

PREDIKSI CURAH HUJAN MENGGUNAKAN *RECURRENT NEURAL NETWORK - LONG SHORT TERM MEMORY* : STUDI KASUS DI STASIUN BMKG KARANGPLOSO, MALANG

SKRIPSI

**Oleh :
MUHAMMAD FEBRIANTORO
155090300111007**



**JURUSAN FISIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
2018**

PREDIKSI CURAH HUJAN MENGGUNAKAN *RECURRENT NEURAL NETWORK - LONG SHORT TERM MEMORY* : STUDI KASUS DI STASIUN BMKG KARANGPLOSO, MALANG

SKRIPSI

Sebagai Salah Satu Syarat Untuk Meraih Gelar
Sarjana Sains dalam Bidang Fisika

Oleh :
MUHAMMAD FEBRIANTORO
155090300111007



JURUSAN FISIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
2018

LEMBAR PENGESAHAN SKRIPSI

PREDIKSI CURAH HUJAN MENGGUNAKAN *RECURRENT NEURAL NETWORK - LONG SHORT TERM MEMORY* : STUDI KASUS DI STASIUN BMKG KARANGPLOSO, MALANG

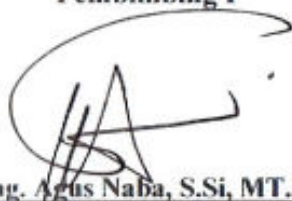
Oleh :

MUHAMMAD FEBRIANTORO

155090300111007

Setelah dipertahankan di depan Majelis Penguji
pada tanggal **03 JAN 2019**
dan dinyatakan memenuhi syarat untuk memperoleh gelar
Sarjana Sains dalam bidang Fisika

Pembimbing I



Dr.Eng. Agus Naba, S.Si, MT., Ph.D

NIP. 197208061995121001

Pembimbing II



Dr.rer.nat. Abdurro'uf, S.Si., M.Si.

NIP. 197209031994121001

Mengetahui,

Ketua Jurusan Fisika

FMIPA Universitas Brawijaya



Prof. Dr.rer.nat. Muhammad Nurhuda

NIP. 19640910199021001

LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : Muhammad Febriantoro
NIM : 155090300111007
Jurusan : Fisika
Penulis Skripsi Berjudul :

PREDIKSI CURAH HUJAN MENGGUNAKAN *RECURRENT NEURAL NETWORK - LONG SHORT TERM MEMORY* : STUDI KASUS DI STASIUN BMKG KARANGPLOSO, MALANG

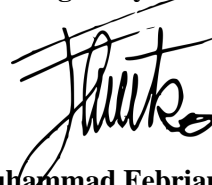
Dengan ini menyatakan bahwa :

1. Isi dari skripsi yang saya tulis dan saya buat adalah benar-benar karya saya sendiri dan tidak menjiplak karya orang lain, selain nama-nama yang termaktub di isi dan tertulis di daftar pustaka dalam skripsi ini.
2. Apabila di kemudian hari ternyata skripsi yang saya tulis terbukti hasil menjiplak, maka saya akan bersedia menanggung segala resiko yang akan saya terima.

Demikian pernyataan ini dibuat dengan segala kesadaran.

Malang, 31 Desember 2018

Yang menyatakan,



Muhammad Febriantoro
155090300111007

PREDIKSI CURAH HUJAN MENGGUNAKAN *RECURRENT NEURAL NETWORK - LONG SHORT TERM MEMORY* : STUDI KASUS DI STASIUN BMKG KARANGPLOSO, MALANG

ABSTRAK

Informasi prakiraan hujan dapat membantu pengaturan sumber daya air di suatu tempat yang khususnya ketika berhubungan dengan munculnya perubahan iklim di daerah tropis seperti di Indonesia. Untuk mengatasi masalah tersebut, prediksi curah hujan dengan metode yang baik serta akurat merupakan hal yang sangat diperlukan untuk mengantisipasinya. Sepanjang pengembangan teknologi komputasi, *Artificial Neural Network* (ANN) telah digunakan untuk membuat prediksi curah hujan. Pada penelitian ini, penulis menggunakan salah satu model ANN untuk memprediksi curah hujan mendatang yang bernama *Recurrent Neural Network – Long Short-Term Memory* (RNN-LSTM). RNN-LSTM merupakan sistem jaringan syaraf buatan yang didesain khusus untuk menangani data deret waktu dalam jangka waktu yang panjang seperti curah hujan. Pada arsitekturnya, Model RNN-LSTM menggunakan 2 *Hidden Layer* LSTM yang terdiri dari 108 Neuron LSTM di masing-masing *layer*-nya. *Activation function* yang digunakan adalah *Tanh*. *Loss function* yang digunakan adalah *Mean Square Error*. Algoritma optimasi yang digunakan adalah *Adam Optimizer*. Hasil yang didapatkan adalah model dapat semakin baik memprediksi curah hujan mendatang jika data *input* yang diberikan ke model semakin panjang yang ditandai dengan nilai *Root Mean Square Error* yang semakin mengecil. Namun model kurang dapat memprediksi curah hujan mendatang jika *output* tahun prediksi semakin panjang yang ditandai dengan nilai *Root Mean Square Error* yang semakin membesar.

Kata kunci : Prediksi, Curah Hujan, *Recurrent Neural Network*, *Long Short Term Memory*

RAINFALL PREDICTION USING RECURRENT NEURAL NETWORK - LONG SHORT TERM MEMORY: CASE STUDY IN STASIUN BMKG KARANGPLOSO, MALANG

ABSTRACT

Rainfall forecast information can help regulate water resources in a place that is especially related to the emergence of climate change in tropical regions such as Indonesia. To overcome this problem, the prediction of rainfall using good and accurate method is very necessary thing to anticipate it. Throughout the development of computing technology, Artificial Neural Network (ANN) has been used to make rainfall predictions. In this research, the authors used one of the ANN models to predict future rainfall called Recurrent Neural Network - Long Short-Term Memory (RNN-LSTM). RNN-LSTM is an artificial neural network system designed specifically to handle time series data in a long period of time such as rainfall. In the architecture, RNN-LSTM Model used 2 Hidden Layer LSTM consisting of 108 LSTM Neurons in each layer. Tanh was used to activation function. Mean Square Error was used to loss function. Adam Optimizer was used to optimization algorithms. The obtained results were the model be able to better predict future rainfall if input data given to model is longer which is indicated by decreasing Root Mean Square Error value. However, the model was less able to predict future rainfall if predicted year output is longer, which is indicated by increasing root mean square error value.

Keywords: Prediction, Rainfall, Recurrent Neural Network, Long Short Term Memory

KATA PENGANTAR

Bismillahirrahmanirrahim, Alhamdulillah robbil ‘alamin. Segala puji dan syukur kepada Allah Subhanahu wa Ta’ala, Tuhan semesta alam yang telah mencurahkan karunia dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “**PREDIKSI CURAH HUJAN MENGGUNAKAN *RECURRENT NEURAL NETWORK - LONG SHORT TERM MEMORY* : STUDI KASUS DI STASIUN BMKG KARANGPLOSO, MALANG**” sebagai salah satu persyaratan untuk memperoleh gelar Sarjana Sains dalam bidang Fisika.

Dengan selesainya karya tulis ini, penulis ingin mengucapkan terima kasih kepada pihak-pihak yang memberikan banyak masukan dan dorongan :

1. Ibu tercinta, Ibu Irtifah dan Kakakku yang paling keren, Muhammad Firdaus dan seluruh keluarga besar yang senantiasa mendoakan dan mendukung penulisnya.
2. Dr. Eng. Agus Naba, S.Si, MT., Ph.D., sebagai dosen pembimbing pertama dan Dr.rer.nat. Abdurro’uf, S.Si., M.Si. selaku pembimbing kedua yang telah memberikan pengarahan dan masukan kepada penulis selama penyusunan skripsi.
3. Ahmad Luthfi, SST, sebagai pembimbing dari BMKG Stasiun Klimatologi Karangploso Kab. Malang yang telah membantu dalam menganalisa data unsur iklim dan curah hujan.
4. BMKG Stasiun Klimatologi Karangploso Kab. Malang yang telah menyediakan data curah hujan untuk keperluan tugas akhir penulis.
5. Segenap Dosen Jurusan Fisika Universitas Brawijaya yang telah memberikan banyak ilmu selama penulis menempuh pendidikan sarjana.
6. Google, Medium, Packt Publishing, O’Reilly Media, Github, Machine Learning Mastery, StackOverflow karena telah menunjang penulisan skripsi ini.
7. Teman-teman Satu Perjuangan Fisika Angkatan 2015, teman-teman Penghuni Rumah Tercinta (*Student Activity Center*), dan semua sahabat saya serta semua pihak-pihak yang tidak dapat penulis sebutkan satu per satu atas kebersamaan, dorongan dan bantuan kepada penulis.
8. Pembaca yang telah meluangkan waktunya untuk membaca skripsi ini.

Penulis sadar bahwa setiap karya manusia tidak ada yang sempurna, sehingga penulis yakin bahwa dalam pembuatan karya tulis ini masih terdapat banyak kekurangan. Penulis berharap bahwa dari para pembaca untuk memberikan kritik dan saran yang bersifat membangun.

Semoga karya tulis ini dapat memberikan manfaat dan inspirasi yang dapat memberikan sumbangan bagi kemajuan ilmu pengetahuan serta peradaban manusia yang baik.

Malang, 31 Desember 2018

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN SKRIPSI	i
LEMBAR PERNYATAAN	iii
ABSTRAK	v
ABSTRACT	vii
KATA PENGANTAR	ix
DAFTAR ISI	xi
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xv
DAFTAR LAMPIRAN	xvii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	4
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	4
BAB II TINJAUAN PUSTAKA	5
2.1 Iklim	5
2.2 Curah Hujan	6
2.3 Pola Hujan Regional	6
2.4 El Nino dan La Nina	9
2.5 Artificial Intelligence	9
2.6 Machine Learning	10
2.7 Supervised Learning	11
2.8 Artificial Neural Networks	12
2.9 Recurrent Neural Networks	13
2.10 Long Short Term Memory (LSTM)	15
2.11 Root Mean Square Error	17
2.12 Python	17
2.13 TensorFlow	17
2.14 Keras	18
BAB III METODE PENELITIAN	19
3.1 Lokasi dan Waktu	19

3.2	Alat dan Bahan	19
3.3	Tahapan Penelitian	19
3.3.1	Persiapan Perangkat Lunak.....	19
3.3.2	Pengambilan Data.....	21
3.3.3	Pengolahan Data	22
3.3.4	Pembuatan Model Neural Network	23
3.3.5	Pelatihan Neural Network.....	24
3.3.6	Pengujian Neural Network	26
BAB IV HASIL DAN PEMBAHASAN		29
4.1	Perbandingan antara RNN-LSTM, FNN dan ARIMA	29
4.2	Prediksi dengan tahun yang berbeda	30
4.3	Prediksi dengan tahun input yang berbeda	34
4.4	Prediksi dengan tahun output yang berbeda	37
BAB V PENUTUP		41
5.1	Kesimpulan.....	41
5.2	Saran	41
DAFTAR PUSTAKA.....		43
LAMPIRAN A DATA HASIL PENELITIAN		45
LAMPIRAN B KODE PROGRAM.....		57

DAFTAR GAMBAR

Gambar 2.1 Tiga iklim regional berdasarkan rata-rata pola hujan tahunan..	6
Gambar 2.2 Pola tahunan curah hujan pada tiga iklim regional	8
Gambar 2.3 Perbandingan AI, Machine Learning dan Deep Learning	10
Gambar 2.4 Alur kerja supervised learning	11
Gambar 2.5 Skematik dari sebuah artificial neural networks	13
Gambar 2.6 RNNs	14
Gambar 2.7 Arsitektur LSTM	15
Gambar 3.1 Install Python Interpreter	20
Gambar 3.2 Data Curah Hujan dari tahun 1989 sampai 2017	22
Gambar 3.3 Model Stacked RNN-LSTM	24
Gambar 3.4 Proses Pelatihan Model.....	25
Gambar 4.1 Perbandingan antara RNN-LSTM, FNN dan ARIMA	30
Gambar 4.2 Grafik tren prediksi dengan tahun yang berbeda	32
Gambar 4.3 Grafik Prediksi 2008.....	33
Gambar 4.4 Grafik Prediksi 2017	34
Gambar 4.5 Grafik tren prediksi dengan tahun input yang berbeda	35
Gambar 4.6 Grafik prediksi 2017 dengan input 5 tahun.....	36
Gambar 4.7 Grafik prediksi 2017 dengan input 28 tahun.....	37
Gambar 4.8 Grafik tren prediksi dengan tahun output yang berbeda	38
Gambar 4.9 Grafik prediksi selama 1 tahun	39
Gambar 4.10 Grafik prediksi selama 5 tahun	39
Gambar A.1.1 Prediksi 2008	45
Gambar A.1.2 Prediksi 2009	45
Gambar A.1.3 Prediksi 2010	46
Gambar A.1.4 Prediksi 2011	46
Gambar A.1.5 Prediksi 2012	47
Gambar A.1.6 Prediksi 2013	47
Gambar A.1.7 Prediksi 2014	48
Gambar A.1.8 Prediksi 2015	48
Gambar A.1.9 Prediksi 2016	49
Gambar A.1.10 Prediksi 2017	49

Gambar A.2.1 Prediksi 2017 dengan input 5 tahun.....	50
Gambar A.2.2 Prediksi 2017 dengan 10 tahun	50
Gambar A.2.3 Prediksi 2017 dengan 15 tahun	51
Gambar A.2.4 Prediksi 2017 dengan 20 tahun	51
Gambar A.2.5 Prediksi 2017 dengan 25 tahun	52
Gambar A.2.6 Prediksi 2017 dengan 28 tahun	52
Gambar A.3.1 Prediksi dengan output selama 1 tahun.....	53
Gambar A.3.2 Prediksi dengan output selama 2 tahun.....	53
Gambar A.3.3 Prediksi dengan output selama 3 tahun.....	54
Gambar A.3.4 Prediksi dengan output selama 4 tahun.....	54
Gambar A.3.5 Prediksi dengan output selama 5 tahun.....	55

DAFTAR TABEL

Tabel 3.1 Data Curah Hujan Bulanan tahun 1989 – 2017	21
Tabel 3.2 Data Hasil Supervised Learning	23
Tabel 3.3 Pengujian dengan tahun yang berbeda	26
Tabel 3.4 Pengujian dengan input yang berbeda	27
Tabel 3.5 Pengujian dengan tahun output yang berbeda	27
Tabel 4.1 Perbandingan antara RNN-LSTM, FNN dan ARIMA	29
Tabel 4.2 Prediksi dengan tahun yang berbeda	31
Tabel 4.3 Prediksi dengan input yang berbeda	35
Tabel 4.4 Prediksi dengan tahun output yang berbeda	37

DAFTAR LAMPIRAN

LAMPIRAN A DATA HASIL PENELITIAN	45
LAMPIRAN B KODE PROGRAM.....	57

BAB I

PENDAHULUAN

1.1 Latar Belakang

Allah Ta'ala berfirman dalam Surah Al-Mukminun ayat 18 :

وَأَنْزَلْنَا مِنَ السَّمَاءِ مَاءً بِقَدَرٍ فَأَسْكَنَّاهُ فِي الْأَرْضِ وَإِنَّا عَلَى ذَهَابٍ بِهِ لَقَادِرُونَ

Dan Kami turunkan air (hujan) dari langit menurut suatu ukuran; lalu Kami jadikan air itu menetap di bumi, dan sesungguhnya Kami benar-benar berkuasa menghilangkannya.
(QS. 23:18)

Pada ayat ini Allah Ta'ala menjelaskan tentang kekuasaan-Nya dalam menurunkan air hujan serta mengatur hujan yang turun sesuai dengan takaran.

Ibnu Katsir di dalam tafsirnya berkata: “Allah Ta'ala menyebutkan berbagai macam nikmat-Nya yang dilimpahkan kepada hamba-Nya yang tiada terhingga jumlahnya dan tidak juga dapat dihitung tetesan air yang diturunkan dari langit. Yakni, sesuai dengan kebutuhan, tidak berlebihan yang hanya akan merusak bumi dan bangunan, dan tidak juga terlalu sedikit sehingga tidak cukup untuk mengairi tanaman dan buah-buahan, tetapi sesuai dengan apa yang dibutuhkan” (Ibnu Katsir, 2004). Apabila manusia mampu memahami fenomena hujan ini dengan baik, mulai dari sebabnya turun hujan, peristiwa terjadinya hujan, kapan prakiraan hujan turun dan akibat yang ditimbulkan dari turunnya hujan maka akan didapatkan sebuah manfaat yang besar bagi masyarakat dan lingkungan di sekitarnya.

Dengan mengetahui informasi prakiraan hujan, dapat membantu pengaturan sumber daya air di suatu tempat khususnya ketika berhubungan dengan munculnya perubahan iklim di daerah tropis seperti Indonesia. Perubahan iklim telah mempengaruhi pola curah hujan sehingga menyebabkan terjadinya beberapa bencana alam seperti hujan sangat deras yang menghasilkan banjir atau tidak adanya hujan

dalam waktu lama yang menghasilkan kekeringan. Untuk menangani masalah tersebut, prediksi curah hujan dengan metode yang baik serta akurat merupakan hal yang sangat diperlukan untuk mengantisipasinya (Mislan, et al, 2015).

Salah satu metode yang sering digunakan secara luas dalam memprediksi data deret waktu (*time series*) adalah *Autoregressive Integrated Moving Average* (ARIMA). Telah banyak penelitian yang melakukan prediksi curah hujan menggunakan ARIMA, salah satunya pernah dilakukan di daerah Lanzhou, China (Wang, et al, 2014).

Sepanjang pengembangan teknologi komputasi, banyak peneliti telah mengembangkan metode ANN (Artificial Neural Network) untuk membuat prediksi di dalam bidang hidrologi. Mislan, H. et al, telah melakukan penelitian dengan ANN dengan model FNN (*Feedforward Neural Network*) untuk memprediksi curah hujan di daerah Tenggarong, Kalimantan Timur (Mislan, et al, 2015). Pada penelitian yang lain di Nagpur, India juga memprediksi data curah hujan untuk 30 tahun dari 1977 sampai 2006 (Charaniya, et al, 2013).

Model FNN (*Feedforward Neural Network*) yang digunakan pada penelitian sebelumnya merupakan model yang bersifat umum. Pada kasus curah hujan diperlukan model yang lebih baik dalam mengolah data deret waktu (*time series*). Model RNN (Recurrent Neural Network) merupakan model yang cocok untuk mengatasi permasalahan tersebut. Namun, RNN memiliki kekurangan dalam memprediksi data yang bersifat jangka panjang seperti curah hujan dari tahun ke tahun.

Untuk menangani masalah tersebut, digunakanlah model RNN tipe LSTM (*Long Short-Term Memory*) dimana arsitekturnya dapat menangani permasalahan yang berhubungan dengan waktu jangka panjang. LSTM memiliki perbedaan dengan Tradisional RNN (*Recurrent Neural Network*) dimana LSTM mampu dalam memproses serta memprediksi data yang berurutan terhadap waktu tanpa kehilangan informasi yang penting. Zaytar, et al, telah melakukan penelitian menggunakan multi stacked LSTM untuk memprediksi data cuaca

(Suhu, Kelembaban, Kecepatan Angin) dalam waktu 24 atau 72 jam di wilayah Turki. Data yang digunakan untuk proses pelatihan adalah data meteorologi dalam hitungan jam selama 15 tahun. Hasil yang didapatkan bahwa LSTM memberikan hasil yang baik dalam memprediksi kondisi cuaca secara umum (Zaytar, 2016).

Berdasarkan penelitian prediksi curah hujan yang sudah dilakukan menggunakan ARIMA dan FNN, penulis berusaha untuk membuat sistem prakiraan baru yang dapat memprediksi curah hujan beberapa tahun ke depan di wilayah Malang menggunakan RNN - *Long Short-Term Memory* yaitu sistem jaringan syaraf buatan yang didesain khusus untuk menangani permasalahan deret waktu dalam jangka waktu yang panjang. Selanjutnya, hasil prediksi RNN-LSTM akan dibandingkan dengan hasil prediksi FNN dan ARIMA untuk diketahui seberapa baik model ini dalam memprediksi curah hujan.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, rumusan masalah dalam penelitian ini adalah:

1. Bagaimana hasil keakuratan prediksi pada model RNN-LSTM jika dibandingkan dengan model *Feedforward Neural Network* (FNN) dan *Autoregressive Integrated Moving Average* (ARIMA)?
2. Bagaimana kemampuan model RNN-LSTM ketika memprediksi tahun yang berbeda?
3. Bagaimana kemampuan model RNN-LSTM ketika memprediksi dengan data *input* yang semakin banyak?
4. Bagaimana kemampuan model RNN-LSTM ketika memprediksi curah hujan dalam beberapa tahun mendatang?

1.3 Batasan Masalah

Ruang lingkup dalam penelitian ini dibatasi oleh beberapa hal sebagai berikut:

1. Curah hujan yang diamati adalah daerah Karangploso, Malang dan sekitarnya.
2. Penentuan prediksi curah hujan hanya menggunakan data curah hujan bulanan dari tahun 1989 sampai 2017.
3. Penentuan prediksi curah hujan ini tidak melibatkan faktor-faktor penyebab hujan (seperti Temperatur Permukaan Laut, Kelembaban Udara, Angin Muson).

1.4 Tujuan Penelitian

Tujuan Penelitian yang dilakukan adalah :

1. Membandingkan kemampuan prediksi antara RNN-LSTM, FNN dan ARIMA dengan mengamati seberapa besar RMSE yang diperoleh ketika memprediksi curah hujan.
2. Mengetahui kemampuan RNN-LSTM ketika memprediksi tahun yang berbeda.
3. Mengetahui kemampuan RNN-LSTM dalam prediksi ketika menggunakan data *input* yang semakin banyak.
4. Mengetahui kemampuan RNN-LSTM ketika memprediksi curah dalam beberapa tahun mendatang.

1.5 Manfaat Penelitian

Manfaat yang didapatkan pada penelitian ini adalah :

1. Bagi Instansi Pemerintah (BMKG) : Hasil model RNN-LSTM akan diserahkan kepada BMKG untuk digunakan sebagai alat bantu analisis prediksi iklim dan musim.
2. Bagi Peneliti di bidang Iklim dan Cuaca : Hasil model RNN-LSTM akan membantu peneliti bidang iklim dan cuaca dalam menganalisa prediksi curah hujan tahun depan.

BAB II

TINJAUAN PUSTAKA

2.1 Iklim

Iklim merupakan kebiasaan cuaca yang terjadi di suatu daerah yang mana iklim tersebut memberikan ciri kecuaan tertentu di suatu daerah dan bukan cuaca rata-rata. Iklim berkaitan dengan periode waktu panjang yang tidak tentu. Iklim tidak mungkin dikatakan dalam periode yang pendek seperti iklim hari ini, iklim besok pagi maupun iklim minggu depan.

Ciri kecuaan suatu daerah ditetapkan berdasarkan kriteria keseringan atau kemungkinan sering munculnya nilai-nilai satu atau lebih unsur iklim yang ditetapkan, misalnya: hujan, suhu, suhu dan hujan, suhu dan angin, hujan dan penguapan.

Dalam aspek wilayah, iklim terbagi menjadi 5 macam yaitu Iklim Kutub (*Polar Climate*), Iklim Tengah (*Temperate Climate*), Iklim Subtropis (*Subtropical Climate*), Iklim Tropis (*Tropical Climate*), Iklim Khatulistiwa (*Equatorial Climate*). Iklim Tropis merupakan iklim di kawasan tropis yang dicirikan dengan suhu selalu tinggi dengan variasi tahunannya kecil dan hujan hampir terjadi di sembarang waktu dalam setahun. Iklim Khatulistiwa dicirikan dengan variasi suhu harian kecil dan hujan di sembarang waktu dan dalam setahun terjadi hujan maksimum sebanyak dua kali.

Dalam aspek lingkungan iklim yang dibentuk, iklim dibagi menjadi 6 macam yaitu Iklim Benua (*Continental Climate*), Iklim Bahari (*Maritime/Marine Climate*), Iklim Monsun (*Monsoon Climate*), Iklim Mediteran (*Mediterranean Climate*), Iklim Tundra (*Tundra Climate*), Iklim Gunung (*Mountain Climate*). Iklim monsun merupakan jenis iklim di kawasan monsun yang dicirikan dengan perubahan unsur-unsur iklim secara musiman. Hujan banyak terjadi pada kawasan tersebut. Umumnya dimiliki oleh daerah yang berada di kawasan tropik(Wirjohamidjojo, 2010).

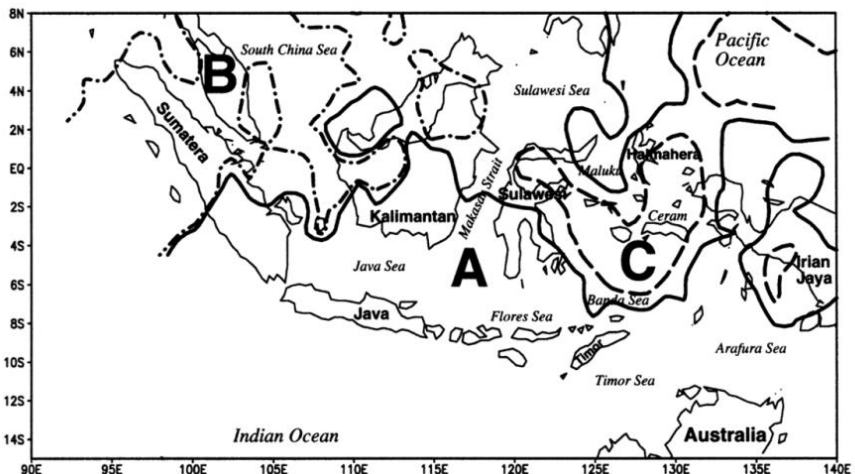
2.2 Curah Hujan

Presipitasi didefinisikan sebagai bentuk cairan (air) dan padatan (es) yang jatuh menuju permukaan bumi. Bentuk presipitasi adalah hujan, gerimis, salju dan hujan batu es (*hail*). Hujan adalah bentuk presipitasi yang sering dijumpai dan di Indonesia yang dimaksud dengan presipitasi adalah curah hujan.

Curah hujan dan suhu merupakan unsur iklim yang sangat penting bagi kehidupan di bumi. Jumlah hujan dicatat dalam inci atau milimeter (1 inci = 25,4 mm). Jumlah curah hujan 1 mm menunjukkan bahwa tinggi air hujan yang menutupi permukaan setinggi 1 mm, jika air tersebut tidak meresap ke dalam tanah atau menguap ke atmosfer (Tjasyono, 2004).

2.3 Pola Hujan Regional

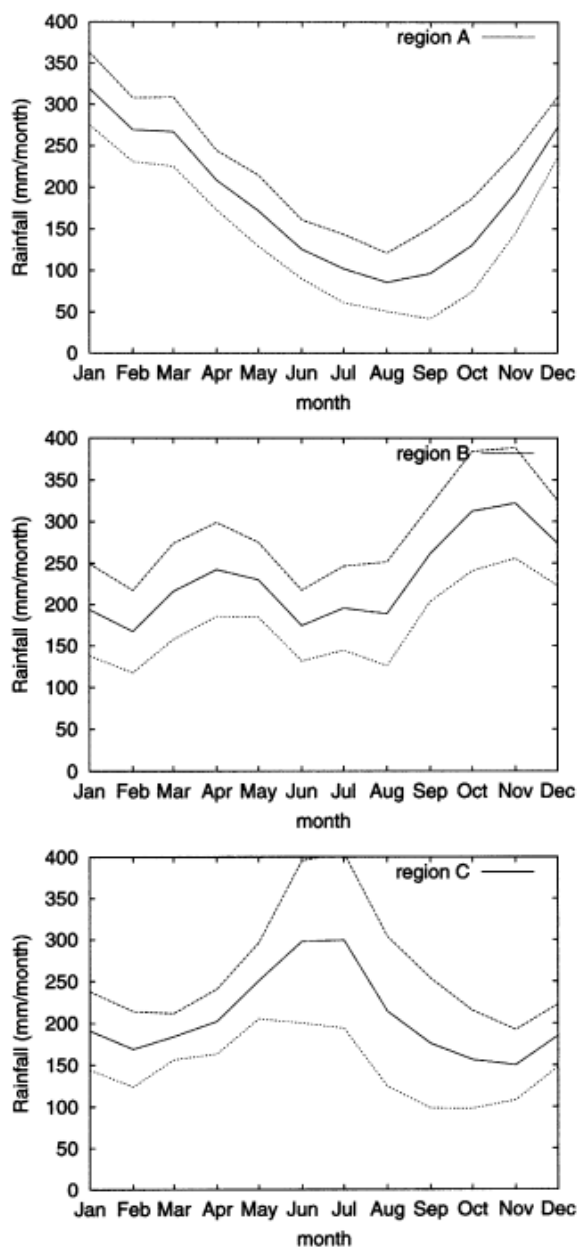
Indonesia memiliki pola hujan tahunan yang berbeda pada tiga iklim regional. Regional A berlokasi di daerah Indonesia bagian selatan mulai dari Sumatra Selatan sampai Pulau Timor, Kalimantan bagian Selatan, Sulawesi dan Irian Jaya. Regional B berlokasi di Indonesia bagian barat laut dari Sumatra bagian Utara sampai Kalimantan bagian Barat Laut. Regional C berlokasi di Maluku dan Sulawesi bagian Utara.



Gambar 2.1 Tiga iklim regional berdasarkan rata-rata pola hujan tahunan

Telah terdapat beberapa penelitian pada variabilitas curah hujan di Kepulauan Indonesia dan hubungannya dengan fenomena iklim skala besar seperti *El Nino Southern Oscillation* (ENSO) dan muson. Indonesia terdiri dari lebih 17000 pulau, yang berlokasi antara samudra Pasifik dan Indian serta diantara benua Asia dan Australia, pengaruh ENSO bervariasi di seluruh regional yang disebabkan oleh topografi pulau dan fluks samudra-atmosfer yang mana dipengaruhi oleh variabilitas suhu permukaan laut.

Berdasarkan Gambar 2.2, Regional A memiliki satu puncak dan satu lembah yang dipengaruhi kuat oleh dua muson yang bernama muson basah barat laut (*North-West*) dari November sampai Maret dan muson kering tenggara (*South-East*) dari Mei sampai September. Regional B memiliki dua puncak, pada bulan Oktober-November dan pada bulan Maret sampai Mei. Regional C memiliki satu puncak di bulan Juni sampai Juli dan satu lembah dari bulan November sampai Februari (Aldrian, 2003).



Gambar 2.2 Pola tahunan curah hujan pada tiga iklim regional

2.4 El Nino dan La Nina

Fenomena El Nino dan La Nina merupakan peristiwa anomali iklim global, gejala munculnya El Nino biasanya dicirikan dengan meningkatnya suhu permukaan laut di kawasan Pasifik secara berkala dengan durasi waktu tertentu dan meningkatnya perbedaan tekanan udara antara Darwin dan Tahiti.

Situasi iklim di wilayah nusantara pada dasarnya dipengaruhi oleh sirkulasi monsoon yang menimbulkan perbedaan iklim antara musim hujan dan musim kemarau. Curah hujan akan dipengaruhi oleh sirkulasi monsoon dan sirkulasi monsoon dipengaruhi oleh kejadian ENSO (*El Nino Southern Oscillation*) yang mana terdiri dari Fenomena El Nino dan La Nina. Fluktuasi hujan sangat berpengaruh sangat berkorelasi dengan kejadian ENSO. Dengan kata lain, fenomena El Nino dapat berdampak pada penurunan curah hujan dan fenomena La Nina dapat menimbulkan peningkatan curah hujan(Irawan, 2006).

2.5 Artificial Intelligence

Para Ilmuwan memiliki dua sudut pandang yang berbeda tentang definisi AI (*Artificial Intelligence*). Yang pertama, AI merupakan bidang ilmu yang hanya fokus pada proses berpikir. Yang kedua, AI merupakan bidang ilmu yang berfokus pada proses tingkah laku.

Definisi AI yang paling tepat saat ini adalah *acting rationally* dengan pendekatan *rational agent*. Komputer dapat melakukan penalaran secara logis serta dapat melakukan aksi secara rasional berdasarkan hasil penalaran yang telah dibuatnya (Suyanto, 2014).

Terdapat empat komposisi dasar untuk pembuatan AI: data, sumber daya komputasi (*hardware*), algoritma (*software*), dan keahlian untuk menyatukan semuanya (O'Reilly Media, 2017).

Kita bisa melihat bahwa *Artificial Intelligence* mencakup semua algoritma pembelajaran (*learning algorithm*), termasuk regresi, klasifikasi dan klustering serta tugas yang bersifat kognitif seperti *reasoning*, *planning* dan *navigation*.

Tujuan dari AI adalah membangun mesin yang mampu melakukan segala sesuatu yang dapat dilakukan manusia dan lebih baik.

Otak manusia memiliki sekitar 100 miliar neuron, adalah karya rekayasa biologi yang sangat canggih dan sangat kompleks, mungkin tidak perlu terlalu terkejut jika belum berhasil mereplikasi semua fitur-fiturnya ke dalam sebuah mesin. Tetapi para insinyur AI telah membuat kemajuan dalam mengembangkan AI. Pencarian ini dikenal sebagai *artificial general intelligence*, atau AGI, dan tujuan utamanya adalah untuk merancang dan membangun *artificial superintelligence*, atau ASI (Morgan, 2018).

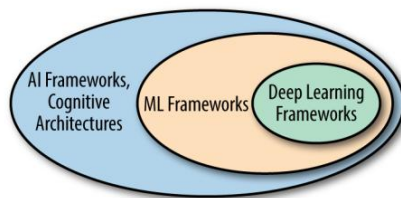
2.6 Machine Learning

Gambar 2.3 menjelaskan bahwa dalam lingkaran AI adalah *machine learning* dengan berbagai macam algoritmanya, termasuk *support vector machines*, *K-means clustering*, *random forests*, dan ratusan lainnya yang telah dikembangkan selama beberapa dekade terakhir (Morgan, 2018).

Machine learning adalah salah satu dari cabang AI, dan lebih khusus cabang dari ilmu computer (*computer science*), yang mana berhubungan dengan mempelajari sistem dan algoritma yang bisa dipelajari dari data, mengganti pengetahuan baru darinya.

Kata belajar secara intuitif menunjukkan bahwa sistem yang berdasarkan pada *machine learning*, atas dasar pengamatan data yang diproses sebelumnya, meningkatkan pengetahuannya untuk mencapai hasil yang lebih baik di masa depan atau memberikan *output* lebih dekat ke *output* yang diinginkan untuk sistem tertentu (Zaccone, 2016).

Machine learning bekerja dari kumpulan data besar, dengan mesin yang belajar dari data, mirip dengan cara manusia dan otak biologis memproses informasi.

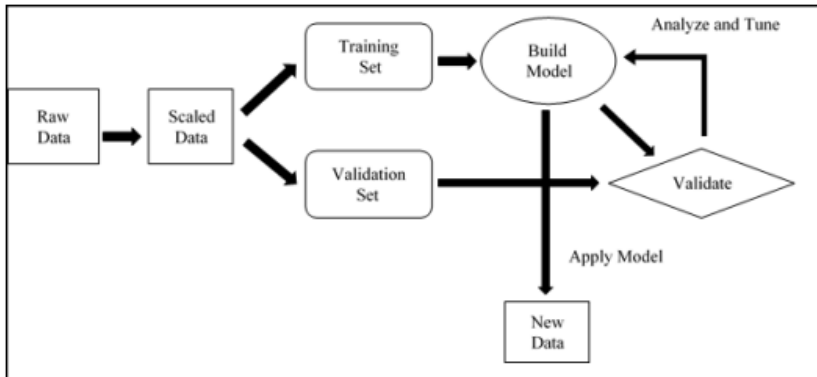


Gambar 2.3 Perbandingan AI, Machine Learning dan Deep Learning

Melalui kerangka kerja *machine learning* didapatkan *deep learning*, yang juga diketahui sebagai *Artificial Neural Networks* (ANNs) karena algoritma tersebut dimodelkan pada bagaimana otak memproses data, meskipun saat ini dalam kerangka kerja yang disederhanakan (Morgan, 2018).

2.7 Supervised Learning

Supervised learning adalah bentuk yang paling umum dari *machine learning*. Seperti yang dijelaskan pada Gambar 2.4, sekumpulan contoh dan set data pelatihan dikirimkan sebagai *input* ke sistem selama fase pelatihan yang mana setiap contoh diberi label dengan nilai *output* yang diinginkan. Sebagian besar Algoritma *supervised learning* memiliki satu karakteristik: proses pelatihan dilakukan dengan meminimalkan fungsi kesalahan tertentu yang disebut *loss function*, yang mana mewakili kesalahan *output* sehubungan dengan sistem *output* yang diinginkan.



Gambar 2.4 Alur kerja supervised learning

Loss function yang paling sering digunakan untuk tipe pelatihan ini adalah menghitung *standart deviation* antara *output* yang diinginkan dan yang dihasilkan oleh sistem. Setelah pelatihan selesai, akurasi model diukur dengan satu set contoh yang terpisah dari set data pelatihan, yang disebut dengan set data validasi (Zaccone, 2016).

Dalam fase ini kemampuan generalisasi model kemudian diverifikasi, model akan diuji coba jika *output* benar untuk *input* yang tidak digunakan selama proses pelatihan (Zaccone, 2016).

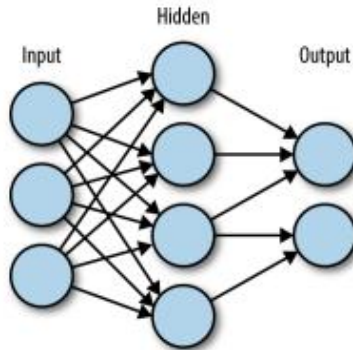
2.8 Artificial Neural Networks

Mungkin seringkali manusia memikirkan bagaimana hebatnya sistem syaraf otak mereka dapat bekerja untuk mengenali berbagai macam pola yang ada di alam ini.

Misalkan, seorang direktur perusahaan yang masih bisa mengenali beberapa wajah karyawannya meskipun jarang sekali berjumpa dengannya dan direktur tersebut banyak menemui wajah wajah karyawan lainnya. Sang Direktur tersebut juga dapat membedakan ciri dari setiap karyawannya dari suara, gerakan tubuh, kebiasaan mereka dan lain-lain. Mungkinkah manusia dapat membuat sistem syaraf buatan yang diterapkan ke dalam sebuah teknik komputasi?

Artificial Neural Networks merupakan salah satu upaya manusia untuk memodelkan cara kerja atau fungsi sistem syaraf manusia dalam melakukan tugas-tugas tertentu. Pemodelan ini didasari oleh kemampuan otak manusia dalam mengorganisir sel-sel neuron.

Artificial Neural Networks (ANNs) adalah sistem prosesor yang terdistribusi secara paralel yang sangat besar dengan memiliki kecenderungan untuk menyimpan pengetahuan yang bersifat pengalaman dan membuatnya siap digunakan. ANNs menyerupai otak manusia dalam dua hal yaitu: Pengetahuan yang diperoleh dari sistem jaringan syaraf melalui proses belajar; kekuatan hubungan antar sel syaraf (neuron) yang disebut sebagai bobot sinaptik yang berguna dalam penyimpanan pengetahuan (Suyanto, 2014).



Gambar 2.5 Skematik dari sebuah artificial neural networks

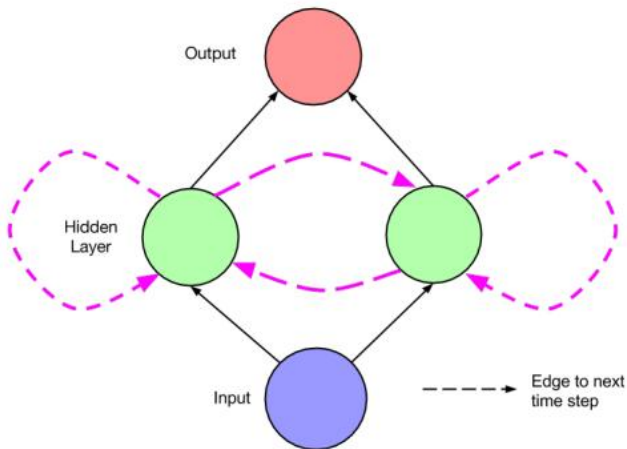
Gambar 2.5 menjelaskan bahwa ANN terdiri dari 2 komponen penting yaitu *node* yang berbentuk lingkaran dan *edge* yang berbentuk garis lurus. Proses kerja ANN diambil dari cara kerja Teori Graf. Proses komputasi terjadi pada masing-masing *node*, yang disebut sebagai pemrosesan paralel secara masif (Morgan, 2018).

2.9 Recurrent Neural Networks

Recurrent Neural Networks (RNN) adalah model sistem jaringan syaraf dengan kemampuan secara selektif memberikan informasi di seluruh langkah-langkah yang berurutan, sambil memproses data sekuensial satu per satu. Dengan demikian mereka dapat memodelkan *input* dan / atau *output* yang terdiri dari urutan elemen yang tidak independen. Lebih lanjut lagi, RNN dapat secara simultan memodelkan ketergantungan sekuensial dan waktu pada berbagai skala.

Input yang menuju RNNs adalah sebuah data yang berurutan, dan/atau targetnya adalah data yang berurutan pula. Sebuah urutan *input* dapat dilambangkan $(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(T)})$ dimana masing-masing titik data $\mathbf{x}^{(t)}$ adalah vektor yang bernilai riil. Demikian pula, urutan target dapat ditandai sebagai $(\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \dots, \mathbf{y}^{(T)})$. Sebuah set pelatihan secara

husus adalah set contoh dimana masing-masing contoh adalah sebuah pasangan *input* yang berurutan, target yang berurutan, meskipun umumnya *input* atau *output* mungkin merupakan titik data tunggal. Urutan mungkin memiliki panjang yang terbatas atau tidak terbatas.

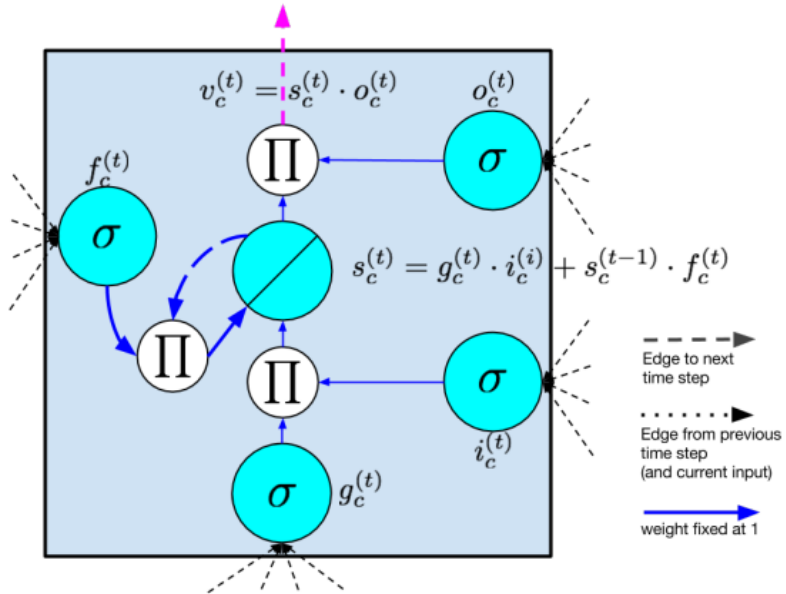


Gambar 2.6 RNN

Pada Gambar 2.6 menjelaskan bahwa *edges* yang menghubungkan setiap langkah waktu yang berdekatan disebut *recurrent edges*, dapat berbentuk siklus panjang yang merupakan hubungan dari sebuah *node* menuju dirinya sendiri sepanjang waktu. Pada waktu t , *nodes* dengan *recurrent edges* menerima *input* dari data saat ini $\mathbf{x}^{(t)}$ dan nilai dari *hidden nodes* $\mathbf{h}^{(t-1)}$ dalam jaringan yang sebelumnya. *Output* $\hat{\mathbf{y}}^{(t)}$ pada tiap waktu dihitung untuk memberikan nilai *hidden nodes* $\mathbf{h}^{(t)}$ pada tiap waktu. *Input* pada $t - 1$ bisa mempengaruhi nilai *output* $\hat{\mathbf{y}}^{(t)}$ pada waktu t (Lipton, Zachary C, et al, 2015).

2.10 Long Short Term Memory (LSTM)

Long Short-Term Memory (LSTM) ditemukan oleh Hochreiter dan Schmidhuber pada tahun 1997 dengan memperkenalkan sel memori, sebuah unit komputasi yang menggantikan *node* tradisional dalam *hidden layer* dari sebuah *neural network* yang ditujukan untuk menanggapi masalah *vanishing gradient* yang muncul ketika proses *training* pada RNNs.



Gambar 2.7 Arsitektur LSTM

Istilah LSTM berasal dari RNN sederhana yang memiliki memori jangka panjang dalam bentuk bobot (*weight*). Bobot berubah perlahan selama proses pelatihan berlangsung, melakukan pengkodean pengetahuan tentang data. LSTM juga memiliki memori jangka pendek dalam bentuk aktivasi sementara, yang mana melewati dari masing-masing *node* menuju ke *node* yang berturut-turut. Model LSTM memperkenalkan jenis penyimpanan melalui sel memori (Lipton, Zachary C, et al, 2015).

Persamaan 2.1 menjelaskan proses LSTM dalam mengolah data input menjadi output. Mula-mula nilai *Input Node* (g), *Input Gate* (i), *Forget Gate* (f), *Output Gate* (o) dihitung terlebih dahulu. *Internal State* (s) didapatkan dengan mengakumulasi nilai g , i , f serta s yang sebelumnya. Kemudian nilai akhir dari memori sel (h) akan digunakan untuk mengubah variabel pada proses LSTM selanjutnya.

$$\begin{aligned}
g^{(t)} &= \phi(W^{gx}x^{(t)} + W^{gh}h^{(t-1)} + b_g) \\
i^{(t)} &= \sigma(W^{ix}x^{(t)} + W^{ih}h^{(t-1)} + b_i) \\
f^{(t)} &= \sigma(W^{fx}x^{(t)} + W^{fh}h^{(t-1)} + b_f) \\
o^{(t)} &= \sigma(W^{ox}x^{(t)} + W^{oh}h^{(t-1)} + b_o) \\
s^{(t)} &= g^{(t)} \odot i^{(i)} + s^{(t-1)} \odot f^{(t)} \\
h^{(t)} &= \phi(s^{(t)}) \odot o^{(t)}.
\end{aligned}
\tag{Persamaan 2.1}$$

Unit komputasi dari LSTM disebut sebagai sel memori, blok memori atau hanya disebut sel saja. Masing-masing memori terdiri satu atau lebih sel-sel memori yang terhubung secara berulang dan terdapat tiga unit multiplikatif – *input gates*, *output gates* dan *forget gates*, ketiga ini memberikan sebuah analogi secara kontinyu untuk menulis, membaca dan reset operasi pada sel sel yang ada.

Long Short-Term Memory (LSTM) adalah salah satu jenis yang paling menarik dari *deep learning* saat ini. LSTM telah digunakan untuk mendemonstrasikan hasil dari berbagai macam masalah kompleks seperti *language translation*, *automatic image captioning* dan *text generation*.

LSTM sangatlah berbeda dengan teknik *deep learning* yang lain, seperti *Multilayer Perceptrons* (MLP) dan *Convolutional Neural Networks* (CNN), karena LSTM dirancang khusus untuk masalah prediksi yang berurutan (Brownlee, 2017).

2.11 Root Mean Square Error

Root Mean Squared Error (RMSE) adalah aturan penskoran kuadrat yang digunakan untuk mengukur besaran kesalahan rata-rata. RMSE adalah akar kuadrat dari rata-rata perbedaan kuadrat antara prediksi dan observasi yang sesungguhnya sebagaimana yang ditunjukkan oleh Persamaan 2.2. RMSE telah digunakan sebagai metrik statistik standar untuk mengukur kinerja model dalam penelitian meteorologi, kualitas udara, dan penelitian iklim (Chai, T., et al, 2014).

$$RMSE = \sqrt{\frac{\sum(y^a - y^p)^2}{n}} \quad (\text{Persamaan 2.2})$$

2.12 Python

Python dapat digunakan untuk lebih dari sekedar keperluan pemrograman secara umum. Python adalah bahasa yang memiliki potensi luar biasa untuk digunakan dalam domain komputasi ilmiah. Python adalah alat yang efektif untuk digunakan saat menggabungkan komputasi ilmiah dan matematika (Fuhrer, 2016).

2.13 TensorFlow

Sistem TensorFlow digunakan untuk *machine learning* yang sebagian besar karena fleksibilitasnya dan keumuman dalam pemrogramannya. Pendekatan ini mendukung berbagai aplikasi *machine learning*, termasuk pelatihan dan penyimpulan dengan *deep neural network* pada sistem yang terdistribusi secara heterogen.

TensorFlow mewakili kerja komputasi yang berdasarkan aliran data graf. Secara khusus, sebuah komputasi bisa jadi melakukan 1 atau lebih langkah dalam proses pelatihan (*training*) untuk membuat sebuah model *machine learning* atau mungkin penerapan model yang sudah dilatih. Dengan demikian, aliran data graf mendukung proses pelatihan dan penyimpulan.

Sebuah aliran data graf terdiri dari *node* dan *edge*, dimana setiap *node* merepresentasikan suatu proses dari operasi, dan nilai mengalir di sepanjang *edge*. Operasi diimplementasikan oleh *kernel* yang dapat

dijalankan pada jenis perangkat tertentu (misalnya, CPU atau GPU) (Abadi, 2017).

2.14 Keras

Keras adalah *library* dari bahasa pemrograman Python untuk keperluan *deep learning* yang dapat berjalan di atas Theano atau TensorFlow. Keras dikembangkan untuk membuat menerapkan model *deep learning* secepat dan semudah mungkin untuk penelitian dan pengembangan. Keras berjalan pada Python 2 atau 3 dan dapat dijalankan secara mulus pada GPU dan CPU yang diberikan kerangka kerja yang mendasarinya (Brownlee, 2016).

BAB III

METODE PENELITIAN

3.1 Lokasi dan Waktu

Penelitian dilaksanakan pada tiga tempat yang berbeda yaitu di Stasiun BMKG Karangploso Kab. Malang, Laboratorium Komputasi dan Pemodelan Jurusan Fisika Univ. Brawijaya Gedung Biomol lantai tiga, dan kediaman penulis yang berada di Singosari Kab. Malang.

3.2 Alat dan Bahan

Alat yang digunakan dalam penelitian ini adalah sebuah Laptop HP 14-r017TX Notebook PC, dengan Spesifikasi Processor Intel(R) Core(TM) i3-4030U CPU @ 1.90 GHz, 1901 MHz, 2 Core(s), 4 Logical Processor(s), RAM 6.00 GB, Hard Drive 500 GB.

Bahan yang digunakan dalam penelitian adalah Data Curah Hujan Bulanan dari tahun 1989 sampai 2017 di Stasiun Klimatologi BMKG Karangploso, Kab. Malang.

3.3 Tahapan Penelitian

Penelitian ini dilakukan dalam beberapa tahap sebagai berikut:

1. Persiapan Perangkat Lunak (*Software*)
2. Pengambilan Data
3. Pengolahan Data
4. Pembuatan Model *Neural Network*
5. Pelatihan *Neural Network* (*Training*)
6. Pengujian *Neural Network* (*Testing*)

3.3.1 Persiapan Perangkat Lunak

Sebelum memulai penelitian, Program Prediksi Curah Hujan dibuat menggunakan Platform Deep Learning milik Google yang bernama Tensorflow, Tensorflow dibangun pada bahasa pemrograman Python. Diperlukan penginstalan Python Interpreter agar *script* program tersebut dapat diolah.

Python Interpreter dapat diunduh melalui *website* resmi Python yaitu <https://www.python.org/downloads/>. Setelah diunduh lakukan penginstalan *software* seperti biasanya. Pada Gambar 3.1, pilih untuk

centang “Add Python 3.7 to PATH” dan klik “Install Now”, tunggu beberapa saat hingga instalasi selesai.



Gambar 3.1 Install Python Interpreter

Buka *command prompt windows*, lakukan instalasi python *library* yang diperlukan agar program dapat dirancang. Berikut beberapa *library* python yang diperlukan :

- Tensorflow
- Keras
- Matplotlib
- Pandas
- Scikit-learn

Untuk melakukan instalasi *library* python gunakan perintah “pip” melalui *command prompt* sebagai berikut:

```
pip install tensorflow
pip install keras
pip install matplotlib
pip install pandas
pip install scikit-learn
```

Setelah semua *library* terpasang, program siap untuk dibuat dan dijalankan melalui Python Interpreter.

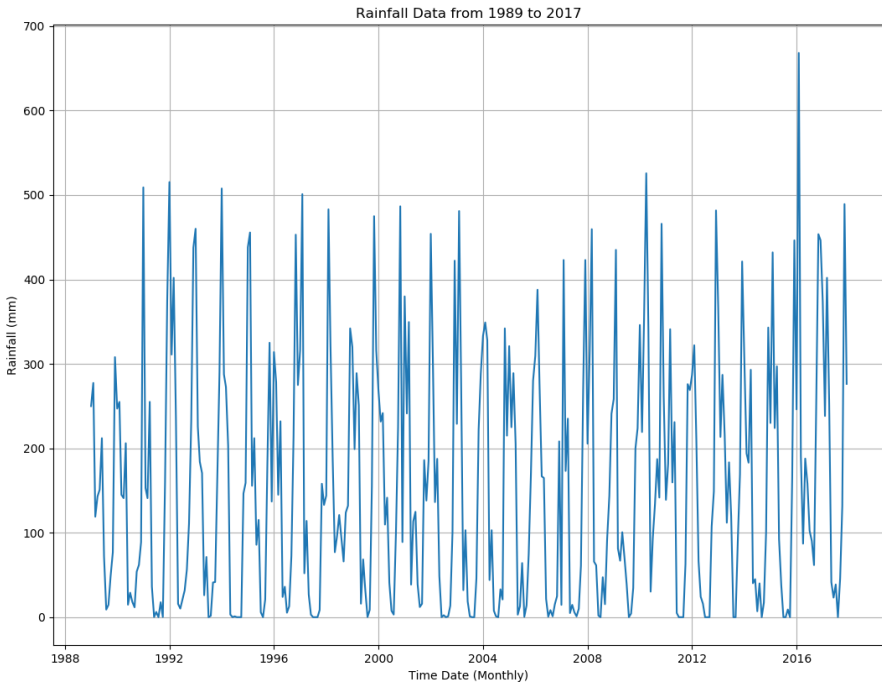
3.3.2 Pengambilan Data

Data yang didapatkan dari Stasiun Klimatologi BMKG Karangploso berformatkan *file* Microsoft Excel (.xlsx). *File* tersebut berisi data curah hujan dasarian dan bulanan dari awal tahun 1981 sampai tahun 2018 bulan agustus untuk 15 pos pengamatan yang ada di daerah Malang yaitu Dampit, Pujon, Ngantang, Bantur, Stasiun Klimatologi Karangploso, Donomulyo, Tlekung, Stasiun Geofisika Karangkates, Sitiarjo, Karangsuko, Jabung, Lawang, Poncokusumo, Tajinan, Lanud Abd Saleh.

Penelitian ini menggunakan data curah hujan bulanan yang ada pada Stasiun Klimatologi Karangploso dari tahun 1989 sampai 2017 (selama 29 tahun) yang ditampilkan pada Tabel 3.1. Data yang digunakan adalah sepanjang 29 tahun yang diharapkan agar model *neural network* semakin banyak mempelajari berbagai curah hujan ekstrim serta kekeringan yang disebabkan oleh fenomena La Nina dan El Nino sebagaimana pola curah hujan dari tahun 1989 sampai 2017 yang ditampilkan pada Gambar 3.2. Data tersebut dipindahkan ke dalam file baru dengan format .csv dengan nama “datahujan_bulanan.csv” agar dapat diolah lebih lanjut pada tahap pengolahan data.

Tabel 3.1 Data Curah Hujan Bulanan tahun 1989 – 2017

	1989	1990	1991	2015	2016	2017
Jan	250	247	509	343	246.1	373.3
Feb	277.4	255	153	230	668.2	238.3
Maret	119	145	141	432	190.8	401.9
...
...
...
Okt	48.5	54	0.3	0	247.7	131.8
Nov	77	62	160.2	165.6	453.6	489.2
Dec	308	89	373.2	446.3	446.3	276.3



Gambar 3.2 Data Curah Hujan dari tahun 1989 sampai 2017

3.3.3 Pengolahan Data

Data yang akan diolah merupakan data deret waktu yang berbentuk matriks 1D dengan ukuran $N \times 1$ yang artinya terdapat banyaknya N baris dan 1 kolom.

Model pembelajaran yang digunakan adalah *supervised learning* dimana *neural network* memerlukan data *input* sebagai masukan model serta data *output* sebagai target untuk melakukan proses pembelajarannya. Data *input* yang digunakan adalah data dari 12 bulan sebelumnya yaitu $t-1, t-2, t-3, \dots, t-11, t-12$ dan Data *output* yang digunakan adalah data dari 12 bulan setelahnya yaitu $t, t+1, t+2, \dots, t+11$. Dalam satu baris akan menghasilkan 12 data sebagai *input* dan 12 data sebagai *output*. Data *input* digunakan sepanjang 12 bulan agar

Model *neural network* dapat mempelajari pola variasi hujan tahunan yang ada di Pulau Jawa khususnya Karangploso dan sekitarnya. Sehingga hasil dari supervised learning ditampilkan pada Tabel 3.2 sebagai berikut :

Tabel 3.2 Data Hasil Supervised Learning

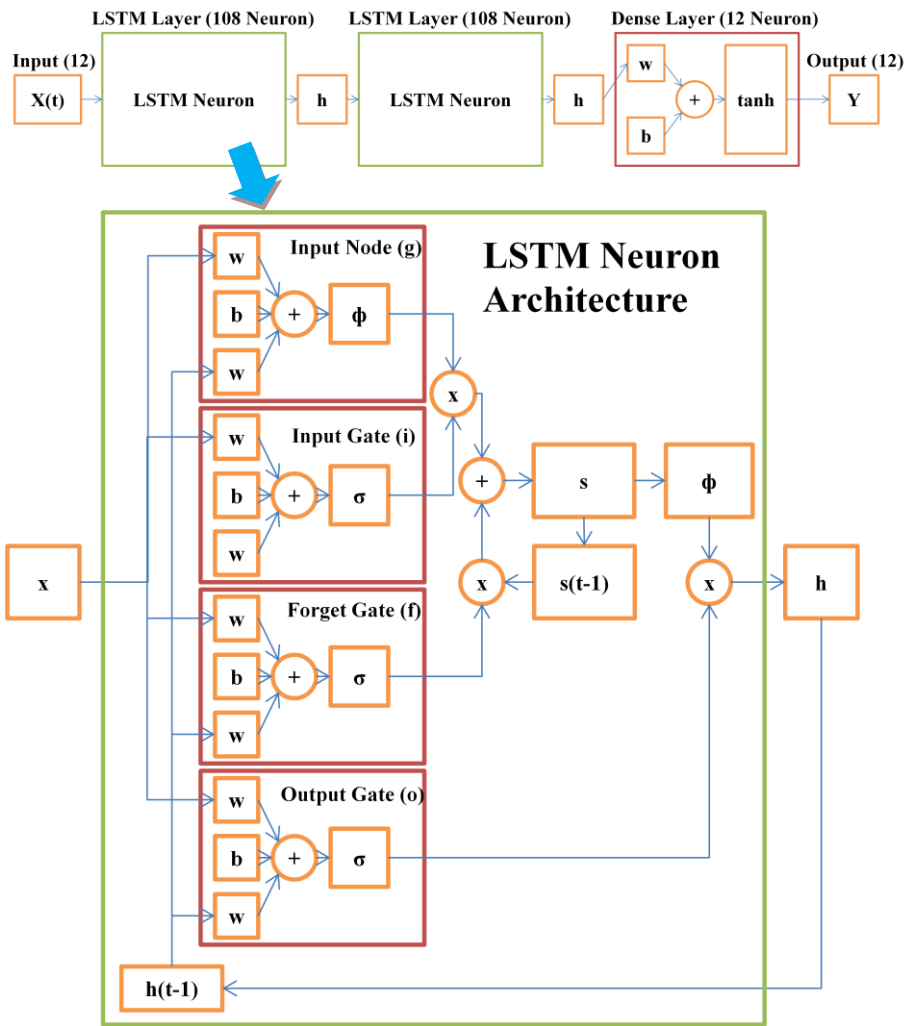
No	Input					Target				
	t-12	t-11	t-10	t-9	t+8	t+9	t+10	t+11
1	250	277	119	143	11.7	54	62	89
2	277	119	143	151	54	62	89	509
3	119	143	151	212	62	89	509	153
...
...
323	165	446	246	668	38.7	0.1	45.9	131.8
324	446	246	668	190	0.1	45.9	131.8	489.2
325	246	668	190	87.1	45.9	131.8	489.2	276.3

Setelah dilakukan pengolahan data, didapatkan data baru berbentuk matriks 2D dengan ukuran Nx24 dimana 12 kolom pertama digunakan sebagai data *input* dan 12 kolom akhir digunakan sebagai data target, kemudian dilakukan normalisasi data dari rentang -1 sampai 1 agar model dapat mengolah data lebih mudah. Data dibagi menjadi 3 kelompok, Data Pelatihan, Validasi dan 1 baris data untuk Pengujian.

3.3.4 Pembuatan Model Neural Network

Model *neural network* yang digunakan adalah *Stacked LSTM* yaitu model *neural network* dengan beberapa lapisan neuron LSTM. Fungsi aktivasi yang digunakan pada setiap neuron adalah *tanh*. Model yang digunakan memiliki 2 *Hidden Layer* yang terdiri 108 Neuron LSTM pada masing-masing *Hidden Layer*. Pada Gambar 3.3, data input diolah terlebih dahulu pada *Hidden Layer* LSTM pertama yang terdiri 108 neuron LSTM kemudian dari *Hidden Layer* pertama diolah ke *Hidden Layer* kedua. Setelah diolah pada *Hidden Layer* LSTM 1 dan LSTM 2, maka hasil olahan data akan didistribusikan menuju *Dense Layer* (*Fully Connected Layer*). *Dense Layer* akan menghasilkan data

output. Penjelasan arsitektur neuron LSTM mengacu pada Persamaan 2.1.



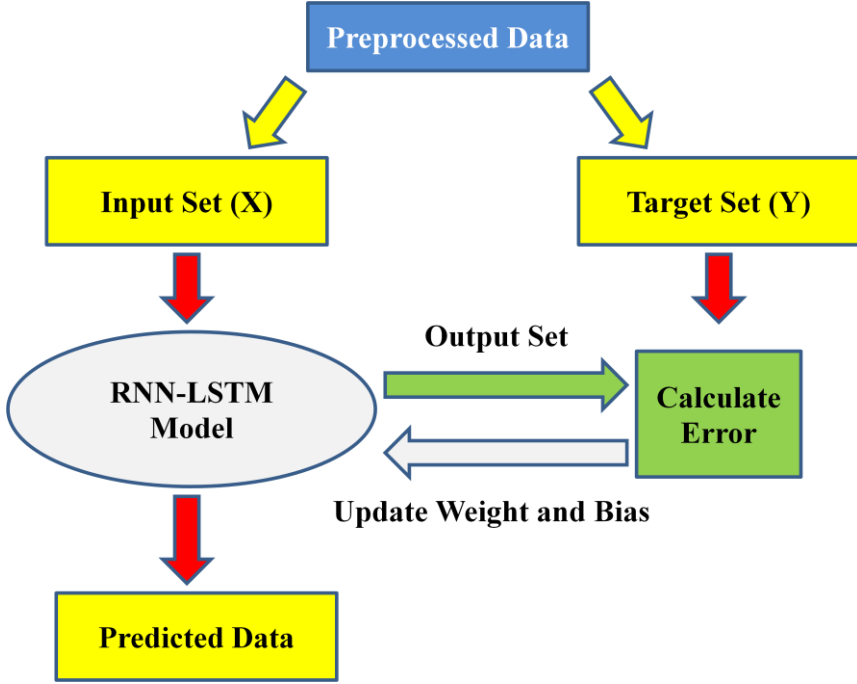
Gambar 3.3 Model Stacked RNN-LSTM

3.3.5 Pelatihan Neural Network

Gambar 3.4 menjelaskan proses pelatihan model RNN-LSTM, data input yang digunakan merupakan sebuah data curah hujan bulanan

24

dengan 12 data yang berurutan dari $t-1$ sampai $t-12$ yang akan diproses ke dalam model RNN tipe Stacked LSTM. Model tersebut akan menghasilkan data output sebanyak 12 urutan data. Data *output* akan kemudian dibandingkan dengan data target yang sudah ditentukan sebanyak 12 data yang berurutan dari t sampai $t+11$. Jika data tidak sesuai maka dengan target maka dilakukan perubahan bobot pada masing-masing neuron yang ada agar *output* yang dihasilkan sesuai dengan target yang dituju. Ketika data *output* tidak sesuai dengan data target maka muncul nilai kesalahan, nilai kesalahan ini dihitung menggunakan *Mean Square Error*. Hasil kesalahan ini digunakan untuk memperbarui nilai bobot yang ada. Diperlukan sebuah algoritma optimasi untuk mengubah dan memperbaiki bobot yang ada. Algoritma optimasi yang digunakan adalah *Adam Optimizer*. Proses ini berlangsung sebanyak iterasi yang diinginkan.



Gambar 3.4 Proses Pelatihan Model

Data yang digunakan selama proses pelatihan dibagi menjadi dua macam yaitu data *Training* dan data *Validation*, 90% data digunakan untuk *training* dan 10% data digunakan untuk *validation*. Tujuannya adalah untuk memastikan bahwa model yang dilatih mendapatkan performa prediksi yang maksimal.

3.3.6 Pengujian Neural Network

Pengujian dilakukan sebanyak empat kali percobaan. Pada percobaan yang pertama hasil keakuratan model RNN-LSTM untuk prediksi tahun 2017 akan dibandingkan dengan *Feedforward Neural Network* (FNN) dan *Autoregressive Integrated Moving Average* (ARIMA). Selanjutnya, Tabel 3.3 menunjukkan pengujian yang kedua dimana model akan diuji untuk memprediksi curah hujan di tahun-tahun yang berbeda dalam durasi waktu satu tahun, dilakukan pengujian pada 10 tahun yang berbeda yaitu di tahun 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, dan 2017.

Tabel 3.3 Pengujian dengan tahun yang berbeda

No	Panjang Tahun	Data Tahun Input	Tahun Prediksi
1	19 tahun	1989-2007	2008
2	20 tahun	1989-2008	2009
3	21 tahun	1989-2009	2010
4	22 tahun	1989-2010	2011
5	23 tahun	1989-2011	2012
6	24 tahun	1989-2012	2013
7	25 tahun	1989-2013	2014
8	26 tahun	1989-2014	2015
9	27 tahun	1989-2015	2016
10	28 tahun	1989-2016	2017

Tabel 3.4 menunjukkan pengujian yang ketiga, model akan diuji untuk memprediksi curah hujan pada tahun yang sama namun dilakukan pengujian menggunakan variasi data *input* yang berbeda-beda, tahun yang akan diuji adalah pada tahun 2017. Terdapat 6 variasi data *input*

yang digunakan yaitu data *input* 5 tahun dari 2012 sampai 2016, data *input* 10 tahun dari 2007 sampai 2016, data *input* 15 tahun dari 2002 sampai 2016, data *input* 20 tahun dari 1997 sampai 2016, data *input* 25 tahun dari 1992 sampai 2016, data *input* 28 tahun dari 1989 sampai 2016.

Tabel 3.4 Pengujian dengan input yang berbeda

No	Panjang Tahun	Data Tahun Input
1	5 Tahun	2012-2016
2	10 Tahun	2007-2016
3	15 Tahun	2002-2016
4	20 Tahun	1997-2016
5	25 Tahun	1992-2016
6	28 Tahun	1989-2016

Tabel 3.5 menunjukkan pengujian yang keempat, model akan diuji untuk memprediksi curah hujan pada durasi waktu yang berbeda beda. Dilakukan prediksi dimulai pada tahun 2013 dengan durasi panjang prediksi selama 1 tahun (2013), 2 tahun (2013-2014), 3 tahun (2013-2015), 4 tahun (2013-2016) dan 5 tahun (2013-2017). Data hasil prediksi tersebut akan dibandingkan dengan data sesungguhnya kemudian dihitung nilai kesalahan totalnya menggunakan *Root Mean Square Error* dimana akan menggambarkan seberapa akurat model tersebut dalam memprediksi curah hujan.

Tabel 3.5 Pengujian dengan tahun output yang berbeda

No	Panjang Tahun	Tahun Prediksi
1	1 Tahun	2013
2	2 Tahun	2013-2014
3	3 Tahun	2013-2015
4	4 Tahun	2013-2016
5	5 Tahun	2013-2017

(Halaman ini sengaja dikosongkan)

BAB IV

HASIL DAN PEMBAHASAN

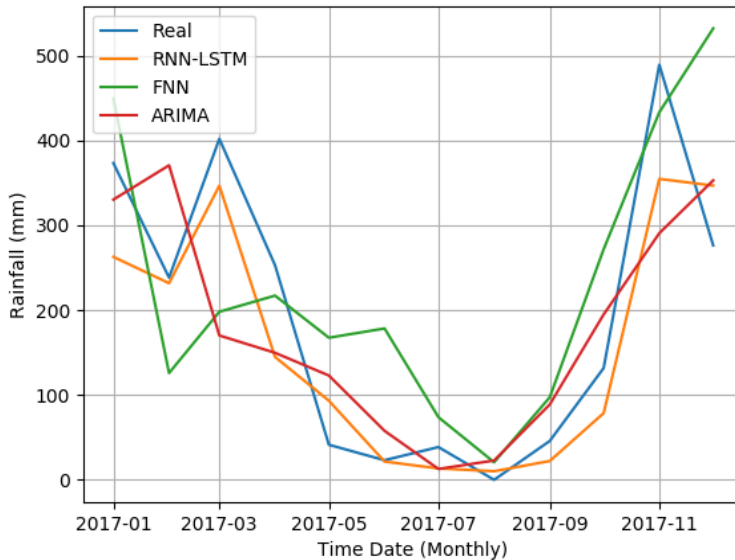
4.1 Perbandingan antara RNN-LSTM, FNN dan ARIMA

Pengujian telah dilakukan menggunakan tiga model yang berbeda yaitu RNN-LSTM, FNN dan ARIMA. Tahun yang diprediksi adalah tahun 2017. Parameter yang diberikan pada tiga model menggunakan parameter yang sama yaitu menggunakan 12 *multistep* data dan orde data yang ke-0. Tabel 4.1 menunjukkan hasil dari ketiga model dimana model RNN-LSTM menghasilkan nilai *Root Mean Square Error* yang terkecil jika dibandingkan dengan model FNN dan ARIMA. Hal ini menjelaskan bahwa model RNN-LSTM memberikan hasil yang lebih akurat dalam memprediksi curah hujan dibandingkan dengan model FNN dan ARIMA.

Tabel 4.1 Perbandingan antara RNN-LSTM, FNN dan ARIMA

No	Model	Root Mean Square Error
1	RNN-LSTM	68,84
2	FNN	127,03
3	ARIMA	109,49

Jika diperhatikan melalui Gambar 4.1, pola curah hujan di tahun 2017 yang dihasilkan oleh model RNN-LSTM hampir sama dengan pola curah hujan yang sesungguhnya di tahun 2017. Pada model FNN, terdapat kesalahan prediksi yang besar pada bulan kedua, ketiga, kelima dan keenam di tahun 2017 dimana model seharusnya menghasilkan curah hujan yang tinggi pada bulan kedua sampai ketiga dan curah rendah pada bulan. Sedangkan pada model ARIMA, terdapat kesalahan prediksi yang besar pada bulan ketiga dan kesebelas dimana model seharusnya menghasilkan curah hujan yang tinggi pada bulan tersebut. Sehingga dari perbandingan antara RNN-LSTM, FNN dan ARIMA, model RNN-LSTM menghasilkan prediksi yang paling baik.



Gambar 4.1 Perbandingan antara RNN-LSTM, FNN dan ARIMA

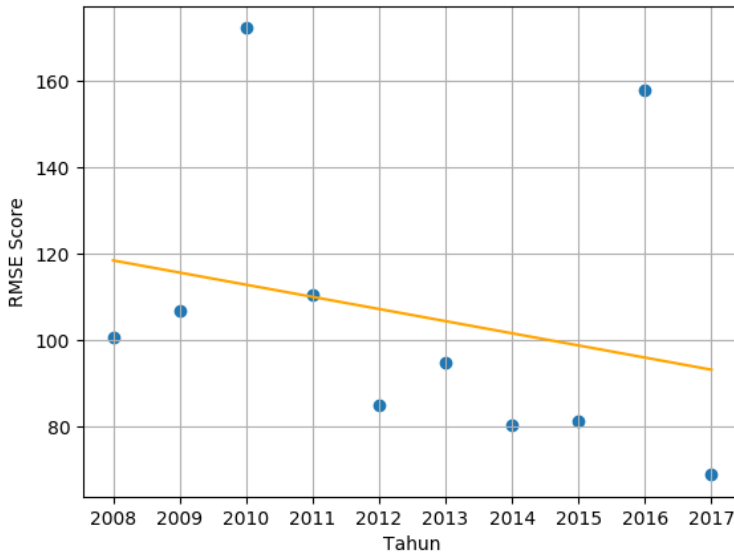
4.2 Prediksi dengan tahun yang berbeda

Model RNN-LSTM telah diuji untuk memprediksi curah hujan selama 1 tahun pada 10 tahun yang berbeda yaitu 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016 dan 2017. Data *input* yang digunakan adalah mulai pada tahun 1989 sampai tahun akhir prediksi. Misal, dilakukan prediksi curah hujan untuk tahun 2013 sehingga data yang digunakan untuk memprediksi tahun 2013 adalah data bulanan dari tahun 1989 sampai 2012. Data di tahun 2013 tidak digunakan sebagai data *input*. Data tersebut digunakan untuk membandingkan data hasil prediksi yang diperoleh dengan data yang sesungguhnya. Data yang telah dibandingkan tersebut akan menghasilkan nilai *Root Mean Square Error* yang digunakan untuk menilai seberapa besar kesalahan dari hasil prediksi. Semakin besar nilai kesalahan yang diperoleh membuktikan bahwa semakin besar kesalahan prediksi terhadap data yang sesungguhnya.

Tabel 4.2 Prediksi dengan tahun yang berbeda

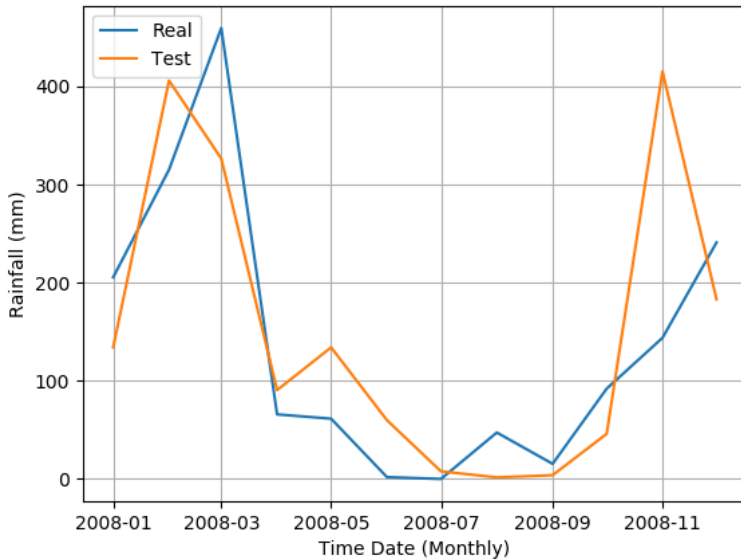
No	Panjang Tahun	Data Tahun Input	Tahun Prediksi	Root Mean Square Error
1	19 tahun	1989-2007	2008	100,73
2	20 tahun	1989-2008	2009	106,65
3	21 tahun	1989-2009	2010	172,22
4	22 tahun	1989-2010	2011	110,52
5	23 tahun	1989-2011	2012	84,80
6	24 tahun	1989-2012	2013	94,77
7	25 tahun	1989-2013	2014	80,41
8	26 tahun	1989-2014	2015	81,16
9	27 tahun	1989-2015	2016	157,94
10	28 tahun	1989-2016	2017	68,84

Tabel 4.2 menunjukkan nilai *Root Mean Square Error* pada tahun 2008 sampai 2017. Tahun prediksi 2008 menggunakan data *input* yang paling sedikit yaitu 1989-2007, kemudian semakin bertambah tahun yang diprediksi maka data *input* yang digunakan semakin banyak sampai pada tahun prediksi 2017 yang menggunakan data *input* paling banyak yaitu 1989-2017.



Gambar 4.2 Grafik tren prediksi dengan tahun yang berbeda

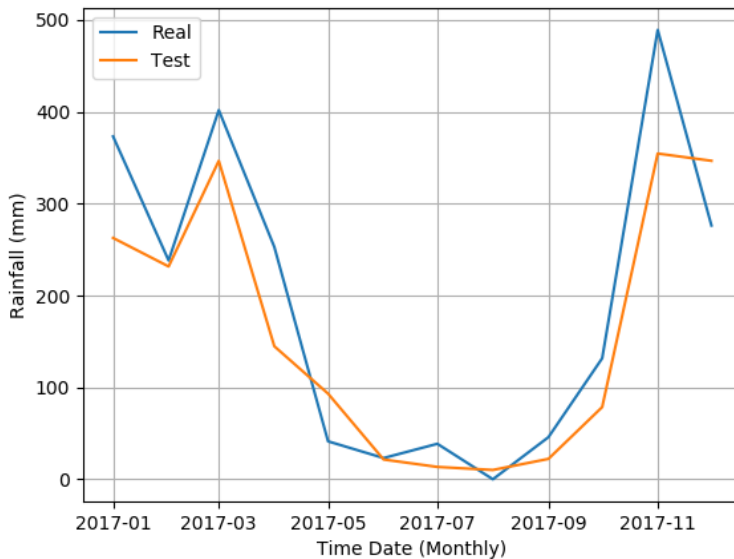
Jika dilihat pada Gambar 4.2, nilai tren RMSE dari tahun 2008 sampai 2017 mengalami penurunan secara bertahap dengan diiringi semakin banyaknya data *input* yang digunakan. Hal ini membuktikan bahwa semakin panjang data *input* yang digunakan maka menghasilkan nilai RMSE yang semakin kecil sehingga semakin panjang data input dapat menghasilkan prediksi yang semakin akurat.



Gambar 4.3 Grafik Prediksi 2008

Gambar 4.3 adalah grafik yang menunjukkan data prediksi curah hujan dengan data curah hujan bulanan yang sesungguhnya pada tahun 2008. Data *input* yang digunakan adalah dari tahun 1989 sampai 2007. Terlihat bahwa prediksi kurang dapat memberikan hasil yang tepat terhadap data yang sesungguhnya dengan nilai RMSE adalah 100,73.

Gambar 4.4 adalah grafik yang menunjukkan data prediksi curah hujan dengan data curah hujan sesungguhnya pada tahun 2017. Data *input* yang digunakan adalah dari tahun 1989-2016. Hasil menunjukkan bahwa model RNN dapat memprediksi curah hujan di tahun selanjutnya dengan lebih baik jika data *input* yang digunakan semakin panjang. Nilai RMSE yang diperoleh pada tahun prediksi 2017 adalah sebesar 68,84.



Gambar 4.4 Grafik Prediksi 2017

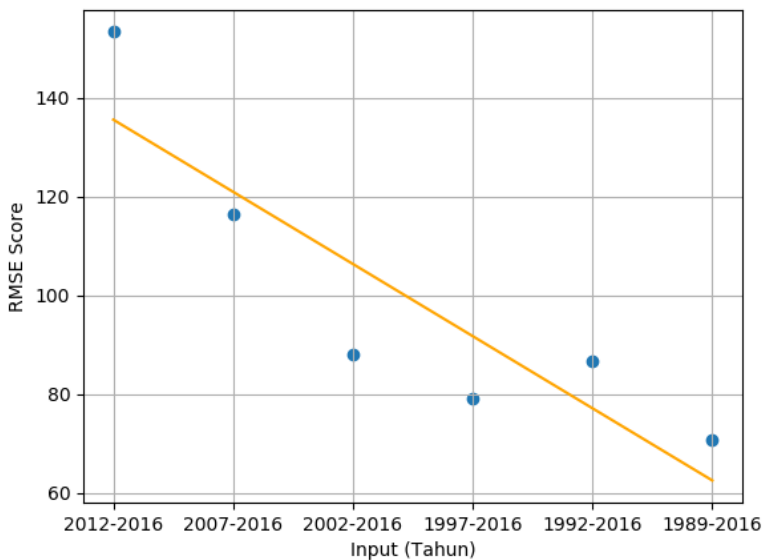
Gambar 4.3 dan 4.4 merupakan contoh hasil grafik data prediksi bulanan pada tahun 2008 dan 2017. Pada kedua grafik tersebut menunjukkan bahwa prediksi 2017 lebih mendekati data yang sesungguhnya daripada prediksi 2008. Hal ini disebabkan karena data *input* yang digunakan untuk prediksi 2017 lebih banyak daripada prediksi 2008.

4.3 Prediksi dengan tahun input yang berbeda

Model RNN-LSTM telah diuji untuk memprediksi curah hujan pada tahun 2017 dengan variasi data *input*. Terdapat 6 variasi data *input* yang digunakan yaitu data *input* 5 tahun dari 2012 sampai 2016, data *input* 10 tahun dari 2007 sampai 2016, data *input* 15 tahun dari 2002 sampai 2016, data *input* 20 tahun dari 1997 sampai 2016, data *input* 25 tahun dari 1992 sampai 2016, data *input* 28 tahun dari 1989 sampai 2016.

Tabel 4.3 Prediksi dengan input yang berbeda

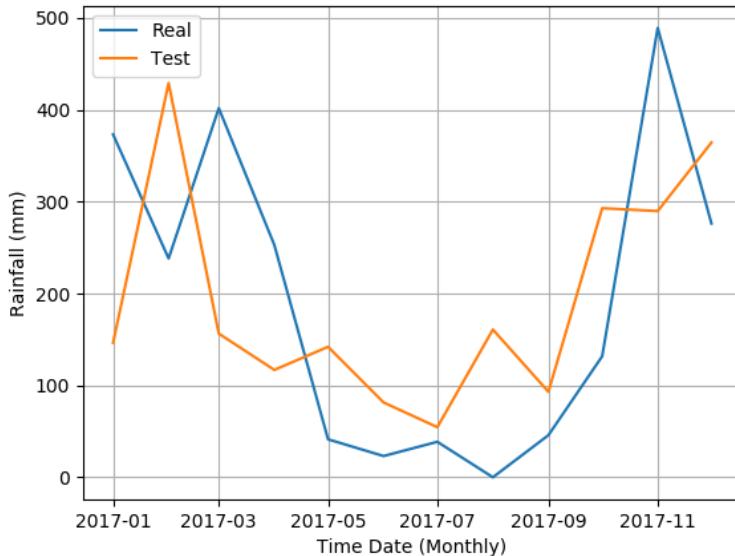
No	Panjang Tahun	Data Tahun Input	Root Mean Square Error
1	5 Tahun	2012-2016	153,29
2	10 Tahun	2007-2016	116,42
3	15 Tahun	2002-2016	87,96
4	20 Tahun	1997-2016	79,08
5	25 Tahun	1992-2016	86,8
6	28 Tahun	1989-2016	70,62



Gambar 4.5 Grafik tren prediksi dengan tahun input yang berbeda

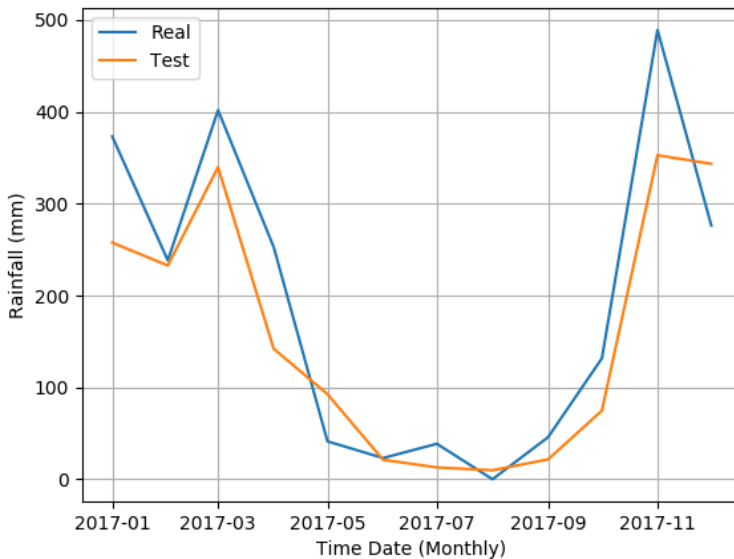
Berdasarkan Gambar 4.5, semakin panjang data *input* yang digunakan maka semakin akurat hasil prediksi yang diperoleh dengan dibuktikan oleh nilai RMSE yang semakin kecil serta tren data yang semakin menurun. Untuk lebih lengkapnya, nilai RMSE dapat dilihat pada Tabel 4.3. Ketika data *input* yang digunakan dari tahun 2012 sampai 2016 menghasilkan nilai RMSE yang terbesar yaitu 153,29. Kemudian semakin panjang data *input* yang digunakan, model RNN

menghasilkan nilai RMSE yang semakin kecil. Pada data *input* yang paling panjang yaitu dari tahun 1989 sampai 2016 menghasilkan nilai RMSE yang terkecil yaitu 70,62.



Gambar 4.6 Grafik prediksi 2017 dengan input 5 tahun

Gambar 4.6 dan Gambar 4.7 menunjukkan prediksi curah hujan pada tahun yang sama yaitu 2017. Terlihat bahwa ketika data *input* yang digunakan adalah selama 5 tahun maka prediksi curah hujan tidak dapat memberikan hasil yang memuaskan dan ketika data *input* yang digunakan adalah selama 28 tahun maka prediksi curah hujan semakin mendekati dengan data yang sesungguhnya.



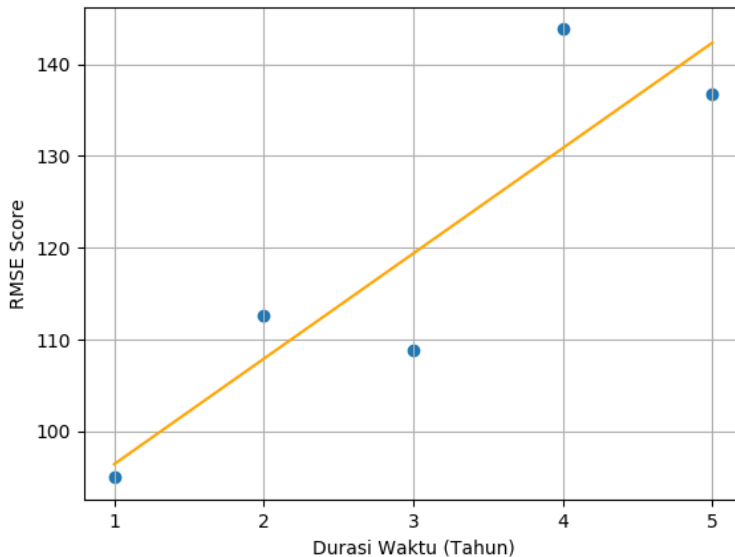
Gambar 4.7 Grafik prediksi 2017 dengan input 28 tahun

4.4 Prediksi dengan tahun output yang berbeda

Model RNN-LSTM telah diuji untuk menghasilkan prediksi dalam jangka waktu yang berbeda. Dilakukan prediksi mulai pada tahun 2012 untuk menghasilkan prediksi selama 1 tahun (2013), 2 tahun (2013-2014), 3 tahun (2013-2015), 4 tahun (2013-2016) dan 5 tahun (2013-2017) sebagaimana yang ditunjukkan oleh Tabel 4.4.

Tabel 4.4 Prediksi dengan tahun output yang berbeda

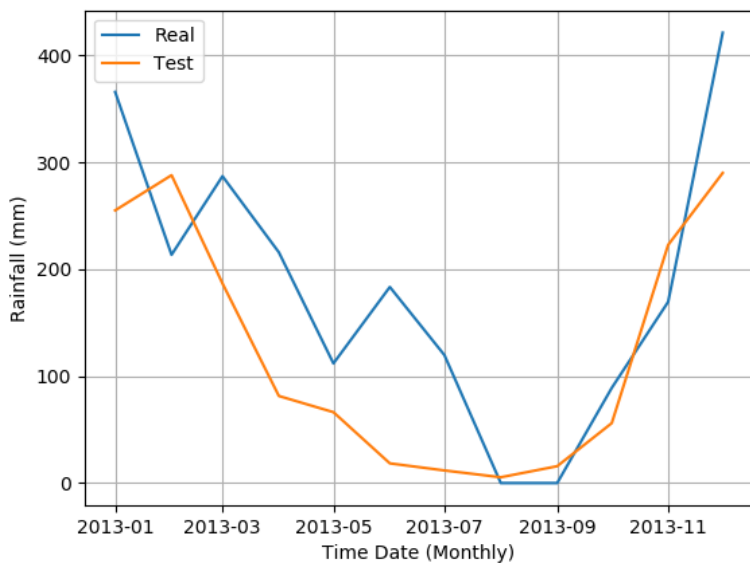
No	Panjang Tahun	Tahun Prediksi	Root Mean Square Error
1	1 Tahun	2013	94,98
2	2 Tahun	2013-2014	112,56
3	3 Tahun	2013-2015	108,81
4	4 Tahun	2013-2016	143,79
5	5 Tahun	2013-2017	136,75



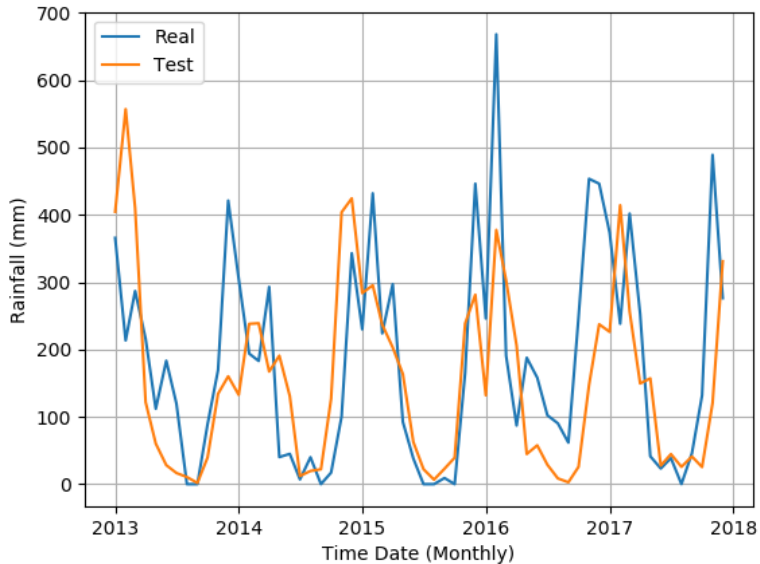
Gambar 4.8 Grafik tren prediksi dengan tahun output yang berbeda

Berdasarkan Gambar 4.8, semakin panjang tahun yang diprediksi maka semakin tidak akurat hasil prediksi yang diperoleh karena tren data yang ditunjukkan semakin naik, hal ini juga dapat dibuktikan oleh nilai RMSE yang semakin membesar pada Tabel 4.3 ketika jangka waktu prediksi semakin panjang. Pada prediksi dalam jangka waktu 1 tahun, nilai RMSE yang didapatkan adalah 94,98. Pada prediksi selama 5 tahun, model RNN menghasilkan nilai RMSE yang terbesar yaitu 136,75. Sehingga model RNN hanya mampu memprediksi curah hujan yang terbaik dalam jangka waktu 1 tahun saja.

Gambar 4.9 dan Gambar 4.10 menunjukkan bahwa semakin panjang tahun yang diprediksi maka kemampuan model dalam memprediksi akan semakin berkurang. Gambar 4.10 menghasilkan prediksi yang kurang baik ketika memprediksi tahun 2016 sampai 2017.



Gambar 4.9 Grafik prediksi selama 1 tahun



Gambar 4.10 Grafik prediksi selama 5 tahun

Secara umum, model RNN menghasilkan prediksi pola curah hujan yang sama dengan pola curah hujan regional A seperti yang ditampilkan pada Gambar 2.2. Daerah Karangploso, Malang termasuk ke dalam kategori curah hujan regional A. Regional A mencakup daerah Indonesia bagian selatan mulai dari Sumatra Selatan sampai Pulau Timor, Kalimantan bagian Selatan, Sulawesi dan Irian Jaya. Curah hujan pada Regional A memiliki satu puncak dari November sampai Maret dan satu lembah dari Mei sampai September (Aldrian, 2003). Untuk lebih lengkapnya, hasil plot grafik dari semua percobaan prediksi dengan tahun yang berbeda, data *input* yang berbeda dan panjang tahun *output* yang berbeda dapat dilihat pada LAMPIRAN A.

BAB V

PENUTUP

5.1 Kesimpulan

Telah dilakukan penelitian untuk mengetahui kemampuan model RNN-LSTM yang kemudian dapat disimpulkan bahwa:

1. Model RNN-LSTM yang dibuat dengan arsitektur 2 *Hidden Layer* yang terdiri dari 108 neuron LSTM pada masing-masing *Hidden Layer*-nya telah mampu memprediksi pola curah hujan bulanan di daerah Karangploso lebih baik jika dibandingkan dengan model FNN dan ARIMA.
2. Prediksi yang didapatkan dari tahun ke tahun semakin baik yang disebabkan oleh data *input* model semakin panjang dengan ditandai nilai RMSE yang semakin mengecil.
3. Jika data *input* model yang digunakan semakin panjang maka model dapat memprediksi curah hujan semakin baik yang ditandai oleh nilai RMSE yang semakin kecil.
4. Model kurang mampu memprediksi data *output* yang semakin panjang seperti prediksi 2 tahun atau 5 tahun mendatang. Nilai RMSE yang semakin besar membuktikan bahwa hasil prediksi semakin tidak sesuai dengan hasil yang sesungguhnya.

5.2 Saran

Pada penelitian selanjutnya, dapat dilakukan dengan beberapa saran sebagai berikut:

1. Menambahkan data *input* tahun yang lebih panjang.
2. Menambahkan data *input* berupa data selisih dari bulan sebelumnya sampai bulan setelahnya pada data curah hujan tersebut.
3. Menambahkan data *input* faktor-faktor penyebab hujan (seperti Temperatur Permukaan Laut, Kelembaban Udara, Angin Muson).

(Halaman ini sengaja dikosongkan)

DAFTAR PUSTAKA

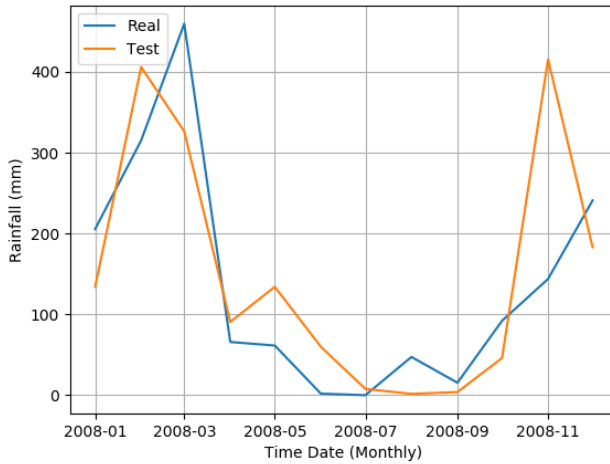
- Abadi, Martin, et al. 2017. *A Computational Model for TensorFlow (An Introduction)*. Available at : <http://dl.acm.org/citation.cfm?doid=3088525.3088527>
- Aldrian, Edwin. And Susanto, R. D. 2003. Identification of Three Dominant Rainfall Regions within Indonesia and Their Relationship to Sea Surface Temperature. *International Journal of Climatology*, 23, 1435–1452.
- Brownlee, Jason. 2016. *Introduction to Python Deep Learning with Keras*. Available at : <https://machinelearningmastery.com/introduction-python-deep-learning-library-keras/> (Accessed: 11 October 2018)
- Brownlee, Jason. 2017. *Long Short-Term Memory Networks with Python*. Available at : <https://machinelearningmastery.com/>
- Chai, T., et al. 2014. Root mean square error (RMSE) or mean absolute error (MAE) – Arguments against avoiding RMSE in the literature. *Journal of Geoscientific Model Development*, 7, 1247–1250.
- Charaniya, N. A. and Dudul, S.V. 2013. Design of Neural Network Models for Daily Rainfall Prediction. *International Journal of Computer Applications*, 61, 23–26.
- Fuhrer, C., Solem, J. E., Verdier, O. 2016. *Scientific Computing with Python*. Packt Publishing, Birmingham
- Ibnu Katsir. 2004. *Tafsir Ibnu Katsir*. Pustaka Imam Asy-Syafi'i. Bogor
- Irawan, Bambang. 2006. Fenomena Anomali Iklim El Nino dan La Nina Kecenderungan Jangka Panjang dan Pengaruhnya terhadap Produksi Pangan. *Forum Penelitian Agro Ekonomi*, 24, 28–45.

- Lipton, Zachary C., et al. 2015. 'A Critical Review of Recurrent Neural Networks for Sequence Learning'. Available at : arXiv:1506.00019v4 [cs.LG] 17 Oct 2015
- Mislan, H., Hardwinarto, S., Sumaryono, Aipassa, M. 2015. Rainfall Monthly Prediction Based on Artificial Neural Network: A Case Study in Tenggarong Station, East Kalimantan – Indonesia. *Procedia Computer Science*, 59,142–151.
- Morgan, Peter. 2018. *Machine Learning is Changing the Rules*. O'Reilly Media, Inc. California
- O'Reilly Media. 2017. *Artificial Intelligence Now*. O'Reilly Media, Inc. California
- Suyanto. 2014. *Artificial Intelligence*. INFORMATIKA. Bandung
- Tjasyono, Bayong. 2004. *Klimatologi Edisi 2*. Institut Teknologi Bandung. Bandung
- Wang, H. R., et al. 2014. An improved ARIMA model for precipitation simulations. *Nonlin. Processes Geophys.*, 21, 1159–1168.
- Wirjohamidjojo, S. & Swarinoto, Y. 2010. *Iklim Kawasan Indonesia (Dari Aspek Dinamik – Sinoptik)*. Badan Meteorologi Klimatologi dan Geofisika. Jakarta
- Zaccone, Giancarlo. 2016. *Getting Started with TensorFlow*. Packt Publishing. Birmingham
- Zaytar, M. A. & Amrani, C. E. 2016. Sequence to Sequence Weather Forecasting with Long Short-Term Memory Recurrent Neural Networks. *International Journal of Computer Applications*, 143, 07–11.

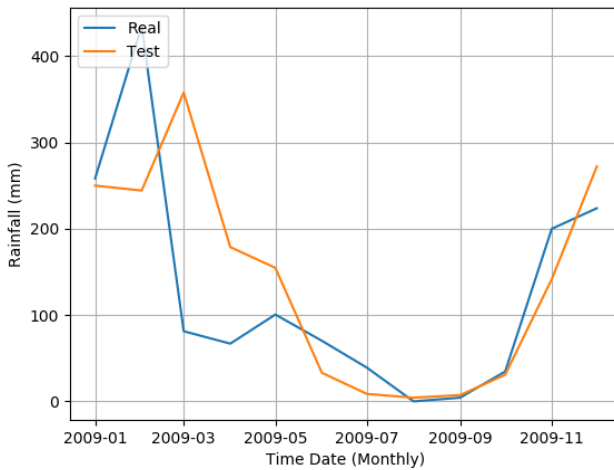
LAMPIRAN A

DATA HASIL PENELITIAN

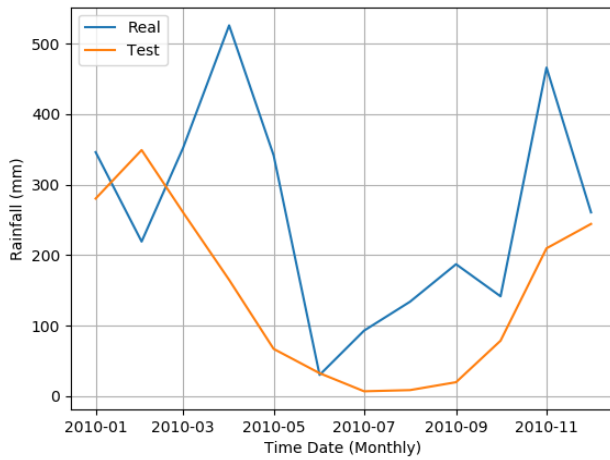
A.1 Prediksi dengan tahun yang berbeda



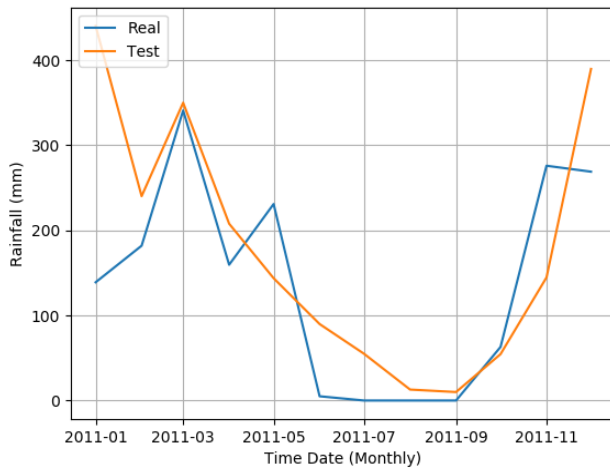
Gambar A.1.1 Prediksi 2008



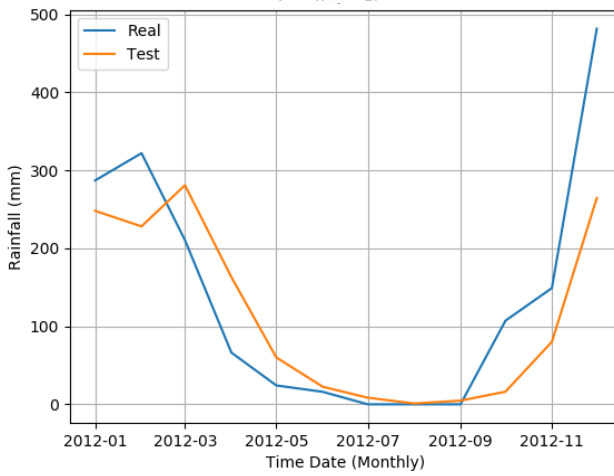
Gambar A.1.2 Prediksi 2009



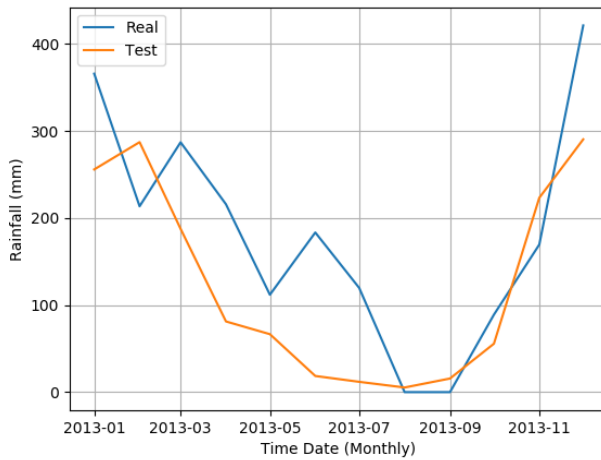
Gambar A.1.3 Prediksi 2010



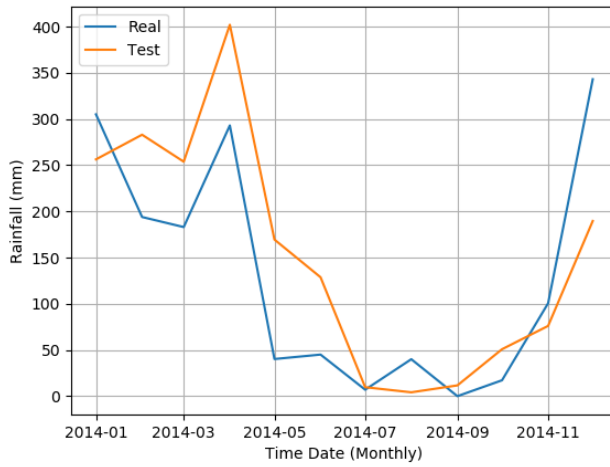
Gambar A.1.4 Prediksi 2011



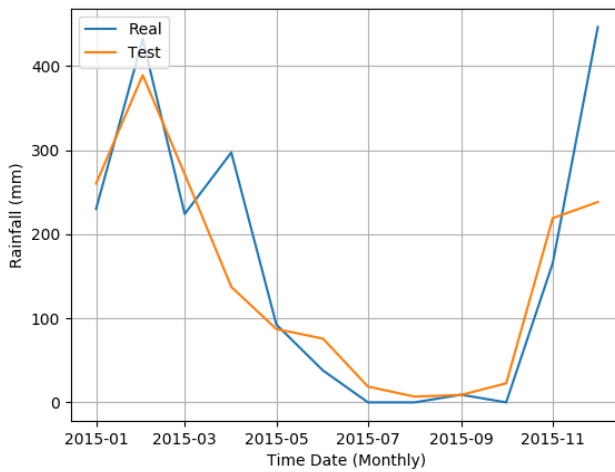
Gambar A.1.5 Prediksi 2012



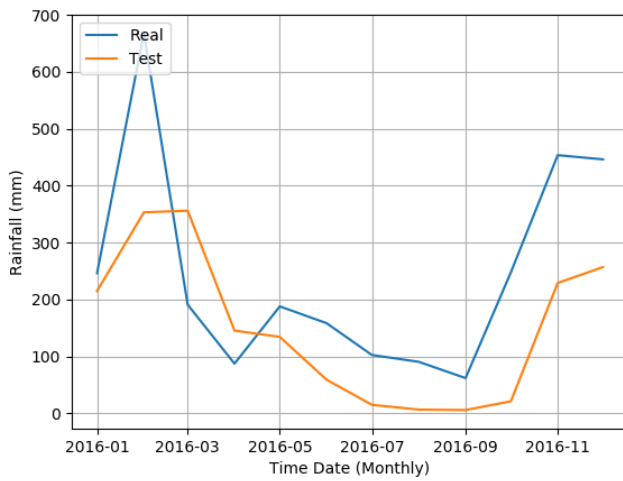
Gambar A.1.6 Prediksi 2013



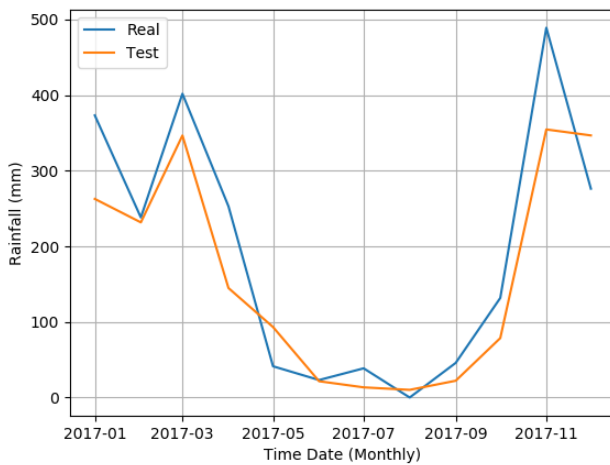
Gambar A.1.7 Prediksi 2014



Gambar A.1.8 Prediksi 2015

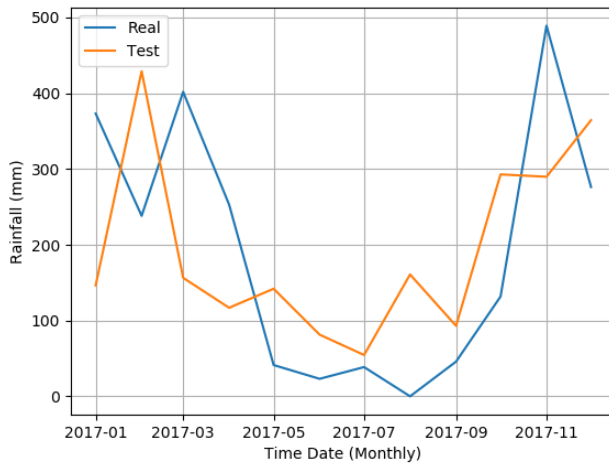


Gambar A.1.9 Prediksi 2016

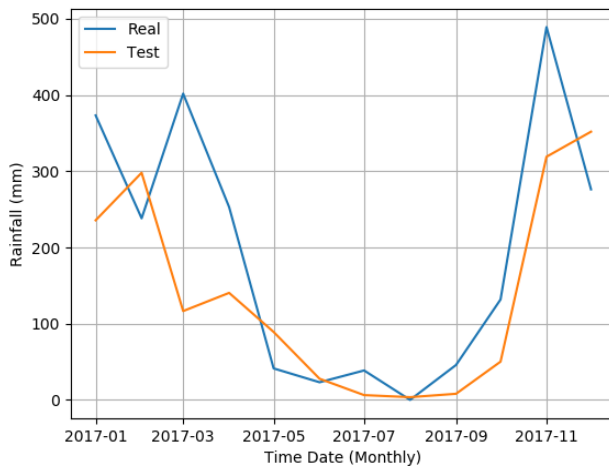


Gambar A.1.10 Prediksi 2017

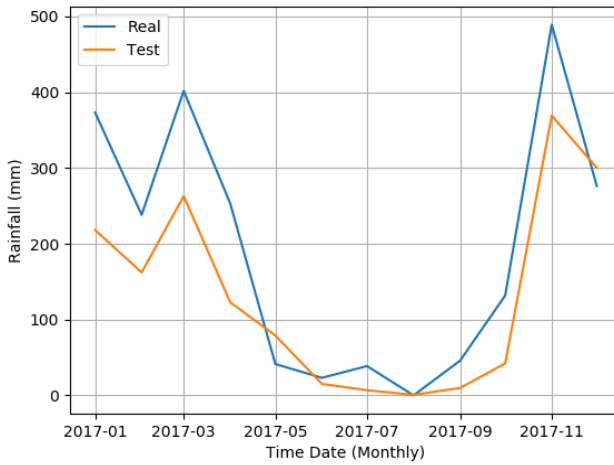
A.2 Prediksi dengan input yang berbeda



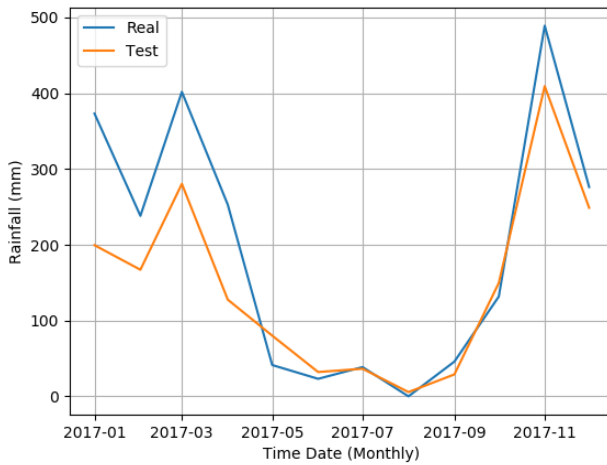
Gambar A.2.1 Prediksi 2017 dengan input 5 tahun



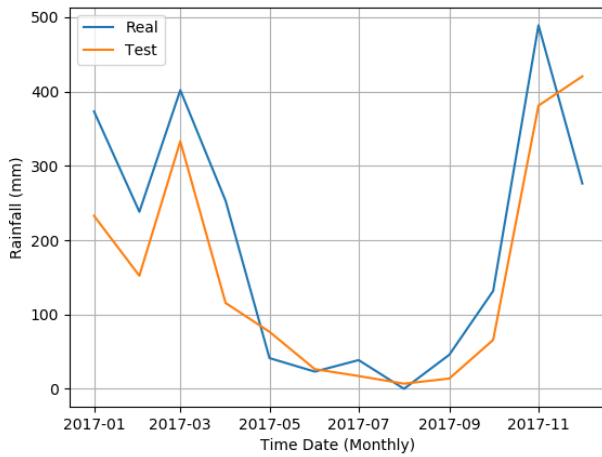
Gambar A.2.2 Prediksi 2017 dengan input 10 tahun



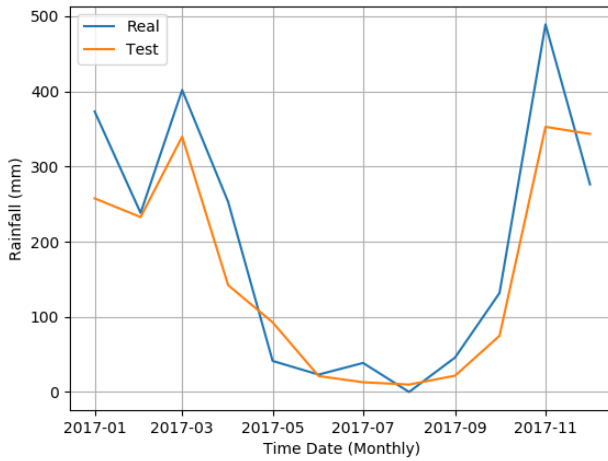
Gambar A.2.3 Prediksi 2017 dengan 15 tahun



Gambar A.2.4 Prediksi 2017 dengan 20 tahun

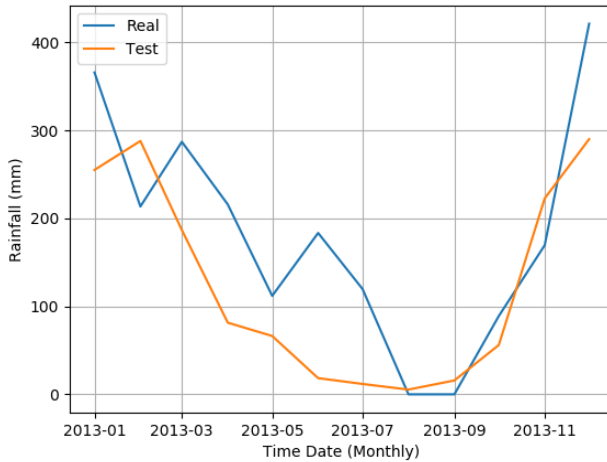


Gambar A.2.5 Prediksi 2017 dengan 25 tahun

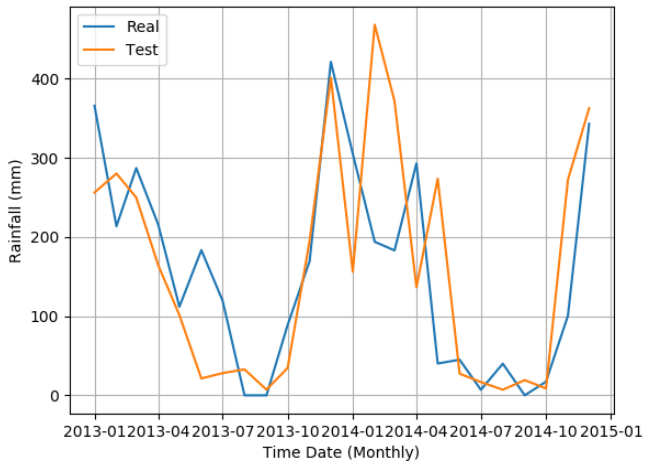


Gambar A.2.6 Prediksi 2017 dengan 28 tahun

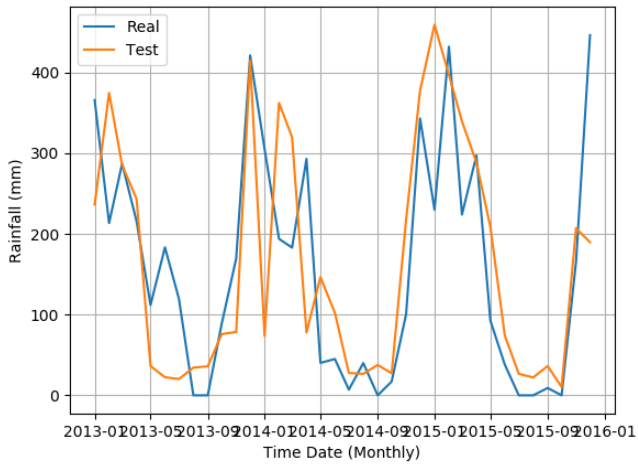
A.3 Prediksi dengan durasi tahun output yang berbeda



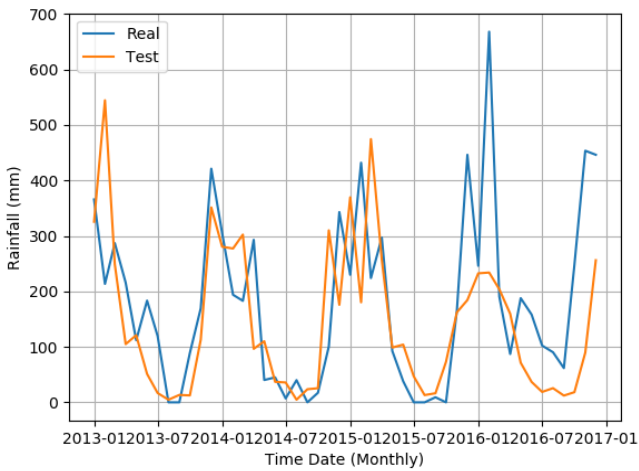
Gambar A.3.1 Prediksi dengan output selama 1 tahun



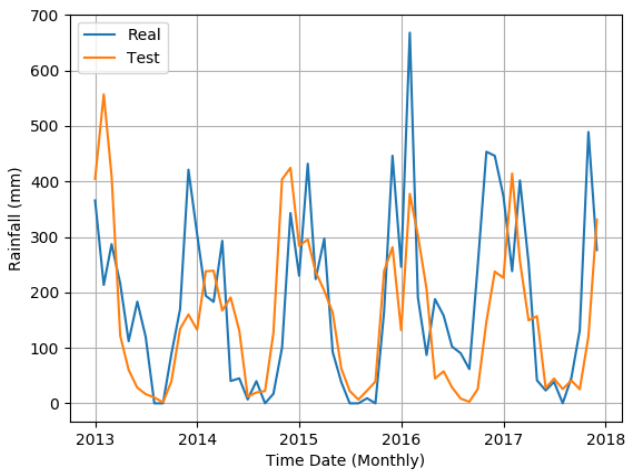
Gambar A.3.2 Prediksi dengan output selama 2 tahun



Gambar A.3.3 Prediksi dengan output selama 3 tahun



Gambar A.3.4 Prediksi dengan output selama 4 tahun



Gambar A.3.5 Prediksi dengan output selama 5 tahun

LAMPIRAN B

KODE PROGRAM

```
from pandas import DataFrame
from pandas import concat
from pandas import read_csv
from pandas import datetime
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
from keras.models import Sequential
from keras.layers import LSTM
from keras.layers import Dense
import matplotlib.pyplot as plt
import numpy as np
import math
import time
import csv
import os

#define function
#=====
=====
def series_to_supervised(data, n_in=1, n_out=1, dropnan=True):
    n_vars = 1 if type(data) is list else data.shape[1]
    df = DataFrame(data)
    cols, names = list(), list()
    # input sequence (t-n, ... t-1)
    for i in range(n_in, 0, -1):
        cols.append(df.shift(i))
        names += [('var%d(t-%d)' % (j+1, i)) for j in
range(n_vars)]
    # forecast sequence (t, t+1, ... t+n)
    for i in range(0, n_out):
        cols.append(df.shift(-i))
        if i == 0:
            names += [('var%d(t)' % (j+1)) for j
in range(n_vars)]
        else:
            names += [('var%d(t+%d)' % (j+1, i))
for j in range(n_vars)]
    # put it all together
    agg = concat(cols, axis=1)
    agg.columns = names
```

```

# drop rows with NaN values
if dropnan:
    agg.dropna(inplace=True)
return agg

def forecast_lstm(model, X, n_batch):
    # reshape input pattern to [samples, timesteps, features]
    X = X.reshape(1, 1, len(X))
    # make forecast
    forecast = model.predict(X, batch_size=n_batch)
    # convert to array
    return [x for x in forecast[0, :]]

def make_forecasts(model, n_batch, data, n_lag):
    forecasts = list()
    for i in range(len(data)):
        X, y = data[i, 0:n_lag], data[i, n_lag:]
        # make forecast
        forecast = forecast_lstm(model, X, n_batch)
        # store the forecast
        forecasts.append(forecast)
    return forecasts

def predict_next_values(model, n_batch, data, n_lag, scaler, date,
folder6, output_filename):
    #predict next values
    print("")
    print("="*50)
    print("Prediction")
    print("="*50)
    data = scaler.fit_transform(data)
    forecasts = list()
    for i in range(len(data)):
        X = data[i, 0:n_lag]
        forecast = forecast_lstm(model, X, n_batch)
        forecasts.append(forecast)

    prediction = scaler.inverse_transform(forecasts)
    last_year = date[-1].astype('datetime64[Y]')
    np.savetxt('../Output/'+folder6+'/Rainfall Prediction
Data for '+str(int(str(last_year))+1)+'.csv',prediction[0,:],
delimiter=",", header='Prediction')
    print(prediction)
    print("")

```



```

fig = plt.figure(figsize=(12.8, 9.6))
fig.canvas.set_window_title(output_filename)
date_ = np.array(date[-1], dtype='datetime64') +
np.arange(n_lag+1)
plt.plot(date_[1:],prediction[0,:], color='orange',
marker='o')
plt.plot(date_[1:], prediction[0,:], color='blue')
plt.title('Rainfall Prediction Data for
'+str(int(str(last_year))+1))
plt.xlabel('Time Date (Monthly)')
plt.ylabel('Rainfall (mm)')
plt.grid()
fig.savefig("../Output/"+folder6+"/Rainfall Prediction
Data for "+str(int(str(last_year))+1)+".png")

def newpath(path):
    #make output directory
    newpath = r'../Output/'+path
    if not os.path.exists(newpath):
        os.makedirs(newpath)

def write_files(output_filename, dataset, n_lag, n_seq, n_epochs,
n_neurons, n_hidden_layers,
val_data, activation, loss,
optimizer, trainScore,
testScore, minute, testY,
testPredict, date):
    #save file
    file = open('../Output/log/LSTM-Log-
'+output_filename+'.txt','w')
    file.write('=====\n')
    file.write('Neural Network Records\n')
    file.write('=====\n')
    file.write('\nINPUT')
    file.write('\nLength of Data \t\t: %d' %len(dataset))
    file.write('\nInput Sequences Data \t: %d' %n_lag)
    file.write('\nOutput Sequences Data \t: %d' %n_seq)
    file.write('\nTraining Epochs \t: %d' %n_epochs)
    file.write('\nThe Number of Neurons \t: %d' %n_neurons)
    file.write('\nHidden Layers \t\t: %s'
%(n_hidden_layers+2))
    file.write('\nTraining Data \t\t: %.2f %%' %(1-
val_data)*100))

```

```

        file.write('\nValidation Data \t: %.2f %%'
%(val_data*100))
        file.write('\nActivation \t\t: %s' %activation)
        file.write('\nLoss \t\t\t: %s' %loss)
        file.write('\nOptimizer \t\t: %s' %optimizer)
        file.write('\n\nOUTPUT')
        file.write('\nTrain RMSE \t\t: %.4f' %trainScore)
        file.write('\nTest RMSE \t\t: %.4f' %testScore)
        file.write('\nExecution Time \t\t: %.5s minutes' %minute)
        file.close()

#save to csv
testY_ = testY[-1,:].reshape(n_seq,1)
testPredict_ = testPredict[-1,:].reshape(n_seq,1)
save_array = np.concatenate((testY_,testPredict_),axis=1)
last_year = date[-1].astype('datetime64[Y]')
np.savetxt(' ../Output/csv/Rainfall Prediction Data for
'+str(last_year)+'-'+output_filename+'.csv',save_array,
delimiter=",",header='testY, testPredict')

#save in one csv file
#input
a = [n_epochs]
b = [n_neurons]
c = [n_hidden_layers+2]
d = [val_data*100]

#output
e = [trainScore]
f = [testScore]
g = [minute]

save_array_data = np.concatenate((a,b,c,d,e,f,g), axis=0)
with open(' ../LogData.csv', 'a') as csvFile:
    writer = csv.writer(csvFile)
    writer.writerow(save_array_data)

def plots(output_filename, loss_history, val_loss_history,
folder3, multistep_date,
            multistep_data, trainY, trainPredict,
folder4, n_lag,
            testY, testPredict, folder5):
    #make plots
    fig1 = plt.figure()

```

```

fig1.canvas.set_window_title(output_filename)
plt.plot(loss_history, label='Train')
plt.plot(val_loss_history, label='Validation')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.title('Plot of Train and Validation Loss')
plt.legend(loc='upper right')
plt.grid()
fig1.savefig("../Output/"+folder3+"/Plot of Train and
Validation Loss-"+output_filename+".png")

fig2 = plt.figure()
fig2.canvas.set_window_title(output_filename)
plt.plot(multistep_date[:-1,-1],trainY[:-1],
label='Real')
plt.plot(multistep_date[:-1,-1],trainPredict[:,-1],
label='Train')
plt.xlabel('Time Date (Monthly)')
plt.ylabel('Rainfall (mm)')
plt.title('Training Results')
plt.legend(loc='upper left')
plt.grid()
fig2.savefig("../Output/"+folder4+"/Training Results-
"+output_filename+".png")

fig3 = plt.figure()
fig3.canvas.set_window_title(output_filename)
plt.plot(multistep_date[-1,n_lag:], testY[-1,:],
label='Real')
plt.plot(multistep_date[-1,n_lag:], testPredict[-1,:],
label='Test')
plt.xlabel('Time Date (Monthly)')
plt.ylabel('Rainfall (mm)')
plt.title('Testing Results')
plt.legend(loc='upper left')
plt.grid()
fig3.savefig("../Output/"+folder5+"/Testing Results-
"+output_filename+".png")

"""
fig4 = plt.figure(figsize=(12.8, 9.6))
fig4.canvas.set_window_title(output_filename)
plt.plot(date,dataset)
last_year = date[-1].astype('datetime64[Y]')

```

```

plt.title("Rainfall Data from "+str(early_year)+" to
"+str(last_year))
plt.xlabel('Time Date (Monthly)')
plt.ylabel('Rainfall (mm)')
plt.grid()
fig4.savefig("../Output/Rainfall Data from
"+str(early_year)+" to "+str(last_year)+"-
"+output_filename+".png")
"""
plt.close('all')

#=====
=====

def main(a, b, c, d):
    #BMKG Machine Forecaster
    print("")
    print("="*50)
    print("RAINFALL FORECASTER")
    print("")
    print("Author : Muhammad Febriantoro")
    print("Computational and Modeling Physics")
    print("Faculty of Mathematics and Natural Sciences")
    print("University of Brawijaya")
    print("More info :")
    print("+62 85 771 883 558")
    print("muhammadfebriantoro@gmail.com")
    print("="*50)
    print(" ")

    #show all files in data directory
    path = '../Data'
    files = os.listdir(path)
    print("Data yang tersedia")
    for name in files:
        print(name)

    #initialize random number
    np.random.seed(10)

    #input
    print(" ")
    data = 'curahhujan_bulanan2017.csv' #input("Masukkan data
yang ingin diolah\t : ") #'curahhujan_bulanan.csv'

```

```

        early_year = 1989 #int(input("Data Tahun Awal\t\t\t : "))
#1989
        output_filename = 'tes_RNN'#input("Output File Name\t\t : ")
        #str(a)+'-'+str(b)+'-'+str(c)+'-'+str(d)

        #import data from csv
        series = read_csv('../Data/'+data, usecols=[0],
engine='python')
        series.dropna(inplace=True)
        series = series.values
        dataset = series.astype('float32')

        #configure parameter
        n_lag = 12
        n_seq = 12
        n_test = 1
        n_epochs = int(a) #int(input('Epoch\t\t\t\t : '))
        n_neurons = int(b) #int(input('Neuron\t\t\t\t : '))
        n_hidden_layers = int(c)-2 #int(input('Hidden Layer\t\t\t\t
: '))-2

        while n_hidden_layers <= -1 :
            print('\nCaution!!!')
            print('Minimum Hidden Layers is 2')
            print('Please, Re-input Hidden Layers')
            n_hidden_layers = int(input('Hidden Layer\t\t\t\t
: '))-2

            print("")

        val_data = float(d)/100 #float(input('Validation Data
(%) \t\t : '))/100
        n_batch = 1
        activation = 'tanh'
        optimizer = 'adam'
        loss = 'mean_squared_error'

        #make date array
        date = np.array(str(early_year)+'-01',
dtype='datetime64') + np.arange(len(dataset))
        date_array2D = date.reshape(len(date), 1)
        multistep_date = series_to_supervised(date_array2D,
n_lag, n_seq)
        multistep_date = multistep_date.values
        multistep_date = multistep_date.astype('datetime64[M]')

```

```

#display data and configuration
multistep_data = series_to_supervised(dataset, n_lag,
n_seq)

print("")
print("="*50)
print("MultiStep Data")
print("="*50)
print(multistep_data)
print("")
#os.system("pause")

print("")
print("="*50)
print('Neural Network Configuration Parameters')
print("="*50)
print('Length of Data \t\t: %d' %len(dataset))
print('Input Sequences Data \t: %d' %n_lag)
print('Output Sequences Data \t: %d' %n_seq)
print('Training Epochs \t: %d' %n_epochs)
print('The Number of Neurons \t: %d' %n_neurons)
print('Hidden Layers \t\t: %s' %(n_hidden_layers+2))
print('Training Data \t\t: %.2f %%' %((1-val_data)*100))
print('Validation Data \t: %.2f %%' %(val_data*100))
print('Activation \t\t: %s' %activation)
print('Loss \t\t\t: %s' %loss)
print('Optimizer \t\t: %s' %optimizer)
print("")
#os.system("pause")

#prepare data
diff_values = dataset.reshape(len(dataset), 1)
# rescale values to -1, 1
scaler = MinMaxScaler(feature_range=(-1, 1))
scaled_values = scaler.fit_transform(diff_values)
scaled_values = scaled_values.reshape(len(scaled_values),
1)

# transform into supervised learning problem X, y
supervised = series_to_supervised(scaled_values, n_lag,
n_seq)

supervised_values = supervised.values
# split into train and test sets
train, test = supervised_values[0:-n_test],
supervised_values[-n_test:]

```

```

#separate input (X) and output (Y)
trainX, trainY = train[:, 0:n_lag], train[:, n_lag:]
trainX = trainX.reshape(trainX.shape[0], 1,
trainX.shape[1])
testX, testY = test[:, 0:n_lag], test[:, n_lag:]
testX = testX.reshape(testX.shape[0], 1, testX.shape[1])

#define model
model = Sequential()
model.add(LSTM(n_neurons,
return_sequences = True,
activation = activation,
stateful = False,
batch_input_shape =
(n_batch, trainX.shape[1], trainX.shape[2])))

#make n hidden layers
if n_hidden_layers > 0 :
    for _ in range(n_hidden_layers):
        model.add(LSTM(n_neurons,

return_sequences = True,

activation = activation,

stateful = False))

        model.add(LSTM(n_neurons,
return_sequences = False,
activation = activation,
stateful = False))

    model.add(Dense(trainY.shape[1], activation =
activation))
    model.compile(loss = loss, optimizer = optimizer)
    print("")
    print("="*50)
    print("Neural Network Model")
    print("="*50)
    print(model.summary())
    print("")
    #os.system("pause")

#start time for training process

```

```

start = time.time()

#training and testing network
print("")
print("="*50)
print("Training Process")
print("="*50)
loss_history = list()
val_loss_history = list()
for i in range(n_epochs):
    end = time.time()
    times = end - start
    minute = times/60
    print("Time : %.5s second = %.5s minute"
%(times, minute))
    print("Epochs : %d, Remains : %d"
%(i+1,n_epochs-(i+1)))
    history = model.fit(trainX, trainY,
validation_split=val_data, epochs=1, batch_size=n_batch,
verbose=1, shuffle=False)
    loss_history.append(history.history['loss'])

    val_loss_history.append(history.history['val_loss'])
    model.reset_states()
print("")

#predict model
trainPredict = make_forecasts(model, n_batch, train,
n_lag)

testPredict = make_forecasts(model, n_batch, test, n_lag)

#inverse scaling values
trainY = scaler.inverse_transform(trainY)
testY = scaler.inverse_transform(testY)
trainPredict = scaler.inverse_transform(trainPredict)
testPredict = scaler.inverse_transform(testPredict)

#evaluate network
print("")
print("="*50)
print("Evaluate Network")
print("="*50)
trainScore = math.sqrt(mean_squared_error(trainY[:,-1],
trainPredict[:,-1]))

```



```

print('Train Score: %.3f RMSE' %trainScore)
testScore = math.sqrt(mean_squared_error(testY[-1,:],
testPredict[-1,:]))
print('Test Score: %.3f RMSE' %testScore)
print("")

#make output directory
folder1 = 'log'
folder2 = 'csv'
folder3 = 'Plot of Train and Validation Loss'
folder4 = 'Training Results'
folder5 = 'Testing Results'
folder6 = 'Next Year Prediction'
newpath(folder1)
newpath(folder2)
newpath(folder3)
newpath(folder4)
newpath(folder5)
newpath(folder6)

#predict next values
#predict_next_values(model, n_batch, testY, n_lag,
scaler, date, folder6, output_filename)

#display execution time
end = time.time()
execution_time = end - start
minute = execution_time/60
print("Execution Time: %.5s second = %.5s minute"
%(execution_time,minute))
print("")
print("Done...")
print("")

#write file txt and csv
write_files(output_filename, dataset, n_lag, n_seq,
n_epochs,
n_neurons, n_hidden_layers,
val_data, activation, loss,
optimizer, trainScore,
testScore, minute, testY,
testPredict, date)

#show plots

```

```

        plots(output_filename, loss_history, val_loss_history,
folder3, multistep_date,
            multistep_data, trainY, trainPredict,
folder4, n_lag,
            testY, testPredict, folder5)

#run program
#start = time.time()

a = 60
b = 108
c = 2
d = 10
main(a, b, c, d)

"""
end = time.time()
execution_time = end - start
minute = execution_time/60
print("")
print("=*50)
print("Total Execution Time: %.5s second = %.5s minute"
%(execution_time,minute))
"""

```