387, 2004.

[12] R. H. Teunter, A. A. Syntetos, and M. Z. Babai, “Intermittent demand: Linking forecasting to inventory obsolescence,” *European Journal of Operational Research*, vol. 214, no. 3. pp. 606–615, 2011.

[13] Siana Halim and Adrian Michael Wibisono, “Penerapan Jaringan Saraf Tiruan Untuk Peramalan,” *J. Tek. Ind.*, 2000.

[14] S. Hui and S. H. Zak, “The Widrow-Hoff Algorithm for McCulloch-Pitts Type Neurons,” *IEEE Trans. Neural Networks*, vol. 5, no. 6, pp. 924–929, 1994.

[15] T. M. Mitchell, *Machine Learning*, no. 1. 1997.

[16] P. Azimi, H. R. Mohammadi, E. C. Benzel, S. Shahzadi, S. Azhari, and A. Montazeri, “Artificial neural networks in neurosurgery,” *Journal of Neurology, Neurosurgery and Psychiatry*, vol. 86, no. 3. pp. 251–256, 2015.

[17] A. Triyono, A. J. Santoso, and P. Pranowo, “Penerapan Metode Jaringan Syaraf Tiruan Backpropagation Untuk Meramalkan Harga Saham (IHSG),” *J. Sist. dan Inform.*, 2016.

[18] D. N. Politis and J. P. Romano, “The stationary bootstrap,” *J. Am. Stat. Assoc.*, 1994.

[19] J. Heaton, “Introduction to the Maths of Neural

Networks," *Neural Networks*, 2012.

[20] J. D. Hamilton, "Time Series Analysis," *Book*. 1994.

[21] R. S. Tsay, *Analysis of Financial Time Series*. 2005.

Halaman ini sengaja dikosongkan

BIODATA PENULIS

Penulis lahir di Surabaya, 13 Februari 1996, dengan nama lengkap Aditya Parama Hadi. Penulis adalah anak kedua dari dua bersaudara.

Riwayat pendidikan dari penulis adalah SD Luqman Al-Hakim Surabaya, SMP Negeri 6 Surabaya, SMA Negeri 5 Surabaya, dan salah satu mahasiswa Sistem Informasi ITS angkatan 2014 dengan jalur SBMPTN dengan NRP 5214100123.

Selama kuliah penulis bergabung dalam himpunan kemahasiswaan Sistem Informasi, dan ikut bergabung dalam biro komunitas. Selain itu, penulis juga salah satu anggota komunitas musik ITS Jazz. Penulis aktif bermain dalam komunitas ITS Jazz sebagai salah satu drummer selama 3 tahun.

Penulis mengambil laboratorium bidang minat Rekayasa Data dan Intelegensia Bisnis (RDIB) di Departemen Sistem Informasi ITS. Penulis dapat dihubungi lewat *email* di: adityaparamahadi@gmail.com

Halaman ini sengaja dikosongkan

LAMPIRAN A

Data iklim harian 1 Januari 2012 – 1 Januari 2017. Provinsi Jawa Timur Kabupaten Malang. Data iklim yang digunakan adalah data Suhu Minimum, Kelembaban Rata-Rata, Curah Hujan dan Kecepatan Angin Rata-Rata

Tanggal	Suhu Minimum	Kelembaban Rata-Rata	Curah Hujan	Kecepatan Angin Rata-Rata
01/01/2012	22.8	91	22	1
01/01/2012	21.2	89	4	1
02/01/2012	20.8	84	38	3
02/01/2012	23	83	4	1
03/01/2012	20.9	90	22.5	1
03/01/2012	22.8	92	39	1
04/01/2012	22	91	4	1
04/01/2012	21.3	92	5	1
05/01/2012	20.2	85	4	1
05/01/2012	22.8	89	14	1
06/01/2012	21.2	90	13	1
06/01/2012	23	86	8	1
07/01/2012	22.8	88	16	1
07/01/2012	21.2	86	22	2
08/01/2012	21.2	91	30	1
08/01/2012	23	93	7	1
09/01/2012	21.2	87	1	2
09/01/2012	22.6	87	3	1
10/01/2012	23.2	87	1	1
10/01/2012	21	86	29	2

Tanggal	Suhu Minimum	Kelembaban Rata-Rata	Curah Hujan	Kecepatan Angin Rata-Rata
11/01/2012	21	85	1	2
11/01/2012	22.8	82	4	1
12/01/2012	22	91	6.5	1
12/01/2012	23	90	9	1
13/01/2012	23	85	62	1
13/01/2012	21	85	2	2
14/01/2012	20.6	89	3	1
14/01/2012	22.4	86	13	1
15/01/2012	20.2	89	43	1
15/01/2012	23	89	1	1
16/01/2012	22.2	87	8	1
16/01/2012	20.6	92	4	1
17/01/2012	20.7	82	7	4
17/01/2012	22.2	90	0	1
18/01/2012	21	86	0	1
18/01/2012	22.5	88	21	1
19/01/2012	21.2	91	2	1
19/01/2012	22.4	88	30	1
20/01/2012	23	91	10	1
20/01/2012	20.7	88	17	2
21/01/2012	20.8	88	3	2
21/01/2012	22	94	27	1
22/01/2012	20.6	75	17	2
22/01/2012	22.6	85	24	1
23/01/2012	22.1	84	27	1
23/01/2012	21.5	74	3	2

Tanggal	Suhu Minimum	Kelembaban Rata-Rata	Curah Hujan	Kecepatan Angin Rata-Rata
24/01/2012	22.2	62	0	6
24/01/2012	22.8	80	0	3
25/01/2012	22.2	60	3	5
25/01/2012	25.2	65	0	5
26/01/2012	25	69	0	4
26/01/2012	21.8	71	0	3
27/01/2012	24.4	58	0	4
27/01/2012	23.2	73	0	3
28/01/2012	21.7	73	3	4
28/01/2012	23	82	9	2
29/01/2012	21.6	82	33	2
29/01/2012	21.6	72	3	3
30/01/2012	20.3	86	26	2
30/01/2012	22.4	92	4	1
31/01/2012	21.4	84	3	2
31/01/2012	22.2	93	17	1
01/02/2012	22.4	83	0	1
01/02/2012	20.4	84	50	1
02/02/2012	22.6	83	0	1
02/02/2012	20.8	86	41	2
03/02/2012	22.2	86	46	1
03/02/2012	20.4	85	41	2
04/02/2012	21.6	91	24	1
04/02/2012	20.4	86	44	2
05/02/2012	23.4	79	0	1
05/02/2012	19.8	80	14	2

Tanggal	Suhu Minimum	Kelembaban Rata-Rata	Curah Hujan	Kecepatan Angin Rata-Rata
06/02/2012	19.9	82	0	2
06/02/2012	22.1	85	1	1
07/02/2012	22.2	84	1	1
07/02/2012	19.6	76	0	4
08/02/2012	22	80	21	1
…	…	…	…	…
01/12/2016	20.6	87	43.1	1
01/12/2016	22.3	91	29	1
02/12/2016	20.9	92	0.9	1
02/12/2016	22	94	76.5	1
03/12/2016	22.8	90	0.5	1
03/12/2016	20	91	0.5	1
04/12/2016	21.2	87	44.2	1
04/12/2016	23	90	26.3	1
05/12/2016	23	86	1	1
05/12/2016	21.8	91	5.1	1
06/12/2016	20.9	87	36.3	1
06/12/2016	23.6	87	1.5	1
07/12/2016	22.8	81	1	1
07/12/2016	21.4	85	0.6	4
08/12/2016	22.1	90	0.2	1
08/12/2016	23.2	86	0.6	1
09/12/2016	22	89	64	1
09/12/2016	21.8	89	11.1	1
10/12/2016	23.2	86	3.9	1.5
10/12/2016	20.8	87	0	2

Tanggal	Suhu Minimum	Kelembaban Rata-Rata	Curah Hujan	Kecepatan Angin Rata-Rata
11/12/2016	21.2	88	6.3	2
11/12/2016	22.8	87	2	1.5
12/12/2016	23.4	87	1.3	1
12/12/2016	21.1	88	0.6	1
13/12/2016	21.2	91	0	1
13/12/2016	22.8	88	2.7	1
14/12/2016	23.2	88	5.3	1
14/12/2016	20.6	90	45.8	1
15/12/2016	20.8	88	3.2	1
15/12/2016	22.2	88	6.3	1
16/12/2016	23.4	87	9.5	1
16/12/2016	20.5	84	15.2	2
17/12/2016	23.8	80	1.3	2
17/12/2016	21.4	82	6.9	2
18/12/2016	24.8	71	4	3
18/12/2016	23.4	65	1	4
19/12/2016	23.8	80	0.5	2
19/12/2016	23.4	74	0	3
20/12/2016	23.8	76	4.8	2
20/12/2016	21.5	77	32	2
21/12/2016	21.9	77	22.73	2
21/12/2016	22.8	76	13.47	2
22/12/2016	22.8	75	4.2	1
22/12/2016	20.2	78	4.2	1
23/12/2016	22.8	80	3.36	2
23/12/2016	22	75	2.52	2

Tanggal	Suhu Minimum	Kelembaban Rata-Rata	Curah Hujan	Kecepatan Angin Rata-Rata
24/12/2016	22.8	78	1.68	2
24/12/2016	20.4	77	0.84	2
25/12/2016	21.2	82	0	2
25/12/2016	23.4	79	0.3	1
26/12/2016	20.6	78	0	2
26/12/2016	23.8	77	0	2
27/12/2016	23.4	82	0	1
27/12/2016	22.7	83	0	3
28/12/2016	21.6	84	0	3
28/12/2016	22.8	83	1	2.5
29/12/2016	21.8	85	0	2
29/12/2016	23.8	84	1.5	2
30/12/2016	22.2	94	0.2	1
30/12/2016	22.2	93	6.1	1
31/12/2016	20.9	85	12	2
31/12/2016	23.6	85	18.4	1
01/01/2017	22.2	87	10.2	2
01/01/2017	24.4	83	2	1

LAMPIRAN B

Lampiran B berisi hasil data asli setelah dilakukan *bootstrap*

Tanggal	Suhu Minimum	Kelembaban Rata-Rata	Curah Hujan	Kecepatan Angin Rata-Rata
01/01/2012	-0.8	2	28.5	0
01/01/2012	1.6	-4	-29.6	0
02/01/2012	-2	0	-2.4	0
02/01/2012	2.1	0	-7.2	0.33
03/01/2012	0	-1	0.6	1
03/01/2012	-1.8	-2	0.6	1
04/01/2012	0.6	-5	0	-1
04/01/2012	1.3	12	0	0
05/01/2012	-0.2	0	0	0
05/01/2012	-0.7	-8	0	0
06/01/2012	-0.1	-5	16	1
06/01/2012	1	9	-11.5	-0.5
07/01/2012	-1.9	-5	-4.5	-0.5
07/01/2012	2.7	9	0	0
08/01/2012	-2.2	-8	0	0
08/01/2012	-0.2	16	0	-1
09/01/2012	6.2	2	0	0
09/01/2012	-1.5	0	0	0
10/01/2012	-0.3	-3	11	0
10/01/2012	-1.2	-1	-11	-1
11/01/2012	-1.7	-3	0	1
11/01/2012	-0.7	-3	0	0
12/01/2012	2.2	1	0	-1
12/01/2012	-1.4	-3	0	0.5

Tanggal	Suhu Minimum	Kelembaban Rata-Rata	Curah Hujan	Kecepatan Angin Rata-Rata
13/01/2012	-0.5	-11	0	0.5
…	…	…	…	…
23/12/2016	-3.2	-7	0	0
23/12/2016	-0.7	2	0	1
24/12/2016	2.8	9	0	-1
24/12/2016	-1	-1	0	-1
25/12/2016	-1.8	-6	0	2
25/12/2016	-1.6	-1	3.25	-1
26/12/2016	-1.6	-1	4.25	1
26/12/2016	4.9	0	12	0
27/12/2016	-0.8	-7	0	1
27/12/2016	2	4	9.5	-0.5
28/12/2016	-1.5	-8	9.8	-0.5
28/12/2016	1.1	13	14.9	0
29/12/2016	-2	-11	-44	0
29/12/2016	2	6	21.1	0.5
30/12/2016	-1.8	-4	-18.1	0.5
30/12/2016	2.6	8	-2	-0.5
31/12/2016	-1.5	-1	-0.2	-0.5
31/12/2016	-0.5	-4	9.6	0
01/01/2017	1.5	2	28.6	0
01/01/2017	-1.4	-5	-39	0

LAMPIRAN C

Lampiran C berisi data hasil *trial and error* parameter dari periode ke 4 untuk menemukan model ANN dengan parameter terbaik

Kode Model	Training MSE	Test MSE
1_1_1_1_1_16	137.204	145.8675
1_1_1_1_2_16	126.5204	131.7145
1_1_1_1_3_16	124.2505	140.8673
1_1_1_1_4_16	132.0094	129.6373
1_1_1_1_5_16	135.5787	134.0147
1_1_1_1_6_16	121.8796	131.346
1_1_1_1_7_16	114.2501	131.363
1_1_1_1_8_16	123.1174	132.9254
1_1_1_1_9_16	123.5357	128.9996
1_1_1_2_1_16	120.1457	140.3683
1_1_1_2_2_16	120.6548	134.83
1_1_1_2_3_16	121.5093	137.4144
1_1_1_2_4_16	112.0162	133.7755
1_1_1_2_5_16	122.1846	129.2946
1_1_1_2_6_16	117.0462	138.6686
1_1_1_2_7_16	118.3065	124.6089
1_1_1_2_8_16	116.9494	138.2515
1_1_1_2_9_16	117.4233	144.6571
1_1_1_3_1_16	121.0677	137.1255
1_1_1_3_2_16	134.9179	131.0809
1_1_1_3_3_16	114.9996	131.381
1_1_1_3_4_16	114.0066	142.1937
1_1_1_3_5_16	131.7398	130.5866
1_1_1_3_6_16	119.6929	130.5519

Kode Model	Training MSE	Test MSE
1_1_1_3_7_16	120.3041	126.047
1_1_1_3_8_16	125.2958	136.3551
1_1_1_3_9_16	134.8752	129.9614
1_1_1_4_1_16	124.6304	126.25
1_1_1_4_2_16	111.783	136.9859
…	…	…
1_1_2_7_9_41	144.9512	145.1548
1_1_2_8_1_41	159.6485	209.4747
1_1_2_8_2_41	117.8365	146.8105
1_1_2_8_3_41	120.3016	132.0434
1_1_2_8_4_41	111.2116	117.7592
1_1_2_8_5_41	135.3643	147.8942
1_1_2_8_6_41	107.1851	146.3073
1_1_2_8_7_41	98.49236	161.9246
1_1_2_8_8_41	113.4183	133.5885
…	…	…
3_1_3_9_2_48	145.7901	136.2771
3_1_3_9_3_48	145.0537	136.076
3_1_3_9_4_48	145.9737	136.607
3_1_3_9_5_48	145.5077	136.3704
3_1_3_9_6_48	145.807	137.0921
3_1_3_9_7_48	146.1725	136.405
3_1_3_9_8_48	145.9793	137.3142
3_1_3_9_9_48	145.2855	135.7356

LAMPIRAN D

Lampiran D berisikan data *training*, *error training*, *testing* dan *error testing* dari model yang terbaik

Training	Error Training	Testing	Error Testing
8.52967	8.52967	4.935349	4.935349
8.188258	8.188258	4.391257	4.391257
3.832378	3.832378	4.447714	4.447714
4.238654	-10.9613	8.958916	8.958916
-6.05311	8.046891	5.799561	5.799561
4.503077	5.403077	-0.64179	-0.64179
7.987677	8.187677	10.63081	10.63081
11.45064	-14.9494	7.066081	7.066081
-8.93926	-15.0393	9.190078	9.190078
-20.3546	9.2454	-1.93279	-1.93279
3.029383	-19.0706	5.979527	5.979527
-3.82847	-1.12847	22.68511	22.68511
-0.16978	22.13022	-3.69009	-3.69009
18.62755	17.52755	6.240589	6.240589
10.27062	11.27062	3.32136	3.32136
3.347317	0.547317	4.409684	4.409684
6.547286	-5.85271	0.949373	0.949373
-7.62663	-0.02663	4.417272	4.417272
3.90902	-4.79098	1.049907	1.049907
-3.33596	13.06404	3.075629	3.075629
3.576235	1.976235	4.069851	4.069851
-28.6675	20.63255	8.726465	8.726465
-0.39835	-0.39835	5.596189	1.596189
2.1701	2.1701	-1.92178	0.078216

Training	Error Training	Testing	Error Testing
6.8094	-8.1906	-1.27941	0.720589
12.00334	-2.99666	-5.82826	-5.82826
…	…	…	…
2.323083	-0.37692	18.18777	1.387772
11.06151	5.361513	-5.57274	-8.37274
6.342227	-2.55777	-1.87807	8.071929
-2.91342	17.68658	-14.0478	-4.09777
3.837226	3.837226	12.06725	12.21725
9.608823	8.608823	5.261992	5.411992
0.206826	-0.29317	0.247542	0.247542
11.71495	11.71495	6.09658	-92.1034
2.49093	3.49093	-80.6409	7.359106
0.498831	0.498831	1.946643	-1.90336
8.295682	8.295682	3.251318	3.251318
30.35271	18.65271	3.098601	3.098601
10.44869	-4.25131	7.03925	7.03925
-13.652	10.74803	5.885667	5.885667
6.4419	-2.8581	7.043585	7.043585
-2.97188	3.328121	6.110059	6.110059
6.161268	3.961268	-1.76171	-1.76171
0.333887	7.533887	3.721553	2.371553
10.58286	-8.01714	20.32448	18.97448
-13.4148	1.885165	7.093305	8.443305
2.323083	-0.37692	5.986949	7.336949
11.06151	5.361513	2.434448	2.134448